



**UNIVERSIDAD TÉCNICA DE AMBATO**

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

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES**

**Tema:**

---

**SISTEMA AUTOMATIZADO DE CONTROL Y MONITOREO BASADO EN  
TECNOLOGÍA LORAWAN Y MQTT PARA EL CULTIVO DE  
HORTALIZAS BAJO INVERNADERO**

---

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

**ÁREA:** Comunicaciones

**LÍNEA DE INVESTIGACIÓN:** Tecnologías de Comunicación

**AUTOR:** Freddy Josué Pérez Reyes

**TUTOR:** Ing. Andrea Patricia Sánchez Zumba, Mg.

**Ambato – Ecuador**

**marzo – 2023**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA AUTOMATIZADO DE CONTROL Y MONITOREO BASADO EN TECNOLOGÍA LORAWAN Y MQTT PARA EL CULTIVO DE HORTALIZAS BAJO INVERNADERO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Freddy Josué Pérez Reyes, 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, marzo 2023.

-----  
Ing. Andrea Patricia Sánchez Zumba, Mg.

TUTOR

## AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA AUTOMATIZADO DE CONTROL Y MONITOREO BASADO EN TECNOLOGÍA LORAWAN Y MQTT PARA EL CULTIVO DE HORTALIZAS BAJO INVERNADERO es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023.



---

Freddy Josué Pérez Reyes

C.C. 1803802782

AUTOR

## 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, marzo 2023.



---

Freddy Josué Pérez Reyes

C.C. 1803802782

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Freddy Josué Pérez Reyes, 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 AUTOMATIZADO DE CONTROL Y MONITOREO BASADO EN TECNOLOGÍA LORAWAN Y MQTT PARA EL CULTIVO DE HORTALIZAS BAJO INVERNADERO, 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, marzo 2023.

-----

Ing. Pilar Urrutia, Mg.

PRESIDENTA DEL TRIBUNAL

-----

Ing. Pamela Castro, Mg

PROFESOR CALIFICADOR

-----

Ing. Clara Sánchez, Mg

PROFESOR CALIFICADOR

## **DEDICATORIA**

*El presente trabajo va dedicado principalmente a mi madre Anita, quien día a día realizó el mayor de los esfuerzos para apoyarme, aconsejarme y motivarme en mis estudios. A mi padre que a pesar de no vivir conmigo siempre me apoyó en todo lo que necesitaba demostrando su amor incondicional de padre.*

*A mi hermana Sonnia quien siempre estuvo en los momentos más duros de esta etapa universitaria brindándome su apoyo y motivándome a culminar. A mis hermanos Erika y Kevin quienes me supieron ayudar cada vez que lo necesitaba.*

*A mi familia en general por su amor y apoyo en todo momento, fueron inspiración para poder esforzarme día a día a conseguir este logro.*

*Freddy Josué Pérez Reyes*

## **AGRADECIMIENTO**

*Agradecer primeramente a Dios por acompañarme día a día, por darme la sabiduría necesaria y haberme respaldado en esta etapa universitaria. Sin Dios nada hubiera sido posible.*

*A mi familia, en especial a mis padres, quienes me han educado y guiado por el camino del bien, inculcándome valores y el temor a Dios, además de proporcionarme los recursos económicos necesarios para mis estudios.*

*A mis compañeros de carrera, muchos de ellos hoy en día considerados amigos, con los cuales nos brindamos un apoyo mutuo en cada paso de nuestra vida universitaria.*

*A la Facultad de Ingeniería en Sistemas, Electrónica e Industrial y sus docentes, quienes supieron compartir sus conocimientos y experiencia.*

*A mi tutora, Ing. Andrea Sánchez, quién demostró su confianza en mí, sabiendo ayudarme y guiarme en mi vida estudiantil, sobre todo en el desarrollo de este proyecto.*

*Freddy Josué Pérez Reyes*

## ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN TRIBUNAL DE GRADO.....	v
DEDICATORIA .....	vi
AGRADECIMIENTO.....	vii
ÍNDICE GENERAL.....	xii
ÍNDICE DE TABLAS .....	xvi
ÍNDICE DE FIGURAS.....	xviii
RESUMEN EJECUTIVO .....	xxiii
ABSTRACT.....	xxv
CAPÍTULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.2 Antecedentes Investigativos .....	1
1.3 Contextualización del problema .....	5
1.4 Fundamentación Teórica .....	7
1.4.1 Automatización en el sector agrícola .....	7
1.4.2 Agricultura de Precisión (AP).....	8
1.4.3 Agricultura 4.0 .....	10
1.4.4 Hardware de la agricultura 4.0 .....	11
1.4.5 Software de la agricultura 4.0 .....	11

1.4.6	Ventajas de la agricultura 4.0.....	12
1.4.7	Desventajas de la agricultura 4.0 .....	13
1.4.8	Importancia del cultivo bajo invernadero .....	13
1.4.9	Proceso de producción de hortalizas bajo invernadero .....	14
1.4.10	Principales hortalizas cultivadas en invernadero .....	14
1.4.11	Internet de las cosas (IoT) .....	22
1.4.12	Tecnología LoRa.....	26
1.4.13	Tecnología LoRaWAN .....	27
1.4.14	Protocolo MQTT .....	29
1.4.15	Node-RED.....	31
1.4.16	Sensores y actuadores .....	33
1.5	Objetivos.....	35
1.5.1	Objetivo General .....	35
1.5.2	Objetivos Específicos.....	35
CAPÍTULO II .....		36
METODOLOGÍA .....		36
2.1	Materiales .....	36
2.2	Métodos .....	36
2.2.1	Modalidad de Investigación .....	36
2.2.2	Recolección de Información .....	37
2.2.3	Procesamiento y Análisis de Datos .....	37
2.2.4	Desarrollo del proyecto .....	37
CAPÍTULO III.....		39
RESULTADOS Y DISCUSIÓN.....		39
3.1	Análisis y Discusión de los Resultados .....	39
3.1.1	Análisis de los principales elementos del sistema.....	39

3.1.2	Desarrollo de la propuesta.....	41
3.1.3	Resultados y discusión .....	131
3.1.4	Adquisición de datos con el sistema funcionando solo para monitoreo .....	146
3.1.5	Eficiencia del sistema.....	147
3.1.6	Presupuesto .....	156
CAPÍTULO IV .....		159
CONCLUSIONES Y RECOMENDACIONES.....		159
4.1	Conclusiones.....	159
4.2	Recomendaciones .....	160
REFERENCIAS BIBLIOGRÁFICAS.....		164
ANEXO 1 .....		169
ANEXO 2.....		171
ANEXO 3.....		172
ANEXO 4.....		173
ANEXO 5.....		174
ANEXO 6.....		175
ANEXO 7.....		176
ANEXO 8.....		177
ANEXO 9.....		178
ANEXO 10.....		179
ANEXO 11.....		180
ANEXO 12.....		181

ANEXO 13.....	182
ANEXO 14.....	183
ANEXO 15.....	185
ANEXO 16.....	186
ANEXO 17.....	187
ANEXO 18.....	190
ANEXO 19.....	201
ANEXO 20.....	207

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Taxonomía del pimiento .....	19
<b>Tabla 2.</b> Bandas ICM reguladas por la Arcotel.....	27
<b>Tabla 3.</b> Formato común de los paquetes de control.....	29
<b>Tabla 4.</b> Parámetros meteorológicos para el pimiento .....	42
<b>Tabla 5.</b> Elementos principales de hardware para el desarrollo del prototipo.....	43
<b>Tabla 6.</b> Requerimientos de software para el sistema. ....	44
<b>Tabla 7.</b> Bombas de agua .....	46
<b>Tabla 8.</b> Materiales, equipos y herramientas para la etapa de monitoreo y control de temperatura.....	48
<b>Tabla 9.</b> Materiales para la etapa de monitoreo y control de humedad.....	49
<b>Tabla 10.</b> Materiales para el monitoreo de pH. ....	49
<b>Tabla 11.</b> Materiales para la etapa de monitoreo de iluminación.....	50
<b>Tabla 12.</b> Materiales para el tablero de control del interior del invernadero .....	50
<b>Tabla 13.</b> Materiales para el tablero de control fuera del invernadero.....	51
<b>Tabla 14.</b> Materiales para la placa del nodo sensor.....	52
<b>Tabla 15.</b> Materiales modulo Lora .....	53
<b>Tabla 16.</b> Comparación entre módulos WiFi .....	55
<b>Tabla 17.</b> Comparación entre módulos LoRa .....	56
<b>Tabla 18.</b> Comparación entre microordenadores .....	58
<b>Tabla 19.</b> Comparación entre sensores de temperatura y humedad relativa .....	59
<b>Tabla 20.</b> Comparación entre sensores de humedad del suelo.....	60
<b>Tabla 21.</b> Comparación entre sensores de pH .....	60
<b>Tabla 22.</b> Comparación entre sensores de luminosidad .....	61

<b>Tabla 23.</b> Coordenadas geográficas de los nodos. ....	74
<b>Tabla 24.</b> Distancia Nodos – Gateway .....	74
<b>Tabla 25.</b> Datos sensados. Semana 1 y 2.....	146
<b>Tabla 26.</b> Datos sensados con el sistema en modo automático. Semana 3. ....	147
<b>Tabla 27.</b> Datos sensados con el sistema funcionando en modo automático. Desde la semana 4 a la semana 8. ....	149
<b>Tabla 28.</b> Valores tomados en la mañana del 11 de enero del 2023 .....	152
<b>Tabla 29.</b> Análisis de eficiencia del sistema .....	155
<b>Tabla 30.</b> Presupuesto del sistema de control y monitoreo .....	156

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Esquema ampliado del sistema.....	2
<b>Figura 2.</b> Esquema Red de sensores inalámbrica.....	2
<b>Figura 3.</b> Arquitectura del sistema.....	3
<b>Figura 4.</b> Diagrama de funcionamiento.....	4
<b>Figura 5.</b> Topología total del prototipo.....	5
<b>Figura 6.</b> Ciclo de la Agricultura de Precisión.....	8
<b>Figura 7.</b> Cultivo de lechuga utilizando AP.....	9
<b>Figura 8.</b> Comparación entre una explotación agrícola inteligente (agricultura 4.0) y una explotación en pequeña escala (agricultura convencional).....	10
<b>Figura 9.</b> IoT, Internet de las cosas.....	23
<b>Figura 10.</b> Concepto de IoT.....	23
<b>Figura 11.</b> Arquitectura IoT.....	24
<b>Figura 12.</b> Aplicaciones de la tecnología IoT.....	25
<b>Figura 13.</b> Arquitectura LoRaWAN.....	28
<b>Figura 14.</b> Arquitectura MQTT.....	31
<b>Figura 15.</b> Logo de Node-RED.....	32
<b>Figura 16.</b> Edición de flujo basada en navegador.....	32
<b>Figura 17.</b> Dashboard de Node-RED.....	33
<b>Figura 18.</b> Diferentes tipos de sensores.....	34
<b>Figura 19.</b> Ejemplo de actuadores.....	34
<b>Figura 20.</b> Sistema de control y monitoreo.....	40
<b>Figura 21.</b> Cortinas de ventilación con las que cuenta el invernadero.....	45
<b>Figura 22.</b> Aberturas de ventilación en la parte superior del invernadero.....	45
<b>Figura 23.</b> (1) y (2) Sistema de mangueras para el riego por goteo.....	45

<b>Figura 24.</b> (1) Bomba de agua para el sistema de riego.....	46
<b>Figura 25.</b> Arduino IDE .....	54
<b>Figura 26.</b> Node-RED .....	55
<b>Figura 27.</b> Esquema general del sistema.....	62
<b>Figura 28.</b> Nodo sensor .....	63
<b>Figura 29.</b> Circuito de acondicionamiento de la señal .....	64
<b>Figura 30.</b> Diagrama de flujo del envío de datos .....	65
<b>Figura 31.</b> Conexiones de la ESP32.....	66
<b>Figura 32.</b> Maestro para lectura .....	66
<b>Figura 33.</b> Configuración banda y pantalla Nodos LoRa.....	67
<b>Figura 34.</b> Inicialización comunicación SPI. ....	67
<b>Figura 35.</b> Configuración para comunicación LoRa.....	68
<b>Figura 36.</b> Envío de datos entre nodo LoRa y Gateway .....	68
<b>Figura 37.</b> Control de actuadores .....	69
<b>Figura 38.</b> Circuito control de actuadores y electroválvulas.....	70
<b>Figura 39.</b> Código del maestro para envío de activación de los triac. ....	70
<b>Figura 40.</b> Código de escritura y lectura de los valores sensados.....	71
<b>Figura 41.</b> Código para la activación y desactivación de las salidas digitales.....	71
<b>Figura 42.</b> Declaración de variable flotante para sensor de radiación. ....	72
<b>Figura 43.</b> Declaración de variable flotante para sensor de radiación. ....	72
<b>Figura 44.</b> Etapas del Control del sistema.....	72
<b>Figura 45.</b> Ubicación de los nodos.....	80
<b>Figura 46.</b> Ubicación de nodos vista satélital .....	81
<b>Figura 47.</b> Red Gateway – Nodo Invernadero .....	82
<b>Figura 48.</b> Red Gateway – Nodo Bomba.....	82

<b>Figura 49.</b> Propiedades de las antenas Gateway – Nodo Invernadero .....	83
<b>Figura 50.</b> Propiedades de las antenas Gateway – Nodo Bomba.....	83
<b>Figura 51.</b> Detalles de la comunicación Gateway – Nodo Invernadero.....	84
<b>Figura 52.</b> Detalles de la comunicación Gateway – Nodo Bomba .....	84
<b>Figura 53.</b> Diagrama de flujo de los nodos sensores.....	85
<b>Figura 54.</b> Diagrama de flujo nodo LoRa Invernadero.....	86
<b>Figura 55.</b> Diagrama de flujo nodo LoRa Bomba.....	87
<b>Figura 56.</b> Diagrama de flujo del nodo Gateway .....	88
<b>Figura 57.</b> AsyncWebServer .....	89
<b>Figura 58.</b> Ip del servidor mqtt.....	89
<b>Figura 59.</b> EEPROM.....	89
<b>Figura 60.</b> Servidor web.....	90
<b>Figura 61.</b> WebSocket.....	90
<b>Figura 62.</b> Función submit_guardar1() .....	91
<b>Figura 63.</b> Modificación de id en el gateway.....	91
<b>Figura 64.</b> Función submit_guardar2() .....	91
<b>Figura 65.</b> Modificación de la ip y del puerto en el gateway .....	91
<b>Figura 66.</b> Declaración del puerto en la configuracion de mosquito .....	92
<b>Figura 67.</b> URL para acceder a Node-RED .....	92
<b>Figura 68.</b> SyncWord y CRC .....	93
<b>Figura 69.</b> Invert IQ Signals en los Nodos LoRa.....	94
<b>Figura 70.</b> Invert IQ Signals en el Gateway.....	94
<b>Figura 71.</b> Programación gráfica del Control Manual .....	95
<b>Figura 72.</b> Código del control manual indicador .....	95
<b>Figura 73.</b> Código del control manual .....	95

<b>Figura 74.</b> Programación gráfica para el reinicio del Gateway.....	96
<b>Figura 75.</b> Código del timestamp para el reinicio .....	96
<b>Figura 76.</b> Código del “mqtt out” control/invernadero .....	96
<b>Figura 77.</b> Programación gráfica para la adquisición de datos .....	97
<b>Figura 78.</b> Código del invernadero 16 .....	97
<b>Figura 79.</b> Código del nodo 16.....	97
<b>Figura 80.</b> Código del readsensores/invernadero .....	97
<b>Figura 81.</b> Programación visual del control de actuadores .....	98
<b>Figura 82.</b> Programación grafica del control de luz y cortinas .....	99
<b>Figura 83.</b> Código de programación del nodo inv 16 f2 .....	99
<b>Figura 84.</b> Programación gráfica de la consulta de datos .....	100
<b>Figura 85.</b> Programación gráfica del almacenamiento de datos y su graficado.....	101
<b>Figura 86.</b> Almacenamiento de datos y creación de tablas para el día y la noche ..	102
<b>Figura 87.</b> Controles límites del Setpoint.....	102
<b>Figura 88.</b> Creación de un nuevo bot .....	103
<b>Figura 89.</b> Cambio de descripción e imagen de perfil .....	104
<b>Figura 90.</b> Comandos de control .....	105
<b>Figura 91.</b> Nodos de Node-RED para Telegram.....	106
<b>Figura 92.</b> Nodo Start.....	106
<b>Figura 93.</b> Función 24 .....	106
<b>Figura 94.</b> Función 25 .....	107
<b>Figura 95.</b> Bot Invernadero_control_bot.....	107
<b>Figura 96.</b> Nodo Switch .....	107
<b>Figura 97.</b> Programacion del Nodo Switch.....	108
<b>Figura 98.</b> Nodo trigger de 300ms .....	108

<b>Figura 99.</b> Nodo MySQL, invernaderos.....	108
<b>Figura 100.</b> Programación gráfica para el control y monitoreo desde la app de Telegram .....	109
<b>Figura 101.</b> Programación gráfica para el envío de notificaciones desde la app de Telegram .....	110
<b>Figura 102.</b> Código del envío de notificaciones de alerta a través de Telegram ....	110
<b>Figura 103.</b> Circuito de los Nodos Sensores (ESP32) .....	111
<b>Figura 104.</b> PCB de los nodos sensores (ESP32).....	112
<b>Figura 105.</b> Vista 3D de la placa de los nodos sensores (ESP32).....	112
<b>Figura 106.</b> Circuito de los Nodos LoRa .....	113
<b>Figura 107.</b> PCB de los nodos LoRa.....	113
<b>Figura 108.</b> Vista 3D de la placa de los nodos LoRa .....	114
<b>Figura 109.</b> Carcasa para placas de nodos sensores y nodos LoRa. (a) Vista frontal de la carcasa; (b) Vista trasera de la carcasa; (c) Carcasa con la placa.....	115
<b>Figura 110.</b> Caja para sensores. (a) Vista frontal; (b) Vista trasera .....	115
<b>Figura 111.</b> Carcasa para sensores de humedad del suelo. (a) Vista frontal y lateral; (b) Vista lateral del interior .....	116
<b>Figura 112.</b> Carcasa para sensor de pH. (a) Vista frontal y lateral; (b) Vista trasera .....	116
<b>Figura 113.</b> Placa sumergida en agua con cloruro férrico.....	117
<b>Figura 114.</b> Placas corroídas. (a) Vista frontal; (b) Vista trasera.....	117
<b>Figura 115.</b> Proceso de perforación de placas.....	118
<b>Figura 116.</b> (a) Placa lista para soldar; (b) Placas con todos los componentes.....	118
<b>Figura 117.</b> (a) Colocación de elementos en rieles 1; (b) Colocación de elementos en rieles 2; (c) Prueba de elementos en el tablero eléctrico .....	119

<b>Figura 118.</b> Ubicación de los diferentes elementos en el interior del invernadero .	120
<b>Figura 119.</b> Ubicación de los focos de alta potencia.....	120
<b>Figura 120.</b> (a) Montaje de tablero eléctrico del invernadero; (b) Montaje de tablero eléctrico de la bomba; (c) Vista frontal del tablero de la bomba .....	121
<b>Figura 121.</b> (a) Pruebas de conductividad en tablero de la bomba; (b) Pruebas de conductividad en tablero del invernadero .....	122
<b>Figura 122.</b> (a) Proceso de suelda de tubos; (b) Proceso de suelda de tubos G .....	123
<b>Figura 123.</b> Proceso de fijación del plástico al tubo. ....	123
<b>Figura 124.</b> Ajuste de chumaceras .....	124
<b>Figura 125.</b> Partes de la cortina.....	124
<b>Figura 126.</b> Base para el motor; (b) Fijación del motor a la base .....	125
<b>Figura 127.</b> (a) Conexiones del motor; (b) Fijación de los finales de carrera.....	126
<b>Figura 128.</b> (a) Instalación de electroválvula para el riego; (b) Instalación de electroválvula para aspersores.....	126
<b>Figura 129.</b> (a) Instalación de nebulizadores. (b) Vista del nebulizador ya instalado .....	127
<b>Figura 130.</b> (a) Instalación del sensor de PH; (b) Instalación de la placa del sensor de pH.....	128
<b>Figura 131.</b> Instalación del sensor de humedad del suelo.....	128
<b>Figura 132.</b> Instalación del sensor de humedad y temperatura de ambiente (DTH11) .....	129
<b>Figura 133.</b> (a) Instalación del cableado para la iluminación; (b) instalación de iluminación.....	130
<b>Figura 134.</b> Instalación de ventiladores .....	130
<b>Figura 135.</b> Ajuste de conexiones en la bomba de agua .....	131

<b>Figura 136.</b> Comunicación entre nodos sensores y nodo LoRa del invernadero ....	132
<b>Figura 137.</b> Datos recibidos por el nodo LoRa .....	133
<b>Figura 138.</b> Comunicación entre Servidor – Gateway - Nodos LoRa (envía).....	134
<b>Figura 139.</b> Comunicación entre Servidor – Gateway - Nodos LoRa (recive).....	135
<b>Figura 140.</b> Comunicación entre Node-RED y MySQL.....	136
<b>Figura 141.</b> Secciones del HMI.....	136
<b>Figura 142.</b> (a) Monitoreo promedio de la humedad del aire. (b) Monitoreo sectorial de la humedad del aire.....	137
<b>Figura 143.</b> (a) Monitoreo promedio de la humedad del suelo. (b) Monitoreo sectorial de la humedad del suelo.....	137
<b>Figura 144.</b> (a) Monitoreo promedio de la temperatura ambiente. (b) Monitoreo sectorial de la temperatura ambiente.....	138
<b>Figura 145.</b> Monitoreo del nivel de pH.....	138
<b>Figura 146.</b> Monitoreo de la iluminación.....	138
<b>Figura 147.</b> Monitoreo del consumo eléctrico .....	139
<b>Figura 148.</b> Control General .....	139
<b>Figura 149.</b> Control Manual.....	140
<b>Figura 150.</b> (a) y (b) Control de actuadores .....	140
<b>Figura 151.</b> Login.....	141
<b>Figura 152.</b> SetPoint.....	141
<b>Figura 153.</b> (a) y (b) Botones para graficar.....	142
<b>Figura 154.</b> (a) y (b) Gráficas estadísticas .....	142
<b>Figura 155.</b> (a), (b) y (c) control y monitoreo del sistema a través de Telegram....	143
<b>Figura 156.</b> (a), (b) y (c) control del sistema a través de Telegram .....	144
<b>Figura 157.</b> Mensajes de alerta en la app Telegram.....	145

<b>Figura 158.</b> Base de datos MySQL .....	146
<b>Figura 159.</b> Monitoreo de la humedad del aire .....	150
<b>Figura 160.</b> Monitoreo de la humedad del suelo.....	150
<b>Figura 161.</b> Monitoreo de la temperatura.....	150
<b>Figura 162.</b> Monitoreo de la radiación.....	151
<b>Figura 163.</b> Monitoreo del pH.....	151
<b>Figura 164.</b> Monitoreo de estado .....	151
<b>Figura 165.</b> Producción por semana.....	153
<b>Figura 166.</b> Comparación entre consumo energético y producción.....	154

## RESUMEN EJECUTIVO

El continuo progreso de la tecnología y la aplicación del Internet de las Cosas (IoT) en el sector agrícola ha permitido un avance en los sistemas que favorecen el desarrollo de los cultivos. En el presente proyecto se desarrolla un sistema de control y monitoreo para el cultivo de hortalizas bajo invernadero en la parroquia Huachi Grande de Ambato. El cual tiene como objetivo brindar al productor un sistema autónomo que sea capaz de tomar decisiones inteligentes sin la necesidad de una constante supervisión por parte de una persona. El sistema permite monitorear y controlar la temperatura y humedad, así como los niveles de pH y la cantidad de radiación que reciben las plantas durante el día, además, cuenta con un sistema de iluminación artificial.

El propósito del presente trabajo de investigación es brindar un ambiente favorable para el cultivo dentro de un invernadero, para esto se trabaja con sensores de temperatura y humedad de ambiente, sensores de humedad del suelo, sensor de pH y sensor de radiación solar. Para poder controlar la temperatura en el interior del invernadero se utilizan dos ventiladores y un sistema de cierre y apertura de cortinas automático por medio de un motor eléctrico de bajas revoluciones, un control de humedad del suelo, que se realiza a través de riego por goteo. Por otra parte, para el monitoreo del pH se utiliza un sensor para el agua que ingresa al cultivo a través de las mangueras de riego, finalmente se utiliza un sensor de radiación solar para monitorear la cantidad de radiación recibida por las plantas, para en el caso de ser necesario activar la iluminación artificial con el objetivo de incrementar el periodo de tiempo durante el cual la planta realiza la fotosíntesis. La tecnología utilizada se basa en LoRaWAN y en el protocolo MQTT. El sistema cuenta con dos nodos Lora y un gateway para realizar la comunicación con el servidor y conectar el sistema a internet. Para el control del sistema se utiliza un HMI creado en base a Node-RED, el cual permite monitorear en tiempo real de forma constante, controlar el sistema de forma automática o manual y visualizar estadísticas diarias. Además, el sistema cuenta con un control y monitoreo a través de la aplicación Telegram, a través de la cual el productor recibe notificaciones de alerta y puede controlar el sistema de forma manual.

**Palabras clave:** IoT, LoRaWAN, MQTT, control, monitoreo, agricultura.

## **ABSTRACT**

The continuous progress of technology and the application of the Internet of Things (IoT) in the agricultural sector has allowed an advance in the systems that favor the development of crops. This project develops a control and monitoring system for the cultivation of vegetables under greenhouse in the Huachi Grande parish of Ambato. The objective is to provide the grower with an autonomous system capable of making intelligent decisions without the need for constant supervision by a person. The system allows monitoring and control of temperature and humidity, as well as pH levels and the amount of radiation that the plants receive during the day, and also has an artificial lighting system.

The purpose of this research work is to provide a favorable environment for cultivation inside a greenhouse, for this we work with temperature and humidity sensors, soil moisture sensors, pH sensor and solar radiation sensor. In order to control the temperature inside the greenhouse, two fans and an automatic curtain closing and opening system by means of a low-speed electric motor are used, as well as soil humidity control, which is carried out by means of drip irrigation. On the other hand, for pH monitoring, a sensor is used for the water that enters the crop through the irrigation hoses. Finally, a solar radiation sensor is used to monitor the amount of radiation received by the plants, in order to activate artificial lighting, if necessary, with the objective of increasing the period of time during which the plant performs photosynthesis. The technology used is based on LoRaWAN and the MQTT protocol. The system has two Lora nodes and a gateway to communicate with the server and connect the system to the Internet. To control the system, an HMI based on Node-RED is used, which allows constant real-time monitoring, automatic or manual control of the system and visualization of daily statistics. In addition, the system has a control and monitoring through the Telegram application, through which the producer receives alert notifications and can control the system manually.

**Keywords:** IoT, LoRaWAN, MQTT, control, monitoring, agriculture.

# **CAPÍTULO I**

## **MARCO TEÓRICO**

### **1.1 Tema de investigación**

“SISTEMA AUTOMATIZADO DE CONTROL Y MONITOREO BASADO EN TECNOLOGÍA LORAWAN Y MQTT PARA EL CULTIVO DE HORTALIZAS BAJO INVERNADERO”

### **1.2 Antecedentes Investigativos**

Para el presente proyecto se toman en consideración tres proyectos de investigación y dos artículos científicos referentes a la aplicación de la tecnología Lora y MQTT en el control y monitoreo de cultivos agrícolas. Se seleccionó la documentación más reciente y relevante, la cual se detalla a continuación.

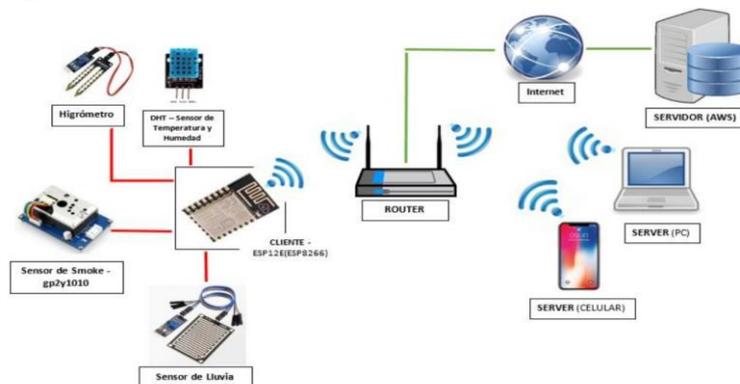
En el año 2019, en Cantabria (España), Eva Hernández, en su trabajo “Desarrollo de un Sistema de monitorización y control de un invernadero aplicando Tecnología IoT” de tipo experimental desarrolla un sistema de control inteligente para invernaderos domésticos. El sistema desarrollado utiliza como elementos principales un módulo ESP32, sensores, una bomba de agua, un ventilador y un panel fotovoltaico para incrementar su autonomía. Para la conectividad del sistema el investigador utiliza un Servidor Adafruit IO, el protocolo TCP/IP y MQTT para enviar los datos recolectados en los sensores hacia la nube y que posteriormente el usuario pueda visualizarlos a través de un sitio web. Además, el sistema cuenta con la función de envío de alertas por email utilizando la plataforma IFTTT. El resultado del proyecto es un prototipo completo capaz de mostrar los datos del invernadero en tiempo real en un sitio web, enviar alertas a través de correo electrónico y tomar decisiones como en el encendido y apagado del ventilador o el control del riego. En la figura 1 se puede observar un esquema ampliado del sistema. [1]



**Figura 1.** Esquema ampliado del sistema

**Fuente:** [1]

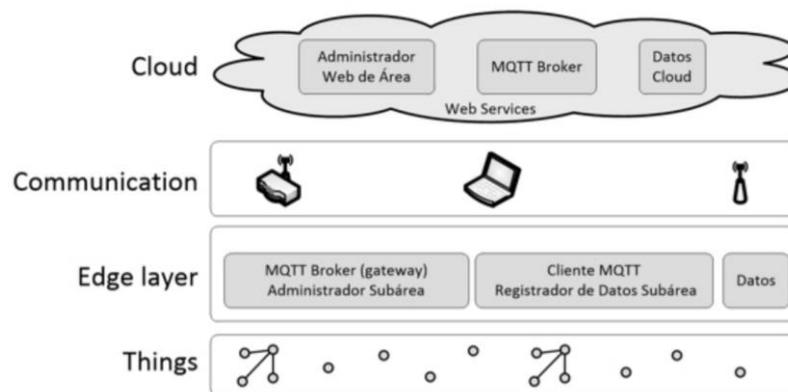
En el año 2019, en Lima (Perú), Hansell Mora y José Rosas, en su proyecto de investigación “Diseño, desarrollo e implementación de una red de sensores inalámbricos (WSN) para el control, monitoreo y toma de decisiones aplicado en la agricultura de precisión basado en Internet de las Cosas (IoT). – Caso de estudio cultivo de frijol” de tipo experimental, se crea un dispositivo electrónico capaz de establecer una red de sensores inalámbricos utilizando el MCU ESP8266 en su módulo ESP-12E aplicando IoT. Los datos son enviados hacia la nube a la interfaz gráfica de Node-Red. Para el software se crea un EC2 en la plataforma de AWS y se programa en el IDE de Arduino para el ESP12E. Al implementar el sistema realizan pruebas comparando los datos recibidos en el Dashboard de Node-Red provenientes de los sensores con los datos arrojados por aparatos electrónicos creados específicamente para medir temperatura y humedad, donde observan una diferencia de 0.1°C en la temperatura y un 4% en la humedad, concluyendo que el sistema es confiable. En la figura 2 se puede observar un esquema de la red de sensores inalámbrica. [2]



**Figura 2.** Esquema Red de sensores inalámbrica

**Fuente:** [2]

En el año 2019, en Atacama (Chile), Villarroel C., Goykovic V., Collao P., Barraza M., Fernández J., Villarroel Á., Valdivia R., y Castro C., en su estudio “Evaluación de desempeño de un invernadero ubicado en el desierto de Atacama, Chile, a través de IoT”, de tipo experimental, el principal nodo sensor utilizado en la WSN, es el módulo ESP8266, compatible con el protocolo TCP/IP, donde emplean sensores de temperatura y humedad ambiental. La arquitectura de monitoreo propuesta se compone de dispositivos inalámbricos conectados entre sí por medio de WiFi y protocolos de comunicación inalámbrica de la IEEE tales como el 802.15.4, además del protocolo de comunicación MQTT. Todos los datos obtenidos son enviados a un servidor, a través de protocolos de IoT para ser procesados y presentados en la plataforma IoT Cloud. Al realizar pruebas de funcionamiento en el mes de mayo obtuvieron una temperatura ambiente máxima de 34.83°C, una temperatura ambiente mínima de 12°C y una temperatura ambiente promedio de 20.33 °C, de la misma forma obtuvieron una humedad de 56.67%, 27.83% y 36.62% respectivamente, analizando los datos de 5 meses concluyeron que el suelo de la zona es extremadamente salino, la conductividad eléctrica (CE) en extracto saturado excede los 16,0mS·cm-1 con una alta presencia de sulfatos, cloruros, calcio y sodio. En la figura 3 se puede observar la arquitectura del sistema. [3]

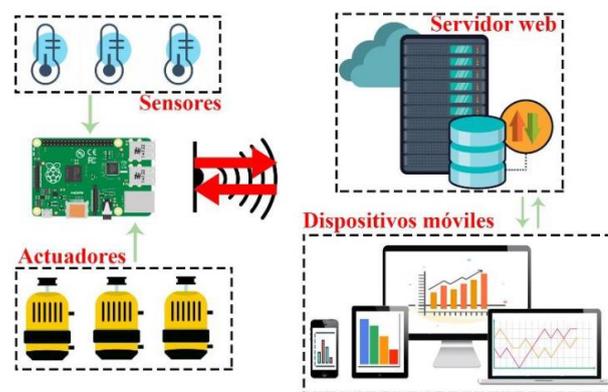


**Figura 3.** Arquitectura del sistema

**Fuente:** [3]

En el año 2020, en Guadalajara (México), Aguilar Y., Sosa S., Galindo J., Morales A., Gallardo R. y Villaseñor I., en su estudio “Diseño y desarrollo de un sistema de monitoreo remoto implementando Internet de las cosas” de tipo experimental desarrollan un sistema de monitoreo en tiempo real de datos atmosféricos (temperatura, humedad y presión atmosférica) recolectados y procesados por una

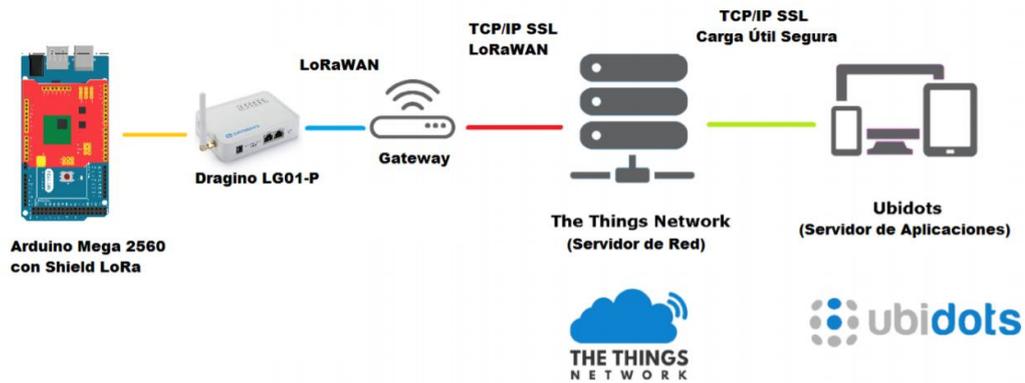
tarjeta Raspberry Pi 3B+, una tarjeta Arduino Uno y un circuito integrado ATMEGA32. La arquitectura del sistema IoT utilizada, está conformado por la parte lógica y la física; que, a su vez, se clasifican en cuatro capas: capa de interfaz donde utiliza HTML, PHP y JavaScript, capa de datos, capa de acceso y la capa de dispositivos. El resultado es una estación de monitoreo multidisciplinar que muestra información en tiempo real en una página web montada en su propio dominio en la cual es posible seleccionar la estación de monitoreo, de esta forma posibilita trabajar en más de un solo lugar, además toda la información es capaz de ser organizada y procesada gracias a una base de datos relacional. En la figura 4 se puede observar un diagrama de funcionamiento del sistema. [4]



**Figura 4.** Diagrama de funcionamiento.

**Fuente:** [4]

En el año 2021, en Quito, Ecuador, Verónica Soria, presenta a través de su proyecto de investigación “Desarrollo de una red IoT con tecnología Lora para gestión de invernaderos” de tipo experimental una aplicación de redes de área extensa y bajo consumo en conjunto con el concepto IoT. La transmisión LoRa entre el nodo cliente utiliza el Dragino LoRa Shield y El Arduino Mega 2560 R3 basada en el microcontrolador ATmega2560 de Atmel. En comunicación hace uso de la tecnología LoRaWAN con acceso a un Gateway el cual recibe los datos de los sensores y los envía a través de internet, el servidor de red para almacenar la información es la plataforma TTN. Para comprobar el funcionamiento del sistema realizan pruebas de envío y recepción de paquetes de datos con un tiempo entre paquetes enviados de 10 minutos, tras realizar 5 envíos el porcentaje de paquetes recibidos es del 100%. Además, al integrar la plataforma TTN con el servicio en la nube de Ubidots los datos pudieron ser visualizados de forma gráfica permitiendo un mejor análisis. En la figura 5 se puede observar un diagrama de la topología total del prototipo. [5]



*Figura 5. Topología total del prototipo*

*Fuente: [5]*

### 1.3 Contextualización del problema

El cultivo en invernadero permite establecer las condiciones óptimas para el buen desarrollo de la producción, ya que consta de un aislamiento con el exterior. La producción bajo invernadero consigue un rendimiento por unidad de superficie hasta tres veces mayor comparado con campo abierto, y dependiendo de la tecnología utilizada el rendimiento puede ser incluso mayor. Los daños causados por el cambio climático no se pueden evitar, pero si reducir [6]. Al tener una producción protegida bajo una estructura, los daños sufridos se minimizan. Un invernadero correctamente diseñado y construido facilita el control de plagas y enfermedades, debido a su aislamiento con el exterior y su cuidadoso sistema de control.

Hoy en día, el cultivo de hortalizas bajo invernadero en el territorio ecuatoriano se realiza utilizando métodos poco tecnificados, en su gran mayoría simplemente cuentan con riego por goteo o un sistema de microaspersión. No se utilizan sistemas tecnificados, que permitan tomar decisiones en base a los diferentes parámetros o variables que se puedan medir en el interior de un invernadero y que influyen directamente en la producción, provocando situaciones poco favorables para los diferentes cultivos como por ejemplo: exceso de humedad en el suelo o aire, temperaturas demasiado elevadas, valores del pH fuera del rango recomendado, entre otros, lo que genera diferentes enfermedades en las plantas reduciendo considerablemente la producción e incrementando el tiempo de cultivo.

Para poder mejorar la producción dentro de un invernadero es necesario la instalación de diferentes sensores que muestren valores reales de variables como temperatura, humedad, pH, e incluso la cantidad de  $Wh/m^2$  que reciben las plantas de la radiación

solar. En la actualidad, las grandes industrias se enfocan en desarrollar sistemas de control y monitoreo para invernaderos a gran escala. Una publicación del Ministerio de Agricultura, Alimentación y Medio Ambiente de España menciona diferentes métodos de automatización tales como interruptores y programadores de horarios, termostatos e higrostatos, autómatas programables o el control por computadoras. Este tipo de automatización implica un alto costo, provocando que gran parte de los productores ecuatorianos opten por no tecnificar sus invernaderos, por este motivo es de gran importancia desarrollar sistemas de control y monitoreo para cultivos a baja escala en invernaderos, y que estos tengan un costo al alcance de los productores con nivel económico medio. [7]

En el año 2021, la página web oficial de la empresa Spagnol, en su publicación, "Automatización y eficiencia" menciona que el beneficio que se puede obtener de las cosechas en invernaderos está relacionado directamente con el costo de la mano de obra y el de los agro-insumos, debido a ello menciona que la automatización y eficacia son puntos claves para obtener beneficios. El automatizar un invernadero incide directamente en poder aplicar estrategias apropiadas para cada cultivo, permite mejorar el proceso de crecimiento y producción, reduce el empleo de mano de obra, reduce en modo significativo el consumo de agua, fertilizantes, pesticidas y productos químicos, consiguiendo mejorar la calidad y la uniformidad de las plantas ya que el manejo de enfermedades es más eficaz con estrategias climáticas apropiadas. [8]

El proyecto es factible ya que se cuenta con amplias fuentes bibliográficas para poder cubrir cada parte de la investigación. Por otro lado, la tecnología Lora e IoT, se centran en aplicaciones agrícolas mejorando el control y producción de las plantaciones, permitiendo controlar el sistema de forma remota e intercambiando información de forma rápida y en tiempo real, así como un ahorro energético. Además, se pretende utilizar el protocolo MQTT debido a su sencillez y ligereza, ya que utiliza un ancho de banda mínimo, es escalable, cuenta con asincronismo, desacoplamiento entre clientes y un mecanismo de calidad del servicio o QoS. Finalmente, el costo de la implementación del sistema no es excesivo, puesto que son elementos, dispositivos y equipos que se pueden adquirir fácilmente en el país.

Con el objetivo de incrementar la producción de hortalizas bajo invernadero es necesario monitorear y controlar la temperatura ambiente, la humedad en suelo y ambiente, monitorear el pH del agua utilizada para el riego, así como la cantidad de

radiación solar que reciben las plantas, ya que se pretende realizar la instalación de radiación artificial para conseguir que la planta realice su proceso fotosintético adecuadamente y durante más tiempo, incrementando la floración, la fructificación, el ahijamiento y la calidad del fruto. El sector agrícola se verá directamente beneficiado, puesto que es un sistema que se puede utilizar para el cultivo de diferentes plantaciones, únicamente modificando los rangos de medición. El sector floricultor también podría verse beneficiado directamente puesto que en su gran mayoría trabajan bajo invernadero y el sistema es aplicable para ese tipo de cultivos. Todo esto beneficiará a la población en general, no solo a los productores, también a los comerciantes y consumidores, sin dejar de lado que es un sistema amigable con el medio ambiente ya que permitiría el ahorro de agua y agro-insumos.

En base a todo lo expuesto, se denota la importancia de realizar esta investigación, ya que contribuye al desarrollo social, económico y productivo del sector agrícola al incorporar nuevas tecnologías en los procesos de producción, mejorando así la calidad del producto e incrementando la cantidad de producto cultivado.

## **1.4 Fundamentación Teórica**

### **1.4.1 Automatización en el sector agrícola**

La automatización es una realidad que continúa avanzando en labores como riego, procesamiento de frutas y verduras, polinización, poda y cosecha. Hoy en día, controlar el riego a distancia con el botón del teléfono celular es algo que se está volviendo más habitual y necesario, así como contar con sistemas masivos de adquisición de datos en los campos a través de sensores o drones. En la actualidad los desarrolladores de tecnología agrícola están yendo más allá ya que necesitan hacer más con menos recursos, debido a que el agua, suelo y clima son elementos cada vez más limitados. En un futuro se va a necesitar producir un 50% más de alimentos para abastecer a los 9.000 millones de personas que habitarán el planeta en 2050. [9]

Desde las empresas, laboratorios y universidades han ido desarrollando softwares, aplicaciones y dispositivos que simplifiquen el proceso de producción con el propósito de acortar tiempos, mantener un mayor control de plagas o evitar pérdidas de cosechas. Los países de Europa y Estados Unidos han sido los pioneros en dotar de inteligencia artificial y tecnología de primera al sector agro, en América Latina estos avances de la automatización no han sido tan acelerados por diversos factores como la cultura de uso

de fertilizantes, el cambio climático, resistencia al cambio, economía e incluso la pandemia. [10]

#### 1.4.2 Agricultura de Precisión (AP)

Se conoce como Agricultura de Precisión a una forma o estrategia de administración que hace uso de tecnología de la información y las comunicaciones para recolectar información útil con el fin de apoyar decisiones asociadas a producción de cultivos. [11]

- Beneficios de la Agricultura de Precisión
- Reducción de costos – disminución en la utilización de insumos.
- Mayor rendimiento de los cultivos con el mismo nivel de insumos.
- Incremento de la calidad de las cosechas debido a una más adecuada combinación de los requerimientos y los insumos aplicados.



**Figura 6.** Ciclo de la Agricultura de Precisión.

**Fuente:** [12]

En la figura 6, se observa como el ciclo inicia con la adquisición de datos en los cultivos, así como de su entorno, para ello se utilizan sensores, muestreos convencionales y observaciones visuales. Una vez que los datos han sido recopilados, hay que extraer la información, uno de los aspectos más importantes para el agricultor es saber si el cultivo se está desarrollando correctamente y de forma uniforme en todo

el terreno. La información extraída se utiliza en la etapa de toma de decisiones, en esta etapa se toman decisiones tales como si factible seguir manejando todo el cultivo de forma uniforme o si su desarrollo ha sido diferente a lo largo del terreno y es necesario fraccionar el cuidado del terreno. Finalmente se llega a la etapa de actuar en el campo, donde se aplican los recursos necesarios para el cultivo, en el caso de ser necesario en esta etapa se utilizan las llamadas tecnologías de Actuación Variable, que permiten que los equipos dosifiquen los recursos dependiendo de los datos recopilados en campo. [12]

La Agricultura de Precisión permite recolectar, interpretar y aplicar datos específicos de las explotaciones, transformando datos e información en conocimiento y rentabilidad. Estos sistemas además de permitir al agricultor tener una gestión agronómica más eficaz, permite incrementar la precisión de las labores y la eficiencia de los equipos. En la figura 7 observa un cultivo de lechuga utilizando AP. [11]



**Figura 7.** Cultivo de lechuga utilizando AP.

**Fuente:** [11]

Barreras de la Agricultura de Precisión [13]:

- Acceso a Internet. Existen zonas rurales a las que aún no llega el internet o llega con dificultades y muy limitado, dificultando su uso en el sector agrícola.
- Acceso limitado. La inversión necesaria para implementar la agricultura de precisión es considerable, limitando su aplicación a medianas y grandes explotaciones.
- Compatibilidad. Normalmente la maquinaria antigua no es compatible con los nuevos dispositivos o requieren altos gastos para poder adaptarlos.

- Conocimientos. Es necesaria una formación básica en informática para que el agricultor tenga la capacidad de implementar y utilizar el sistema.

### 1.4.3 Agricultura 4.0

La agricultura 4.0 consiste en un tipo de agricultura que combina un conjunto de innovaciones para producir productos agrícolas. Estas innovaciones engloban a agricultura de precisión, el internet de las cosas (IoT) y macrodatos para lograr una mayor eficiencia en la producción. [14]

La agricultura continúa evolucionando constantemente con la ciencia y la tecnología, al punto que ya está llegando el Internet de las Cosas (Internet of Things, en inglés, y su abreviatura IoT) a la industria agrícola. Los avances técnicos de las nuevas tecnologías agrícolas deberían [14]:

- Optimizar la calidad del producto;
- Optimizar la eficiencia de la producción;
- Minimizar los riesgos relacionados con la producción;
- Minimizar el impacto ambiental.



**Figura 8.** Comparación entre una explotación agrícola inteligente (agricultura 4.0) y una explotación en pequeña escala (agricultura convencional)

**Fuente:** [14]

En la figura 8 se observa una comparación entre una explotación agrícola inteligente y una explotación en pequeña escala. Algunos ejemplos que reflejan estos avances en

la tecnología se pueden observar en [14]:

- La agricultura de precisión;
- La inteligencia artificial para diagnóstico de plagas y enfermedades, así como las opciones de control;
- La adopción de la tecnología de cadenas de bloques en las cadenas de valor (por ejemplo, en el procesamiento, etiquetado, envasado, clasificación, lavado, almacenamiento o transporte);
- La teledetección (utilización de imágenes a través de drones o satélites);
- El uso de sensores en el terreno (estaciones meteorológicas, suelo o cultivo);
- Los equipos automatizados para las operaciones agrícolas.

#### **1.4.4 Hardware de la agricultura 4.0**

El hardware más prominente en la agricultura 4.0 involucra a los robots y sus sensores. Los robots abarcan los drones (aéreos o acuáticos), así como los tractores sin conductor, más pedestres. Estos equipos cuentan con Inteligencia Artificial y un número considerable de sensores los cuales pueden ser biológicos o eléctricos, acústicos, visuales u olfativos, las imágenes pueden llegar a ser desde hiperspectrales tomadas a través de un satélite hasta aplicaciones de teléfonos inteligentes con reconocimiento facial [15].

Robots o bots: la mayoría de los robots están lejos de ser inteligentes, la mayoría solo ejecutan tareas rudimentarias en una línea de montaje, algunos incluso pueden ser reprogramados para realizar otras labores [15].

Drones: aéreos o acuáticos. Los drones aéreos pueden barrer los campos, detectar y rociar la maleza, permitiendo al agricultor ahorrar combustible y reducir las toxinas [15].

Sensores: normalmente se utilizan conectados con robots u otros dispositivos de inteligencia artificial, los sensores son capaces de oler, saborear, oír, ver o sentir, o cualquiera de sus combinaciones, al estar en contacto con el suelo o desde arriba a través de aviones y satélites [15].

#### **1.4.5 Software de la agricultura 4.0**

La utilización de tecnologías en sectores específicos es una realidad, en el sector agro, por ejemplo, la tendencia se basa en el uso de tecnologías de máquina a máquina,

servicios de GPS, el Internet de las Cosas que se enfoca en el uso de sensores y el Big Data para optimizar el rendimiento de los cultivos pudiendo disminuir considerablemente el volumen de desperdicios. Para ello en el mercado existen diferentes empresas las cuales se centran en brindar un servicio completo planeando a futuro, controlando el presente y analizando el pasado [16].

- Planea
- Agendas técnicas basadas en mejoramiento del suelo, fertilización y abonado.
- Presupuestos para diferentes cultivos y sistemas de producción.
- Comparación de proyectos de inversión alternativos.
- Estrategias de crecimiento y diversificación del negocio.
- Controla
- Agenda diaria de cada predio.
- Mejor desempeño de los técnicos de campo.
- Información del clima en tiempo real y el pronóstico de cualquier predio.
- Registro de eventos en una bitácora directamente de campo.
- Indicadores de desempeño.
- Cortes parciales y estados de resultados durante y al final del ciclo.
- Analiza
- Análisis de datos (Big Data, Data Analytics).
- Indicadores de desempeño de cualquier ciclo de producción y de toda la operación.
- Archivos históricos de cada predio.
- Rastreabilidad de productos en anaquel.
- Consultas y reportes exportables a otros formatos.

#### **1.4.6 Ventajas de la agricultura 4.0**

Aumento de la productividad. Al utilizar drones y sensores es posible conocer las necesidades de las plantas.

Mejora el proceso de toma de decisiones. Debido a la gran variedad y cantidad de datos que se recogen es posible tomar decisiones en base a las necesidades de las plantas.

Reduce el impacto en el medio ambiente. Se reduce el uso de los recursos al conocer perfectamente las necesidades de las plantas.

Reduce el uso de químicos. Es posible evitar malgastos de suministros debido a que se tiene un mayor control en enfermedades y plagas.

Reduce los costos. Una mejor gestión de los recursos y suministros provoca una reducción en la utilización de estos disminuyendo a su vez los importes de las facturas [17].

#### **1.4.7 Desventajas de la agricultura 4.0**

**Inversión.** Es necesario una inversión de capital para adaptar los dispositivos existentes a las nuevas tecnologías.

**Acceso a Internet.** Hay explotaciones agrícolas muy alejadas de las ciudades donde la señal de internet es muy débil.

**Formación.** Los agricultores necesariamente deben recibir capacitaciones para poder manejar los nuevos dispositivos [17].

#### **1.4.8 Importancia del cultivo bajo invernadero**

Se conoce como invernadero a un recinto cerrado delimitado por una estructura de metal o madera, recubierta por algún material plástico de naturaleza transparente, donde normalmente se cultiva hortalizas o plantas ornamentales, en estaciones del año donde las condiciones climáticas externas no son las más favorables para obtener el producto deseado. [18]

Hoy en día la población demanda cada vez más cantidad de productos agrícolas de alta calidad y así como una gran diversidad de ellos durando todo el transcurso del año, debido al continuo cambio climático esto se torna difícil de cumplir para los agricultores, por este motivo el uso de invernaderos es tan importante para contrarrestar dicho efecto. [19] Los invernaderos permiten tener unas condiciones climáticas adecuadas para la planta en el interior de un espacio cerrado, creando una barrera física entre el interior del invernadero y el clima exterior, protegiendo a los cultivos de daños meteorológicos como de lluvias demasiado fuertes o la caída de granizo, de la misma forma permite al agricultor tener un mayor control de plagas y enfermedades. En conclusión, los invernaderos posibilitan al agricultor a cosechar durante todo el año, reducir el tiempo entre cada cosecha, así como obtener productos agrícolas de mayor calidad. [20]

#### **1.4.9 Proceso de producción de hortalizas bajo invernadero**

El uso de los invernaderos varía tanto de un país a otro que no es posible ofrecer un prototipo de cultivo que sea el mismo para todos, pero si similares. Entre las razones de esa diversidad se pueden mencionar:

- Peculiaridades del clima local. Los tipos de plantas cultivadas se adaptan naturalmente a las condiciones climáticas locales. Las condiciones climáticas del verano pueden afectar la elección de los cultivos a sembrar e incluso crear una tendencia a no utilizar invernaderos en lugares donde el calor es elevado en los meses más calurosos.

**En algunos países**, las plantas con altos requerimientos de radiación y temperatura, como los tomates, se cultivan a principios de invierno, aunque cuando se cultivan en invernaderos fríos existe el riesgo de bajas temperaturas cercanas o incluso por debajo del umbral fisiológico aceptable. Este riesgo se compensa con los beneficios de enviar el producto al mercado europeo.

Factores sociológicos del país. En ciertos países, la demanda local de productos bajos en calorías y que tienden a cultivarse bajo una cubierta fría en invierno es casi inexistente.

#### **1.4.10 Principales hortalizas cultivadas en invernadero**

##### **1.4.10.1 Tomate**

###### **Descripción general**

El tomate es una planta herbácea en las primeras etapas de crecimiento porque el tallo se vuelve ligeramente leñoso en las últimas etapas de crecimiento. Sin embargo, para su cultivo requiere estacas porque el área del cuello es muy débil. Solo las plantas que son enanas o pequeñas se pueden cultivar sin apoyo. [21]

###### **Botánica**

Las raíces de los tomates jóvenes son raíces pivotantes, pero la ramificación posterior se vuelve tan importante que es difícil distinguir la raíz principal de las raíces secundarias. Las raíces de los tomates pueden alcanzar una profundidad de más de

1,25 m, pero la mayoría de las raíces se encuentran en la capa superior del suelo (más del 70 % en la zona de 0-20 cm). En general, las variedades determinantes del crecimiento tienen raíces más pequeñas y menos profundas; Las plantas de siembra directa tienen raíces primarias más importantes y raíces más profundas que las plantas trasplantadas. [21]

Las plantas de tomate se caracterizan por un crecimiento simpodial. Como regla general, con el desarrollo de 5-8 hojas, aparece un racimo floral terminal en el tallo principal. Las altas temperaturas y la poca radiación, especialmente una combinación de las dos, retrasarán la aparición de este primer racimo floral. El tallo principal se convierte en brotes secundarios más fuertes que crecen desde la esquina de la hoja justo debajo del racimo. El crecimiento continúa con brotes laterales más fuertes y brotes laterales inferiores más débiles. [21]

Las hojas son complejas y tienen un número impar de folíolos verdes, dependiendo de la variedad y ubicación de las hojas en la planta. Asimismo, la forma, el tamaño, la estructura, el grosor y el color son factores que también dependen de la variedad. En general, las hojas de las variedades tardías son más gruesas y oscuras, aunque también afectan a las condiciones de cultivo.

Cuando la planta es muy fuerte, las hojas giran alrededor de su eje, mientras que el hinchamiento de las hojas jóvenes puede ser causado por estrés hídrico o presión, especialmente en el tipo determinado. [21]

La fruta de buen tamaño generalmente se obtiene de flores de buena calidad que se desarrollan en racimos de 5-12 flores. Dependiendo de las condiciones ambientales, el número de flores puede ser muy bajo a altas temperaturas y poca radiación o muy grande a bajas temperaturas, mientras que las plantas que prosperan en condiciones húmedas producirán racimos con brotes de hojas y flores con sépalos muy largos. A bajas temperaturas, algunas variedades tienden a formar flores dobles, produciendo frutos acostillados. [21]

En general, el polen es más sensible a las bajas temperaturas que los óvulos, mientras que la viabilidad de los óvulos también es muy baja a altas temperaturas. Tanto las

temperaturas bajas por debajo de los 13°C como las altas por encima de los 32°C dificultan la reducción del polen. Por otra parte, también la calidad del polen se ve afectada por las bajas temperaturas, especialmente cuando la flor se encuentra en estado de pequeño botón, aproximadamente dos semanas antes de la floración. [21]  
Las bajas temperaturas y la alta humedad relativa impiden la liberación de polen.

Cuando la calidad del polen es mala, la baja temperatura retrasará la germinación del polen y la formación de óvulos; Estos problemas no son graves en presencia de polen de buena calidad. [21]

El número de semillas que contiene el fruto y su volumen es directamente proporcional al número de óvulos fecundados. Si la cantidad y la calidad del polen son insuficientes, y la viabilidad del óvulo es insuficiente, especialmente a bajas temperaturas nocturnas, se pueden usar estimulantes de plantas fortificados con auxinas sintéticas. Los resultados obtenidos dependerán de factores como la variedad, condiciones ambientales o tipo de fumigaciones empleadas. Tanto el uso de altas concentraciones como el uso de productos agresivos en variedades sensibles dará como resultado una distorsión y una fruta excesivamente acanalada, hueca por dentro o puntiaguda o hueca. [21]

Generalmente, los tomates de invernadero son redondos, de tamaño mediano, con 2-5 compartimentos y de color rojo.

El tamaño del fruto depende directamente del número de óvulos fertilizados, pero además de factores como la nutrición, el riego, la temperatura y el número de células, si la planta está bien provista de agua y nutrientes, incluso a bajas temperaturas, producirá bayas grandes y rugosas. Las variedades expuestas a temperaturas de agua más altas producen frutos más pequeños y redondos. [21]

La calidad es un factor comercial muy importante medido por una serie de factores como la consistencia, el color, el sabor, la resistencia al transporte y la capacidad de almacenamiento.

- El aspecto del fruto va en función de la genética como de las condiciones de

crecimiento. La alta temperatura en el invernadero ayuda a suavizar la fruta. Para obtener frutos firmes, la temperatura debe ser inferior a 30°C. Al mismo tiempo, se debe lograr un buen manejo del riego

- El color del tomate debe ser lo más uniforme posible. Además del clásico rojo por su contenido en licopeno, se presenta en amarillo, rosa, morado y rojo oscuro. Hay frutos que alcanzan la madurez y no son de color uniforme. Otro cambio de color es la aparición de collares amarillos en frutos maduros a temperaturas muy altas por encima de los 25°C. Esto se debe al crecimiento débil debido a la deficiencia de potasio y la compactación excesiva.

- El sabor de la fruta está determinado principalmente por la acidez y el contenido de azúcar que deben ser cantidades significativas para que el producto sea de buena calidad.

- La fruta debe tener un pH inferior a 4,4 y un contenido total de azúcar superior al 4-4,5%. Las temperaturas más altas (30-35°C) reducen la acidez y reducen el sabor a fruta. [21]

### **Requerimientos del cultivo**

Luminosidad o radiación.

Los tomates son cultivos que no se ven afectados por la duración del día o el fotoperíodo. Las necesidades de radiación varían de 8 a 16 horas por día. [22]

La temperatura

El rango óptimo de crecimiento de la planta es 28-30°C durante el día y durante la noche 15-18°C durante el día. Temperatura superior a 35°C y menos de 10°C durante la floración provoca caída de flores y limitación del cuajado. Aunque hay material genético que si cuajan a altas temperaturas. [22]

### **Humedad**

Humedad relativa

La humedad relativa que mejor le va a la planta de tomate es del 65-70%. Dentro de este margen de humedad prima el desarrollo adecuado de la polinización dando como resultado una buena producción. [22]

Humedad del suelo

La humedad del suelo que mayor beneficio da y donde la planta se encuentra en mejor condición para poder producir está en el rango del 60-80%. [22]

### **Suelos**

El suelo debe suministrar a las plantas cuatro necesidades principalmente, oxígeno, agua, nutrientes y soporte. Los suelos más aptos para la producción de tomate son de media o mucha fertilidad, profundos y drenados. Otro factor importante es el pH, el cual debe estar entre el 5.9 y 6.5, de esta forma se aprovecharán de mejor manera los fertilizantes. [22]

#### **1.4.10.2 Pimiento**

##### **Descripción general**

Según varios estudios el pimiento es de origen boliviano y peruano donde se cultivan varias especies además de la conocida especie *Capsicum annum*. Al continente europeo se conoce que fue llevado por Cristóbal Colón en el año 1493, en su primer viaje. El cultivo de pimiento en España ya se había difundido en el siglo XVI y de ahí paso al resto del continente y del mundo. [23]

##### **Botánica**

Su sistema de raíces es una raíz principal con raíces profundas, y muchas raíces invertidas cubren un área amplia horizontalmente.

Su tallo principal es de crecimiento un tanto limitado y erecto. Echa 2-3 ramas desde cierta altura y continúa ramificándose hasta el final de un ciclo. Asimismo, sus tallos secundarios se ramifican tras desarrollar unas pocas hojas. Por este motivo, es necesario vigilar las ramas para evitar romperlas. [24]

Como la cualquier otra planta de hortalizas, el pimiento consta de cuatro etapas de crecimiento. En la etapa de germinación el pimiento realiza correctamente este proceso a temperaturas cálidas de suelo, entre los 25 y 30 °C, a pesar de que la temperatura idónea es de 30 °C. La mayoría de las semillas crecerán a los 10 días, aunque hay casos en los que tardan hasta seis semanas. En la etapa de crecimiento el pimiento que es a partir de la semana número seis el pimiento ya cuenta con raíces lo suficientemente fuertes, la temperatura necesaria será cálida alrededor de los 28 °C. En la etapa de floración y polinización algunas plantas se polinizan solas dejando caer polen de las

anteras al estigma y otras se polinizan de forma cruzada cuando los insectos se cruzan de una flor a otra, según Samuel Contreras, del Departamento de Ciencias Vegetales de la Pontificia Universidad Católica de Chile, este tipo de polinización ocurre entre el 8 y 37 % de las veces. En la etapa de fructificación se produce la fertilización del ovario de la flor dando como resultado el fruto. Una vez producida la fertilización del ovario se convierte en un pericarpio carnoso, el pericarpio del pimiento son las paredes comestibles del ovario. [25]

La planta de pimiento es herbácea de tallo erecto y ramificado donde alcanza una altura entre el 0.5 y 1 metro, la raíz se puede considerar pivotante, con hojas ovales, alargadas verde – oscuras con bordes enteros. Las flores de este tipo de hortaliza son más bien solitarias donde solo en muy pocas ocasiones se pueden encontrar en grupos de 2 o 3. En la parte del cáliz tiende a tener una forma enredada provista de 4 o 5 sépalos verdes unidos entre sí. La taxonomía del pimiento se puede analizar en la tabla 1. [23]

**Tabla 1.** Taxonomía del pimiento

<b>Reino</b>	<b>Vegetal</b>
<b>Clase</b>	Angiospermae
<b>Subclase</b>	Dicotyledoneae
<b>Orden</b>	Tubiflorae
<b>Familia</b>	Solanaceae
<b>Género</b>	Capsicum
<b>Especie</b>	Annum Millar

**Fuente:** [23]

### **Variedades**

Variedades de pimiento que se cultivan bajo invernadero.

#### **Canario**

Pimiento que puede ser cultivado tanto dentro como fuera del invernadero, tiene un ciclo precoz. Se trata de un pimiento híbrido con 3-4 lóbulos, el tamaño del fruto está entre los 10 y 14 cm. Tiene un color verde oscuro, piel lisa, pesa alrededor de los 370g, con una carne espesa de 9mm. Esta planta se caracteriza por ser fuerte y vigorosa con una buena cantidad de follaje que la cubre. Consta una buena capacidad de resistencia al HR TMV (virus el Mosaico del Tabaco). Este pimiento cuenta con una gran acogida en el mercado fresco e industrial. [26]

#### **Martha**

La variedad de pimiento Martha tiene un ciclo de alrededor de los 130 días, y brinda la posibilidad de ser cosechado en coloración verde o roja, su peso oscila entre los 160 y 200 g, su piel es lisa, además de ser una variedad bastante fuerte, por lo que se le puede cosechar en cualquier época del año, presentando una gran resistencia a la marchitez por Phytophthora. [27]

#### **Alazan**

Esta variedad de pimiento puede ser cultivada tanto dentro como fuera de un invernadero, su ciclo de crecimiento es rápido. Se considera un pimiento híbrido de 3 o 4 lóbulos. Las características del fruto son 8 cm de longitud por 9 cm de diámetro, su color es verde que termina siendo un anaranjado luciente, su peso está alrededor de los 300 g. Es una planta que cuenta con un vigor medio alto siendo adecuada para un mercado fresco. [26]

#### **Kaiman F1**

El pimiento Kaiman F1 es una variedad que se cosecha de preferencia bajo invernadero, donde tiene una alta producción, con un ciclo de producción largo, es considerado un híbrido con excelente cuaje. Su fruto tiene una forma rectangular y uniforme presentando una buena calidad, siendo resistente a enfermedades como por ejemplo el virus del bronceado del tomate. [27]

#### **Red Madona**

El pimiento red madona consiste en un pimiento clásico, muy vigoroso, con una muy buena cobertura foliar, con unas paredes gruesas. El tiempo de cosecha de esta planta es prolongado, superando el año de cosecha con una madurez relativa intermedia. Cuenta con 3 o 4 lóbulos en su gran mayoría, el tamaño de la fruta suele ser grande o extragrande, con una forma alargada de color rojo y aspecto liso. El pimiento red madona cuenta con una buena resistividad a bacterias y virus. [26]

#### **Itamara F1**

El pimiento Itamara F1 es una variedad vigorosa, de gran porte y compacta. Cuenta con una buena resistencia al Virus del Bronceado del tomate (TSWV). El pimiento tiene un color verde que termina siendo rojo en la maduración, con una piel lisa. [27]

### **CLXPHSA4 F1**

Este tipo de pimiento cuenta con una vegetación equilibrada y altos rendimientos. Es una variedad vigorosa que soporta las bajas temperaturas adaptada a los ciclos largos. Sus frutos son homogéneos, de color verde oscuro, de piel lisa y alargada. El pimiento CLXPHSA4 F1 tiene una gran resistencia al Virus del Mosaico del Tabaco (TMV). [26]

### **14PE9681**

El pimiento 14PE9681 cuenta con una capacidad de cuaje bastante alta, así como una alta resistencia al virus del moteado suave del pimiento. Los frutos de este tipo de pimiento son uniformes, con un color verde intenso, además su cultivo es abundante, conservando un buen tamaño en el transcurso de su ciclo. [27]

### **Morfología**

La morfología está encargada de estudiar la estructura y forma de los seres vivos, grupo al cual pertenecen las plantas. Esta ciencia tiene por objetivo describir la funcionalidad y características de las diferentes partes que conforman la planta. [28]

Según Chiriboga Jhonny [26] las características morfológicas principales son:

#### **Sistema radicular.**

Las raíces del pimiento son pivotantes y profundas, donde depende mucho del tipo de suelo donde son plantadas, muchas raíces crecen de forma horizontal alcanzando los 50 cm y 1 m.

#### **Tallo**

El tallo se caracteriza por tener un crecimiento limitado y fuerte alcanzando entre dos o tres ramificaciones. Además, todo su ciclo emitirá un cierto número de brotes y hojas.

#### **Hoja.**

El pimiento se caracteriza por tener una hoja lampiña, entera y lanceolada, con un ápice muy pronunciado y un peciolo largo y con poco aparente.

**Planta**

Herbácea perenne con un ciclo de cultivo cercano al año con una altura característica entre los 0.5 metros hasta los 2 m dependiendo la variedad de pimiento y si el cultivo se realiza dentro o fuera de un invernadero.

**Flor**

Las flores van apareciendo en cada nudo de tallo, son de tamaño pequeño y de color blanco la polinización es auto gama.

**Fruto**

El fruto tiene varios colores dependiendo la variedad puede ser verde rojo amarillo, naranja, violeta o blanco algunas variedades van pasando de verde a rojo o anaranjado o amarillo cuando se encuentran en estado de madurez. Su peso depende de la variedad, pero normalmente se encuentran alrededor de los 500 g.

**Semillas**

Las semillas se encuentran dentro de una placenta cónica, son de color amarillo pálido ligeramente Rainford mes, con una forma redondeada y una longitud variable entre los tres y 5 cm.

**1.4.11 Internet de las cosas (IoT)**

El Internet de las cosas tiene sus raíces en el trabajo realizado por Mark Weiser para Xerox PARC en la década de 1990, donde no estudió el futuro de las redes que conectarían las redes entre sí, sino lo que pasaría cuando la tecnología fuese tan barata que sería incorporada a los objetos del día a día. [1]

El término “Internet de las cosas” (en inglés, Internet of Things, abreviado IoT) hace referencia a cualquier dispositivo u objeto cotidiano que cuenta con la capacidad de estar conectado a internet y tener cierto tipo de inteligencia. La figura 9 hace una ilustración a este tipo de tecnología.

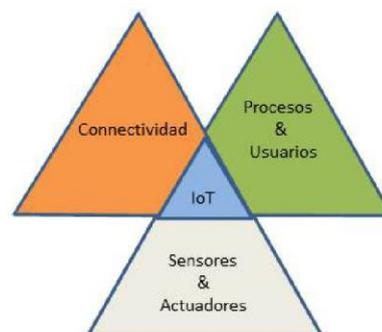


**Figura 9.** *IoT, Internet de las cosas*

**Fuente:** [29]

IoT permite que los objetos puedan conectarse con otros sistemas o personas obteniendo información útil para desempeñar una acción específica. Lo que sugiere la tecnología IoT es que, en lugar de tener un conjunto pequeño de equipos informáticos altamente potentes, se podría tener un conjunto grande de dispositivos periféricos con poca potencia. Los objetos inteligentes conectados requieren la suficiente autonomía para alimentar un sistema que analice la información y la envíe a otros sistemas inalámbricamente. El sistema recibe estímulos de diferentes tipos constantemente, para posteriormente convertirlos en datos que serán procesados y enviados. El dispositivo tiene la función de conectarse a un objeto real y generar respuestas por medio de actuadores. [1]

La tecnología IoT puede ser considerada como una combinación de sensores y actuadores los cuales cuentan con la capacidad de proporcionar y recibir información digital y colocarla en redes bidireccionales que son capaces de transmitir dicha información para que pueda ser procesada, analizada y utilizada por diferentes servicios y usuarios finales. La figura 10 representa los conceptos por los que está formada la tecnología IoT. [30]



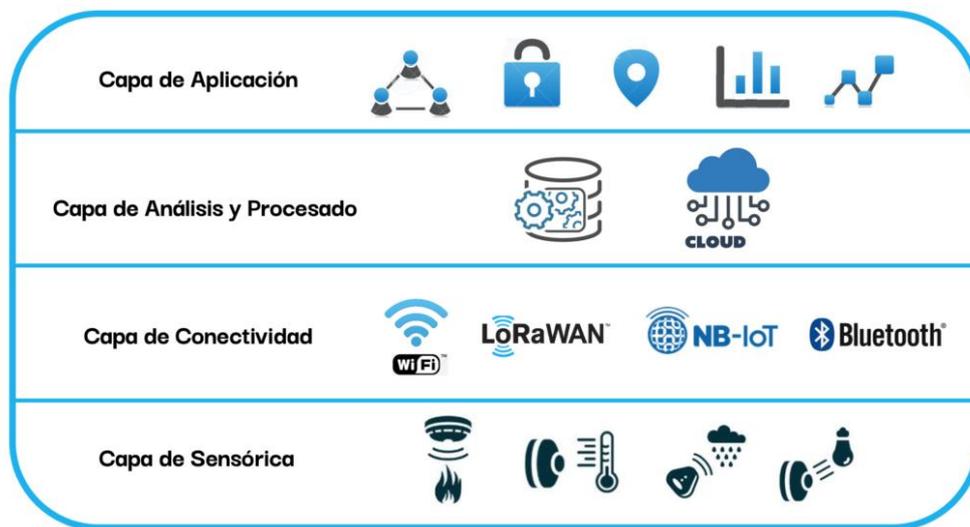
**Figura 10.** *Concepto de IoT*

**Fuente:** [30]

### 1.4.11.1 Arquitectura IoT

Un requisito fundamental de la tecnología IoT consiste en que los objetos de la red deben estar conectados entre sí, ya que dicha tecnología realiza el trabajo de conectar el mundo físico con el mundo virtual. La arquitectura de la tecnología IoT debe contar con un diseño el cual tenga en cuenta la escalabilidad, la extensibilidad y la operatividad entre dispositivos. Debido a que las cosas se pueden mover la arquitectura debe ser adaptable para que los dispositivos interactúen entre ellos de forma dinámica y admitan una comunicación adecuada en tiempo real. [31]

En la figura 11 se puede observar que la arquitectura de la tecnología IoT está formada por cuatro capas, la capa sensorica, la capa de conectividad, la capa de análisis y procesado y por último la capa de aplicación. [32]



*Figura 11. Arquitectura IoT*

*Fuente: [32]*

La función que realiza cada capa es la siguiente:

**Capa sensorica:** Consiste en la capa que reúne a todos los dispositivos del sistema los cuales son capaces de realizar medidas de ciertas magnitudes físicas o parámetros. Estos dispositivos también conocidos como sensores son capaces de detectar automáticamente el entorno e intercambiar datos entre dispositivos, Cada 1 de los objetos o sensores posee una identidad digital propia y puede ser rastreado de forma fácil y rápida. [31]

**Capa de conectividad:** la función de esta capa consiste en recibir la información útil en forma de señales digitales desde la capa sensorica y transmitirla hacia la capa de análisis y procesado haciendo uso de medios de transmisión cómo WiFi, Bluetooth,

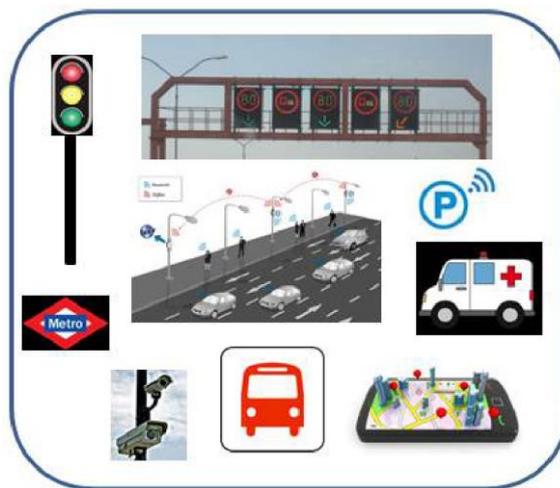
LoRaWAN, NB-IoT, WiMax, Zigbee, GSM, 3G, entre otros, utilizando protocolos como MQTT, IPv4, IPv6, DDS, entre otros. [33]

Capa de análisis y procesado: esta capa se encarga del procesamiento de la información adquirida en la capa sensorica, se encarga de filtrar los datos no deseados e integrar la información principal en conocimiento que pueda ser útil para los servicios que se desean realizar y para los usuarios finales. [30]

Capa de aplicación o Interfaz de usuario: esta capa conecta los dispositivos con los usuarios habilitando su administración, permitiendo integrar, monitorear, organizar y gestionar de manera remota los dispositivos, además de ofrecer características específicas para mantener la conectividad, el estado y la seguridad del sistema durante su etapa de trabajo. [32]

#### 1.4.11.2 Aplicaciones de la tecnología IoT:

El número de aplicaciones y servicios en los cuales se puede incorporar la tecnología IoT puede llegar a ser muy limitado, ya que se puede adaptar a muchos campos de la actividad humana, llegando a facilitar y mejorar la calidad de vida de muchas maneras. Algunas de las aplicaciones en las cuales se puede utilizar los servicios de la tecnología IoT son las que se observan en la figura 12 y se detallan a continuación:



*Figura 12. Aplicaciones de la tecnología IoT.*

*Fuente: [30]*

- Edificios inteligentes conectados.
- Ciudades inteligentes y transporte.
- Educación.
- Electrónica de consumo.

- Salud.
- Control de tráfico.
- Agricultura y medio ambiente.
- Servicios de energía.
- Conectividad inteligente.
- Fabricación.
- Compras.

#### **1.4.12 Tecnología LoRa**

Long Range o más conocido como LoRa consiste en un tipo de modulación de espectro ensanchado (FSK) y una variación del espectro extendido chirp (CSS), donde se mantiene las características de bajo consumo de potencia de FSK y se incrementa el área de comunicación. Como su nombre lo indica los dispositivos inalámbricos LoRa son de largo alcance, pero con un bajo consumo de potencia, lo que les permite ser utilizados por las redes IoT. [34]

La tecnología Lora y su protocolo abierto LoRaWAN admiten la aplicación de IoT resolviendo desafíos mundiales de hoy en día como: regulación de la contaminación, uso eficiente de energía y recursos naturales, protección de desastres, entre otros. Con la existencia a nivel global de más de 158 millones de terminales conectados a internet en 92 países, se está formando un planeta más eficiente gracias a los terminales LoRa. [5]

##### **1.4.12.1 Modulación LoRa**

La modulación LoRa consiste en un tipo de modulación de espectro expandido que hace uso de la técnica denominada chirp, la cual modula el mensaje a través de una señal que constantemente presentan variaciones en su frecuencia. Por otro lado, cuenta con un amplio rango de cobertura puesto que sus señales son robustas, las cuales resisten a ruidos e interferencias. Este tipo de modulación también se utiliza en el área militar y espacial debido a sus características para comunicar a largas distancias y su capacidad de resistir y evadir las interferencias. [35]

##### **1.4.12.2 Bandas de frecuencia LoRa**

Ecuador se rige a las normas establecidas por la ARCOTEL (agencia de regulación y

control de las telecomunicaciones), la cual se rige a las normas internacionales establecidas por la UIT, donde se establecen las normas para la utilización de las bandas ISM (Bandas de radio industriales, científicas y médicas), o lo que es lo mismo se establece el uso libre del espectro de radiofrecuencia para los campos industriales, científicos y médicos. [36]

La tecnología LoRa puede trabajar en varios rangos de frecuencia en diferentes regiones del mundo. La banda de frecuencia utilizada en USA, Canadá, Singapur, Australia o Israel es la banda ISM de 902-928 MHz, utilizando de 13 canales con ancho de banda de 2.16 MHz por canal. [34] En la tabla 2 se puede observar las Bandas ICM reguladas por la Arcotel.

**Tabla 2.** Bandas ICM reguladas por la Arcotel

<b>Bandas</b>	<b>Frecuencia Central</b>	<b>Intensidad de campo</b>
<b>6765 – 6795 KHz</b>	6780 KHz	80 – 100 (dBuV/m)
<b>13553 – 13567 KHz</b>	13560 KHz	80 – 120 (dBuV/m)
<b>26957 – 27283 KHz</b>	27120 KHz	70 – 120 (dBuV/m)
<b>40,66 – 40,70 MHz</b>	40,68 MHz	60 – 120 (dBuV/m)
<b>902 – 928 MHz</b>	915 MHz	60 – 120 (dBuV/m)
<b>2400 – 2500 MHz</b>	24,50 MHz	30 – 120 (dBuV/m)

*Fuente:* [37]

#### **1.4.12.3 Arquitectura de una red LoRa**

La arquitectura con la que trabaja una red LoRa es de topología tipo estrella, lo que permite conservar el tiempo de vida útil de las baterías ya que en este tipo de topología los nodos están conectados con los Gateway o puertas de enlace y no con todos los demás dispositivos, con ello lo que se crean son nodos sencillos los cuales cuentan con un largo alcance de comunicación y bajo consumo de potencia sin reducir la cobertura de la red. La complejidad de la red se encuentra en el servidor el cual es el encargado de administrar la red y filtrar los paquetes de datos redundante recibido, además de encargarse de la seguridad de la red. [38]

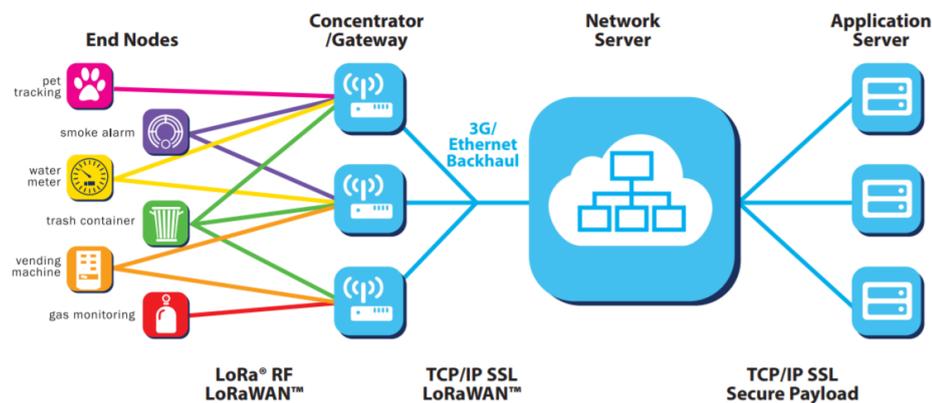
#### **1.4.13 Tecnología LoRaWAN**

La tecnología LoRaWAN se define como un protocolo MAC (Media Access Control), consiste en un protocolo de red de área amplia de baja potencia creada con el fin de comunicar de manera inalámbrica cosas que se encuentren separadas a larga distancia

las cuales se encuentran conectadas a internet en redes regionales nacionales o globales. LoRaWAN cuenta con un protocolo que permite utilizar funciones para comunicaciones bidireccionales móviles, de bajo costo y seguras para IoT, máquina a máquina (M2M), aplicaciones industriales y ciudades inteligentes. [39]

#### 1.4.13.1 Arquitectura LoRaWAN

La arquitectura de una red LoRaWAN tiene cuatro partes importantes las cuales se pueden observar en la figura 13:



*Figura 13. Arquitectura LoRaWAN*

*Fuente:* [40]

Las cuatro partes importantes de una red LoRaWAN son [41]:

- Dispositivos finales o nodos: se trata de los elementos que transmiten la información a través de ondas de radio, estos deben ser de bajo consumo.
- concentrador o Gateway: consisten en las antenas que reciben la información de los nodos y la reenvían hacia el servidor a través de una comunicación TCP/IP.
- servidor de red: este envía la información hacia una aplicación final haciendo uso de una API.
- Aplicación: se trata del software que le permite al usuario obtener la información.

Los nodos pueden unirse y comunicarse a la red LoRaWAN por medio de 2 métodos diferentes [41]:

- ABP (Activation by Personalization). En este método las claves de cifrado se

cargan previamente en los dispositivos, lo que provoca una aceleración en la conexión, pero incrementa los riesgos.

- OTAA (Over-the-Air-Activation). En este método las claves de cifrado se negocian entre nodos y servidor al momento de conectarse, por lo que es del método más y el que más se recomienda utilizar, pero a su vez es más lento que el método ABP. Los parámetros de configuración necesaria en el método OTAA son:

- ✓ AppEUI: único identificador de la aplicación consta de 64 bits.
- ✓ AppKey: clave de seguridad AES de 128 bits entre nodo y red.
- ✓ DevEUI: identificador de fábrica del dispositivo.

#### 1.4.14 Protocolo MQTT

Message Queue Telemetry Transport (MQTT) consiste en un protocolo definido como machine to machine (M2M) basado en un protocolo de mensajería del tipo publicación/suscripción. Se trata de un protocolo de código abierto, lo que significa que cualquier persona puede acceder a él, lo que hace que sea relativamente fácil implementarlo en los dispositivos IoT. [42]

Uno de los puntos fuertes del protocolo MQTT consiste en que es un protocolo simple y ligero, motivo por el cual es utilizado en muchas investigaciones, por su bajo consumo de ancho de banda, baja latencia y menor consumo de energía. [2]

El protocolo MQTT cuenta con 13 distintos paquetes de control con los que ejecuta todo el proceso de comunicación: subscribe, puback, pubrel, connect, suback, publish, unsubsack, pingresp, connack, disconnect, pubcomp, pingreq y unsubscribe. Cada uno de ellos contiene tres partes tal y como se observa en la tabla 3:

**Tabla 3.** *Formato común de los paquetes de control*

Cabecera fija presente en todos los paquetes de control MQTT
Cabecera variable presente en algunos paquetes de control MQTT
Carga útil presente en algunos paquetes de control MQTT

**Fuente:** [43]

#### **1.4.14.1 Conceptos básicos de MQTT**

Algunos de los conceptos básicos del protocolo MQTT son [43]:

**Publicar/suscribir:** En este protocolo el editor realiza publicaciones de mensajes donde los usuarios se suscriben a temas concretos relacionados con ellos y reciben todos los mensajes que se publican en estos temas. Además, los clientes tienen la capacidad de publicar mensajes a los temas de esta forma se permite a todos los suscriptores acceder a los mensajes de estos temas.

**Temas y suscripciones:** Los editores publica un mensaje en temas que pueden considerarse el asunto del mensaje. el suscriptor se suscribe solo en los temas que le interese para obtener mensajes específicos.

**Niveles de calidad de servicio:** El protocolo MQTT incorpora calidades de servicio (QoS) con el fin de obtener confirmación en la entrega de paquetes, e incorpora funcionalidades como retain message (mensaje retenido). Soporta tre niveles de QoS que se describe en continuación:

**QoS0 (Como máximo una vez):** El mensaje se envía como máximo una vez y no se garantiza la entrega del mensaje.

**QoS1 (al menos una vez):** Los datos se envían al menos una vez y es posible entregar un mensaje más de una vez estableciendo el valor del indicador de duplicado en 1.

**QoS2 (exactamente una vez):** El mensaje se envía exactamente una vez utilizando handshaking de 4 vías.

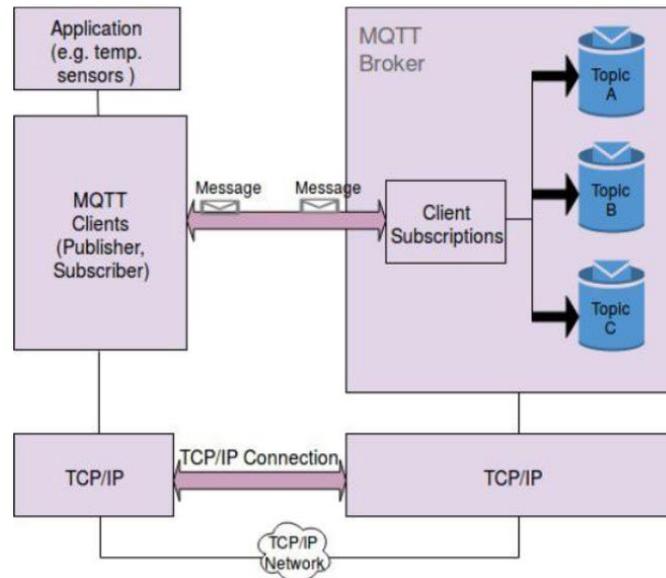
**Mensajes retenidos:** En el protocolo MQTT los mensajes son retenidos en él broker después de distribuirlos a todos los clientes presentes. Cuando se adquiere otra membresía para el tema idéntico es cuando los mensajes retenidos de esos temas c envían al nuevo cliente.

**Sesiones limpias y conexiones fiables:** cuando un abonado es asociado con el bróker, la asociación de sesión limpia es considerada permanente en el caso de que su valor sea falso.

**Testamentos o mensajes:** Un cliente puede informar al broker que contiene un mensaje que debe ser distribuido a un tema o temas en específico.

#### **1.4.14.2 Arquitectura del protocolo MQTT**

La arquitectura de protocolo MQTT más común se puede dividir en dos partes principales como se observa en la figura 14:



**Figura 14. Arquitectura MQTT**

**Fuente:** [43]:

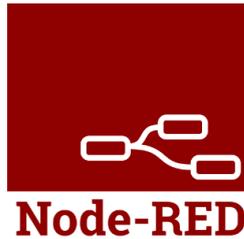
Las dos partes más importantes de la arquitectura MQTT son el cliente y el broker, la función que realiza cada uno es [43]:

**Cliente:** Puede ser un publicador o un suscriptor y es el encargado de establecer la conexión de red con el servidor (bróker). es capaz de realizar la publicación de mensajes para los usuarios interesados, suscribirse a un tema de su interés para recibir mensajes, darse de baja de los temas suscritos o separarse de bróker.

**Broker:** Se encarga de controlar la distribución de los datos, pero principalmente de recibir todos los mensajes del editor filtrarlo, analizar quiénes están interesados en ellos y en Villar los mensajes a todos los clientes suscritos. El broker realiza las funciones de aceptar las solicitudes de los clientes, Receptan los mensajes que publican los usuarios, procesa diferentes tipos de peticiones como suscribirse y darse de baja de los usuarios, una vez recibidos los mensajes del editor los selecciona y los envía a los usuarios interesados.

#### **1.4.15 Node-RED**

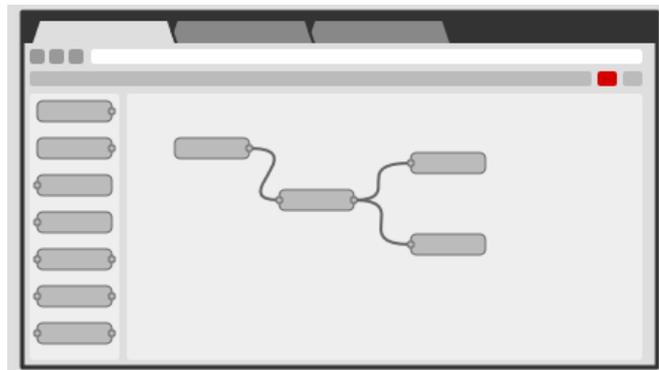
Node-RED consiste en un software de programación visual el cual tiene la capacidad de conectar dispositivos hardware, API y servicios en línea. está basado en javascript, desarrollado en la plataforma Node.js y además utiliza de la mejor manera su modelo sin bloqueo basado en eventos. [44]



*Figura 15. Logo de Node-RED*

*Fuente:* [45]

El software Node-RED contienen nodos los cuales se representan a través de iconos, estos cuentan con diferentes funciones como por ejemplo monitoreo de flujos, permitiendo a los desarrolladores o programadores conectar nodos de entrada, nodos de salida, así como nodos de procesamiento para crear flujos en los cuales puedan ser procesados los datos. [44] En la figura 16 se observa un ejemplo de edición de flujo.



*Figura 16. Edición de flujo basada en navegador*

*Fuente:* [45]

Las tres partes más importantes que componen el software Node-RED son:

Panel de nodos

Panel de flujo

Panel de información y depuración.

#### **1.4.15.1 Ventajas de Node-RED**

Node-RED consiste en una herramienta FBP (Programación en base al flujo), enfocada en realizar el control de datos para aplicaciones web e IoT. las ventajas de este software son [46]:

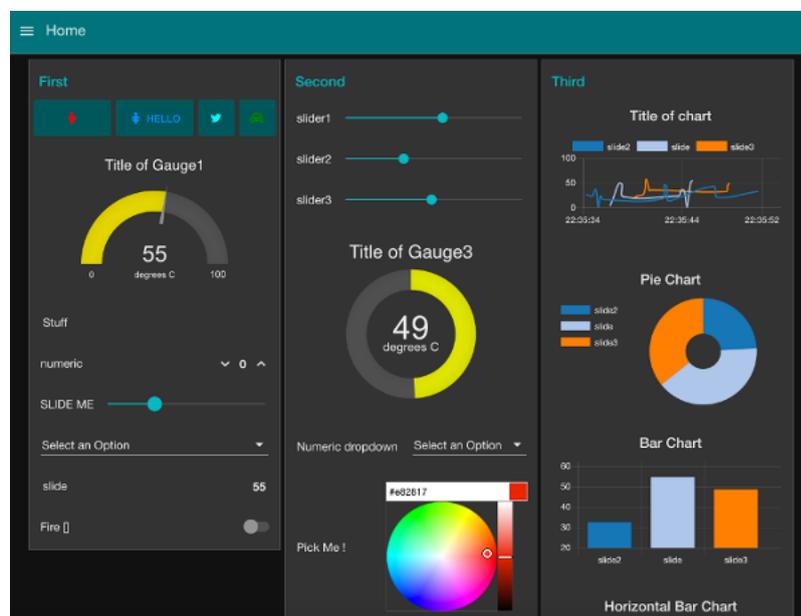
**Simplificación:** Hacer un tipo de programación gráfico su complejidad se reduce ya que la codificación se elimina y la programación se completa de manera intuitiva.

**Eficiencia:** el FBP tipificado por Node-RED Se puede completar prácticamente solo

con operaciones que se encuentran en la interfaz gráfica de usuario, el editor de flujo De este software desarrolla la construcción del entorno de ejecución para la aplicación, el entorno de desarrollo integrado o más conocido como IDE, la sincronización de bibliotecas, así como la preparación del editor para que el usuario o programado tenga la posibilidad de enfocarse en el desarrollo.

#### 1.4.15.2 Dashboard de Node-RED

El Dashboard es una interfaz, figura 17, donde se presenta los datos en tiempo real, donde el usuario es capaz de observar de forma gráfica los diferentes datos, de manera ordenada y agrupada para que la visualización sea didáctica y fácil de comprender [47].



*Figura 17. Dashboard de Node-RED*

*Fuente: [47]*

#### 1.4.16 Sensores y actuadores

Un sensor se define como un dispositivo de entrada que provee una salida manipulable de la variable física medida. El concepto de sensor está relacionado con el concepto de un transductor ya que los sensores hacen uso de los transductores, pero con la diferencia que el sensor no solo cambia el dominio de la variable física medida, sino que además la salida del sensor es un dato útil para un sistema de medición. Es decir, los sensores imitan la capacidad de percepción de los seres humanos de detectar diferentes fenómenos. [48]



**Figura 18.** *Diferentes tipos de sensores*

**Fuente:** [49]

En la figura 18 se observan diferentes tipos de sensores. Los sensores se pueden clasificar de diferentes modos, las clasificaciones más comunes son por el principio de transducción utilizado o por el tipo de variable a medir. Clasificación de los sensores por el principio de transducción es: fotoeléctrico, capacitivo, químico, piezoresistivo, termoeléctrico, piezoeléctrico, magnético, ultrasónico. clasificación por el tipo de variable a medir es: de corriente, de posición, velocidad y aceleración, biométricos, de nivel y proximidad, de gas y pH, de humedad y temperatura, de color, radiación y visión, de fuerza y deformación, de flujo y presión. [50]

Se define como actuador a un dispositivo que cuenta con la capacidad de generar una fuerza que provoca un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía. En la figura 19 se observan algunos ejemplos de actuadores. Normalmente los actuadores se clasifican en dos grupos, por el tipo de energía utilizada, grupo en el que se encuentran los actuadores hidráulicos, neumáticos y eléctricos, y por el tipo de movimiento que generan, grupo en el que se encuentran los actuadores rotatorios y lineales. [50]



**Figura 19.** *Ejemplo de actuadores*

**Fuente:** [51]

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Implementar un sistema automatizado de control y monitoreo para el cultivo de hortalizas bajo invernadero basado en tecnología Lora y MQTT.

### **1.5.2 Objetivos Específicos**

Analizar la situación actual del proceso de cultivo de hortalizas bajo invernadero para determinar las necesidades existentes para incrementar la producción.

Determinar los dispositivos, equipos y tecnología necesaria para desarrollar un sistema automatizado de control y monitoreo para el cultivo de hortalizas bajo invernadero basado en tecnología Lora y MQTT.

Desarrollar un sistema automatizado para el cultivo de hortalizas bajo invernadero capaz de obtener datos en tiempo real y controlar el invernadero de forma remota.

## **CAPÍTULO II**

### **METODOLOGÍA**

#### **2.1 Materiales**

Para el desarrollo e implementación del proyecto de investigación se utilizan diferentes materiales como: libros, revistas, papers, artículos de revista, tesis, revistas de tecnología, páginas web de tecnologías, libros especializados en el cultivo de hortalizas, artículos científicos acerca del cultivo de hortalizas y pimiento, así como un invernadero de cultivo de pimiento con un tiempo de producción de 6 meses, para probar que el sistema funciona correctamente. Además de todo esto se utilizan diferentes elementos electrónicos y eléctricos para poder implementar dicho sistema.

#### **2.2 Métodos**

##### **2.2.1 Modalidad de Investigación**

El presente proyecto de investigación implementó la investigación aplicada, ya que su principal objetivo es poner en práctica todos los conocimientos y destrezas adquiridas en el área de comunicación inalámbrica, electrónica y programación, durante toda la formación académica.

El tipo de investigación es bibliográfica ya que su propósito es recolectar información confiable relacionados con el tema de monitoreo y control de diferentes parámetros como temperatura, humedad, pH o luminosidad dentro de invernaderos. Los documentos en los cuales se basa el proyecto de investigación son revistas, tesis, papers, proyectos de investigación, libros, páginas web oficiales y verificadas, de donde se extrae información relevante la cual pueda ser utilizada en el desarrollo del proyecto.

A lo largo del desarrollo del proyecto de investigación se realizaron una serie de pruebas de funcionamiento donde se pudo notar una serie de errores que puede llegar a tener un sistema de control y monitoreo de cultivos de hortalizas, De igual forma se

realizaron diferentes tipos de pruebas para determinar que la comunicación inalámbrica funcione de la manera esperada.

Finalmente, en el proyecto de investigación realizado también se utilizó la investigación de campo puesto que fue necesario adquirir datos directamente del invernadero a través de los diferentes sensores.

### **2.2.2 Recolección de Información**

Para la recolección de información para el presente proyecto de titulación se tomó en cuenta diferentes fuentes de información previamente verificadas como libros, artículos académicos, revistas, tesis o proyectos de investigación de bases de datos de repositorios universitarios. La información recolectada está directamente relacionada con el control y monitoreo de diferentes parámetros utilizando comunicación inalámbrica, así como información relacionada al cultivo de hortalizas bajo invernadero.

### **2.2.3 Procesamiento y Análisis de Datos**

Para el procesamiento y análisis de datos se sigue los pasos mencionados a continuación:

Análisis de información recolectada en fuentes bibliográficas

Determinar de los diferentes parámetros a medir dentro de un invernadero de hortalizas los cuales influyen directamente en la producción

Determinar los diferentes elementos electrónicos y eléctricos utilizados para el control y monitoreo de un invernadero de cultivo de hortalizas

Planteamiento de la propuesta de solución

### **2.2.4 Desarrollo del proyecto**

Para el desarrollo del proyecto de siguen una serie de pasos que se describen a continuación:

Identificación de las condiciones adecuadas para el cultivo de hortalizas.

Identificación de los materiales, equipos y tecnologías necesarios para desarrollar un sistema automatizado de control y monitoreo basado en tecnología LoRaWAN y

MQTT para el cultivo de hortalizas bajo invernadero.

Elaboración de una tabla comparativa de los diferentes tipos y marcas de hardware y software necesarios para la elaboración del proyecto.

Definición de las características de materiales, equipos y tecnologías necesarias para el proyecto.

- Instalación de sensores y actuadores.
- Adquisición y procesamiento de datos.
- Diseño de una interfaz para el control y monitoreo.
- Comunicación inalámbrica del sistema automatizado de control y monitoreo.
- Generar alertas automáticas al celular.
- Pruebas de funcionamiento del sistema automatizado de control y monitoreo para el cultivo de hortalizas.
- Elaboración de informe final del proyecto de investigación.

## **CAPÍTULO III**

### **RESULTADOS Y DISCUSIÓN**

En el presente capítulo se realiza el desarrollo del sistema propuesto, describiendo las partes que lo conforman, los dispositivos y elementos electrónicos utilizados, el diseño, la programación, entre otros. Además, se presentan los resultados obtenidos mediante las pruebas realizadas con el sistema.

Como caso de estudio se utiliza un cultivo de pimiento el cual ya tiene un año y dos meses, por lo que ya se encuentra en su última etapa de producción.

#### **3.1 Análisis y Discusión de los Resultados**

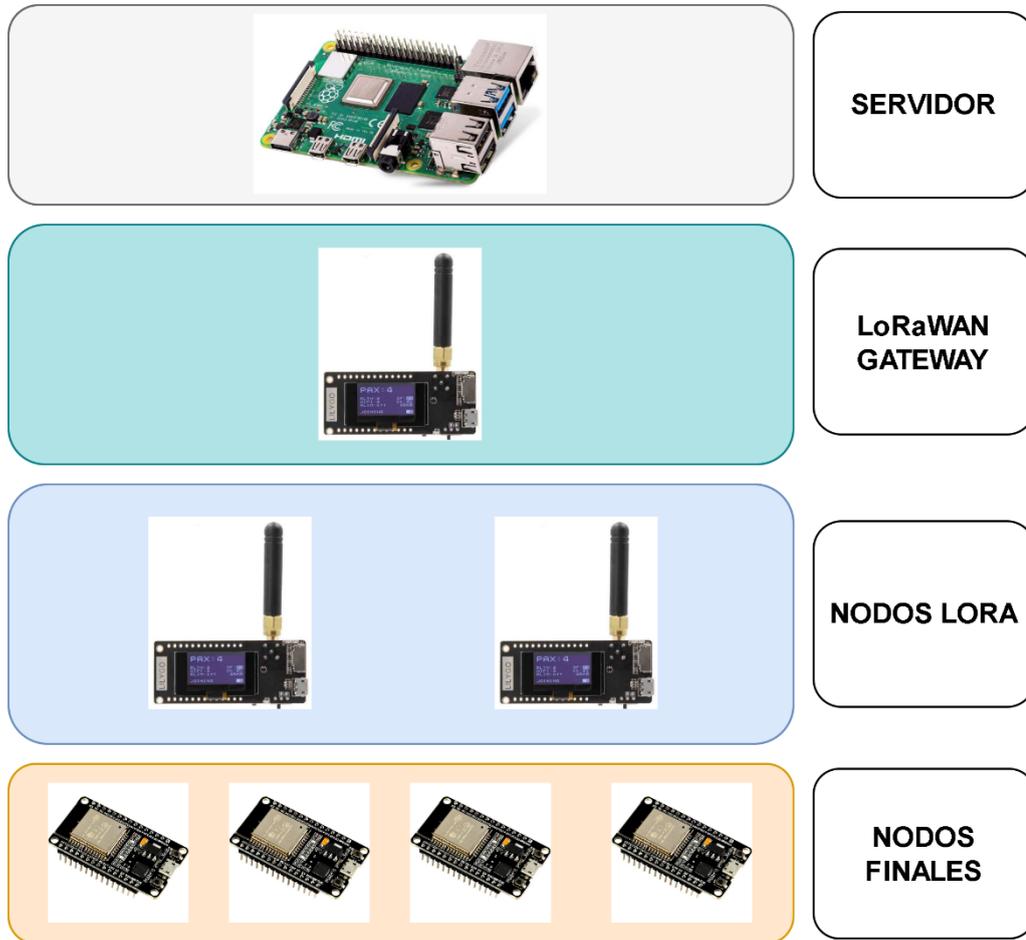
La implementación de un sistema de control y monitoreo para el cultivo de pimiento bajo invernadero permite al agricultor llevar sus cultivos de forma tecnificada, teniendo un conocimiento exacto de lo que ocurre con su plantación, donde el sistema tomará decisiones acordes a las condiciones atmosféricas registradas a través de los sensores en el interior del invernadero. El sistema propuesto pretende medir variables de temperatura, humedad, pH y luminosidad en el interior del invernadero. Con ello es posible brindar a las plantas un ambiente y unas condiciones adecuadas para su crecimiento y desarrollo. Para el sistema propuesto se pretende utilizar tecnología LoRa debido a las ventajas como su largo alcance, duración de batería, instalación simple, bajo coste, tecnología abierta, de confianza, movilidad global, bidireccional, localización y seguridad.

Para el desarrollo del sistema se emplea software y hardware de uso libre, utilizando elementos y equipos electrónicos y de fácil adquisición y que cuya disponibilidad sea alta e inmediata en el mercado, con el fin de asegurar que el sistema sea fácilmente reparable y mejorable en cualquier momento.

##### **3.1.1 Análisis de los principales elementos del sistema**

El sistema está basado en una arquitectura conocida como maestro-esclavo, donde se encuentran nodos finales enviando los datos recopilados hacia un servidor donde el usuario final puede observar y analizar la información, pasando por un Gateway formando de esta forma una red LoraWAN. En consecuencia, se puede definir cuatro

elementos principales para el funcionamiento del prototipo, el cual se puede apreciar en la figura 20.



*Figura 20. Sistema de control y monitoreo*

*Fuente: El investigador*

**Nodos finales.** – Estos están formados por los microcontroladores colocados directamente en el invernadero juntamente con los sensores, se divide en cuatro etapas como son las de adquisición, procesamiento y transmisión de datos.

Adquisición de datos: en esta etapa los sensores realizan el papel principal puesto que son los encargados de recolectar información sobre las variables del entorno que se pretenden medir, para que posteriormente se transmitan hacia el servidor donde se procesarán y almacenarán. En esta etapa las señales eléctricas (analógicas) son acondicionadas para que el microcontrolador a través de su conversor ADC cuantifique los datos para su posterior procesamiento.

Procesamiento de datos: en esta etapa se realiza el procesamiento de la información a través de fórmulas y métodos matemáticos en el microcontrolador para que

posteriormente se visualice en una pantalla LCD.

Transmisión de datos: en esta etapa se realiza el proceso de envío de información a través del microcontrolador hacia los nodos LoRa mediante una comunicación inalámbrica utilizando una frecuencia de 2,4 GHz utilizando el estándar 802.11 y el protocolo MQTT.

**Nodos LoRa.** – En la red del sistema se forman dos nodos LoRa para los cuales se utilizan tarjetas de desarrollo basadas en ESP32 las cuales incorporan un módulo LoRa 868/915 MHz, las tarjetas mencionadas permiten al usuario trabajar con datos en forma bidireccional a distancias considerables, y a través de su pantalla OLED permite la visualización de información evitando el típico cableado para integrar un display. Por medio de estas tarjetas se enviará la información a largas distancias a través de un Gateway. La frecuencia utilizada es de 915MHz.

**LoRaWAN Gateway.** – Los gateways son los encargados de conectar toda la red, es decir son los que crean la comunicación entre los nodos finales y el servidor permitiendo enviar y recibir información.

**Servidor de Red LoRaWAN.** – El servidor se encarga de recibir la información del Gateway y transmitirla hacia los dispositivos finales, este caso dispositivos móviles o computadoras. Para el servidor se utiliza en este caso un microordenador conocido como Raspberry Pi-4.

### **3.1.2 Desarrollo de la propuesta**

#### **3.1.2.1 Condiciones adecuadas para el cultivo de hortalizas.**

Uno de los factores más importantes en el cultivo de hortalizas es la temperatura, se considera que la temperatura es un limitante para la dispersión natural de las especies vegetales. Las hortalizas como cualquier otro organismo necesitan de unas condiciones adecuadas para que su desarrollo o crecimiento sea adecuado, las temperaturas cardinales son: [18]

- Mínima: temperatura bajo la cual el crecimiento se detiene.
- Óptima: temperatura en la cual el crecimiento es más rápido.
- Máxima: temperatura sobre la cual el crecimiento se detiene.

Las temperaturas cardinales son diferentes para cada cultivo y varían dependiendo la zona, época y método de cultivo. Por ello las hortalizas se pueden clasificar dependiendo de la temperatura en la que se desarrollan: hortalizas de estación cálida o verano y hortalizas de estación fría o de invierno. [18]

### **Hortalizas de estación fría**

Grupo A: hortalizas que poseen temperaturas óptimas de desarrollo entre 15 y 18°C. No toleran temperaturas promedio mayores a 24°C, así como heladas suaves. En este grupo se encuentran hortalizas como: haba y espinaca. [18]

Grupo B: se diferencian con el grupo anterior en que son susceptibles a heladas cerca de su madurez. En este grupo se encuentran hortalizas como: lechuga, escarola, apio, perejil, hinojo, acelga y achicoria. [18]

Grupo C: Estas hortalizas se adaptan a temperaturas entre 13 y 24°C y soportan las heladas por lo cual no se cultivan bajo invernadero. En este grupo se encuentran hortalizas como: cebollino, ajo, puerro, cebolla, chalota. [18]

### **Hortalizas de estación cálida**

Grupo D: Las hortalizas pertenecientes a este grupo se adaptan a temperaturas entre 18 y 27°C y no soportan heladas. En este grupo se encuentran hortalizas como: pimiento, melón, zapallo, tomate, pepino y poroto para chaucha. [18]

Grupo E: Este grupo se diferencia del anterior en que las temperaturas idóneas están sobre los 21°C. En este grupo se encuentran hortalizas como: sandía y berenjena. [18]

### **Parámetros climáticos adecuados para el cultivo de pimiento**

En base a lo expuesto se realiza la tabla 4 con los parámetros que se van a monitorear en el presente proyecto de investigación:

**Tabla 4.** *Parámetros meteorológicos para el pimiento*

	<b>Mínima</b>	<b>Ideal</b>	<b>Máxima</b>
<b>Temperatura ambiente</b>	18°C	28°C	35°C
<b>Humedad aire</b>	50%	65%	75%
<b>Humedad suelo</b>	50%	60%	70%
<b>pH</b>	5,2	6,0	6,5

**Fuente:** [52]

### 3.1.2.2 Identificación de los materiales, equipos y tecnologías

#### 3.1.2.2.1 Requerimientos del sistema

Para el diseño del proyecto es necesario tomar en cuenta los requerimientos del sistema de control y monitoreo inalámbrico, determinando las necesidades tanto de hardware como de software, así como las condiciones físicas que debe haber en el interior del invernadero.

#### Requerimientos de hardware

Para el desarrollo e implementación del sistema de control y monitoreo se requieren una serie de elementos electrónicos y eléctricos que se mencionan en la tabla 5.

**Tabla 5.** Elementos principales de hardware para el desarrollo del prototipo.

Dispositivo	Descripción	Características técnicas
<b>Microordenador</b>	Consiste en un ordenador de bajo costo y un tamaño reducido utilizado para el desarrollo de prototipos a baja escala.	4 GB de memoria RAM Procesador basado en x64 2.4 GHz y 5.0 GHz IEEE 802. LAN inalámbrica 11B/g/n/ac.
<b>Placas de desarrollo</b>	Placa de desarrollo ESP32 para la adquisición de señales y control de actuadores.	E/S analógicas E/S digitales Alimentación de entre 5 y 12 Vdc. Corriente de alimentación de 0.5 a 2A.
<b>Módulos de comunicación inalámbrica LoRa</b>	Módulo de comunicación inalámbrica que utilice la tecnología LoRa para el envío y recepción de datos.	Comunicación LoRa entre los dos nodos y Gateway.
<b>Sensores</b>	Sensores para la adquisición de datos de las variables tales como: temperatura, humedad, pH, luminosidad.	El sensor de temperatura debe medir temperaturas entre 0 y 50 grados centígrados. El sensor de humedad de ambiente debe medir entre 0 y 100%. El sensor de humedad de suelo debe medir entre 0 y 100%. El sensor de pH debe medir entre el rango de 0 a 14Ph. Voltaje de alimentación entre 5 a 12 Vdc.
<b>Actuadores</b>	Los actuadores para controlar son: ventiladores, bomba de agua, motor eléctrico, emisores de luz.	Voltaje de alimentación 110 o 220 Vac. Corriente de alimentación hasta 8A.
<b>Elementos de maniobra y control</b>	Elementos que permiten control el sistema de forma manual tales como pulsadores, selectores, relés, breakers.	Pulsadores que trabajen con corrientes de hasta 12A. Selectores que trabajen con corrientes de hasta 12A. Relés que trabajen con corrientes de hasta 12A. Breakers que trabajen con corrientes de hasta 12A.
<b>Cableado de conexión</b>	Cableado para conexiones eléctricas.	Cable que trabaje hasta con 20A. Cable que trabaje hasta con 15A.
<b>Fuentes de alimentación</b>	Fuente de alimentación para alimentar las diferentes placas de los dos nodos.	Voltaje de salida de 5Vdc y 24Vdc.

*Elaborado por: El investigador*

## Requerimientos de software

Además del hardware mencionado, es necesario contar con un software para el prototipo, el cual permita procesar información, así como ejecutar diferentes tipos de tareas y enviar órdenes hacia los actuadores del sistema, a continuación, en la tabla 6 se detallan los requerimientos principales para desarrollar el sistema.

*Tabla 6. Requerimientos de software para el sistema.*

Componente	Descripción	Características técnicas
<b>Software de programación</b>	Permite el desarrollo o creación de programas que cumplan con los requerimientos del sistema.	Acceso libre sin licencias.
<b>Servidor</b>	Almacena distribuye y suministra información, se basan en un modelo cliente servidor, donde el cliente solicita información al servidor y este se la suministra si es que está autorizado.	Sistema operativo de código abierto. Alojamiento de página web. Servidor de base de datos. Lenguaje de programación dinámico para páginas web.
<b>Base de datos</b>	Consiste en una agrupación de información que se almacena electrónicamente y puede ser buscada de manera fácil y rápida.	Código abierto Vistas personalizables Procesamiento de datos
<b>Aplicación de mensajería instantánea</b>	Aplicación de mensajería instantánea para aparatos electrónicos como computadoras o smartphones que funciona con internet.	Peso reducido en el dispositivo móvil. Almacenamiento ilimitado. Creación de bots.

*Elaborado por: El investigador*

### 3.1.2.2.2 Situación previa a la implementación del sistema en el invernadero

Antes de comenzar con el diseño del sistema de control y monitoreo para el invernadero es necesario realizar un estudio previo de la situación en la que se encuentra el invernadero. Tras analizar cómo trabaja el invernadero hasta el momento se puede verificar que dentro del invernadero en los días soleados alcanza temperaturas excesivas llegando a rozar los 50 °C debido a que no tiene la suficiente entrada ni salida de aire, es decir la ventilación es inadecuada para el cultivo de hortalizas pudiendo observarse que cuenta con cortinas con una apertura muy limitada como se observa en la figura 21, además cuenta con unas salidas de aire en el techo, figura 22, las cuales no superan los 20 cm, por tanto la planta llega a un nivel de estrés excesivo durante todo su tiempo de vida.



**Figura 21.** Cortinas de ventilación con las que cuenta el invernadero

**Fuente:** El investigador



**Figura 22.** Aberturas de ventilación en la parte superior del invernadero

**Fuente:** El investigador

Por otro lado, se puede observar en la figura 23 y 24 que el riego del invernadero se realiza a través de mangueras de goteo utilizando una bomba de agua, pero al no tener un control de la humedad del suelo, se riega cuando el encargado del cuidado de la plantación cree que es necesario, por tanto, el suelo tiene una humedad fuera del rango indicado para la correcta producción, este es otro factor que hace que la planta no rinda al cien por cien de su capacidad.



(1)



(2)

**Figura 23.** (1) y (2) Sistema de mangueras para el riego por goteo

**Fuente:** El investigador



(1)



(2)

**Figura 24.** (1) Bomba de agua para el sistema de riego.  
(2) Tanque reservorio para el riego del invernadero

**Fuente:** El investigador

Como se aprecia en las imágenes anteriores el propietario del invernadero ya cuenta con una bomba de agua por lo tanto para el presente proyecto se hará uso de esta. A continuación, en la tabla 7, se detallan las características técnicas de la bomba:

**Tabla 7.** Bombas de agua

Bomba de agua del propietario	
Ilustración	
Marca	PAOLO
Caudal (L/min)	220
Aplicaciones	Uso industrial y agrícola. Camaroneras, bananeras, etc. Bombeo de fuentes de agua.
Diámetro (pulgadas)	1 1/2
Elevación (m)	18
Modelo de fabrica	CP-2004
Máxima potencia (HP)	1
Tensión (voltios)	110/220
Corriente	8/16
Ciclos (hz/rpm)	60-3450
Precio (\$)	279,08

**Fuente:** [53]

Por otra parte, el invernadero no cuenta con monitoreo del pH del agua. Por lo que la implementación de este control permite al productor saber si la tierra o agua tienen el

pH necesario para que los productos químicos u orgánicos provoquen el efecto esperado en las plantas al momento de la fertilización o fumigación.

En ocasiones, en la zona en donde se está implementando el sistema hay días muy nublados y las plantas no llegan a recibir toda la cantidad de radiación necesaria para que la fotosíntesis se realice correctamente provocando una reducción en la producción. Por tanto, después de realizar un estudio en el invernadero, el productor solamente cuenta con cortinas de apertura manual, las cuales no son suficientes para ventilar el invernadero dando como resultado temperaturas inadecuadas. El invernadero cuenta también con sistema de riego por goteo, el cual no es controlado técnicamente. Todos estos factores influyen en que el invernadero no trabaje al 100% de su capacidad. Con el sistema de monitoreo, se puede incrementar la producción y evitar que la planta experimente niveles de estrés elevados.

### **3.1.2.2.3 Materiales equipos y tecnologías necesarios para desarrollar el sistema**

Para el desarrollo del proyecto de investigación se necesitan diferentes tipos de elementos de hardware y software los cuales se describen a continuación.

#### **Hardware**

Para la implementación del sistema se requieren diferentes elementos electrónicos, eléctricos y mecánicos. En este apartado se divide el sistema en cuatro etapas, como monitoreo y control de temperatura, monitoreo y control de humedad, monitoreo del pH y monitoreo de luminosidad.

#### **Etapas de monitoreo y control de temperatura**

Para monitorear la temperatura en el interior del invernadero es necesaria la instalación de sensores de temperatura de ambiente, con el fin de obtener datos reales, se decide implementar cuatro sensores de temperatura de ambiente ubicados en zonas estratégicas del invernadero.

Para el control de la temperatura se instalaron 2 ventiladores, para mejorar el sistema de ventilación del invernadero reajustando el sistema de cortinas para lograr una mayor apertura y ventilación dentro del invernadero para reducir su temperatura. Para lo cual, se requiere primero la instalación de los ventiladores, un motor eléctrico de bajas revoluciones y la adaptación de un nuevo sistema de apertura y cierre de cortinas.

Los materiales y equipos necesarios en la etapa de control y monitoreo de la temperatura se detallan en la tabla 8:

**Tabla 8. Materiales, equipos y herramientas para la etapa de monitoreo y control de temperatura**

Material	Descripción	Cantidad
<b>Sensores de temperatura</b>	Sensores con capacidad de medir temperaturas entre los 0 °C y los 50 °C.	4
<b>Ventiladores de altas revoluciones</b>	Ventiladores con capacidad de funcionamiento a 220Vac	2
<b>Motor de bajas revoluciones y un torque de nivel medio-bajo</b>	Motor con capacidad de funcionamiento a 110 o 220Vac y que pueda elevar al menos 50 kg	1
<b>Tubos (para enrollar las cortinas)</b>	Tubo de al menos 6 m de largo y 3 mm de grosor y ancho de 3/4"	5
<b>Varillas</b>	Varilla de 10mm de grosor	5
<b>Rulimanes</b>	Rulimanes de 1.5"	8
<b>Tubo G</b>	Tubo G de 2"	6
<b>Pegamento de contacto</b>	Cemento de contacto (por litros)	2
<b>Discos de corte</b>	Discos de corte para amoladora	2
<b>Electrodos</b>	Electrodos para suelda ligera (por libras)	1.5
<b>Pernos y tuercas</b>	pernos y tuercas para unir los tubos	9
<b>Chumaceras</b>	Para tubos de 3/4"	10

*Elaborado por: El investigador*

Además de herramientas como soldadura, taladro, amoladora, martillo, desarmadores, chaveta, entre otros. Y elementos de protección como guantes, careta o cascos de reducción de ruido.

### **Etapa de monitoreo y control de humedad**

Para el monitoreo y control de la humedad se divide en dos partes el monitoreo y control de humedad en el suelo y Por otra parte en monitoreo de la humedad en el ambiente. Antes de implementar el sistema el propietario del invernadero cuenta solamente con un sistema de riego por goteo poco tecnificado al no contar con sensores de humedad, por ende, para el control de la humedad tecnificado es necesario la instalación de sensores para el suelo, además de un sistema de aspersión para incrementar la humedad en el interior del invernadero cuando así lo requiera, en base a lo mencionado se describe a continuación los materiales necesarios para esta etapa. Debido a que el sistema debe ser controlado de forma remota es necesario a su vez la instalación de electroválvulas tanto para el sistema de riego como para el sistema de microaspersión. Los materiales para la etapa mencionada se detallan en la tabla 9.

**Tabla 9. Materiales para la etapa de monitoreo y control de humedad**

Material	Descripción	Cantidad
<b>Sensores de humedad de ambiente</b>	Sensores que midan una humedad entre 0 y el 100%	4
<b>Sensores de humedad de suelo</b>	Sensores que midan una humedad entre 0 y el 100%	4
<b>Electroválvulas</b>	Electroválvulas que funcionen a 12 Vac y sean de 1" y 1.5"	3
<b>Manguera para sistema de microaspersión</b>	Manguera para conectar al sistema de riego general (por metros)	90
<b>Mangueras para nebulizadores</b>	Manguera de 0.5" (por metros)	10
<b>Nebulizadores</b>	Nebulizadores para manguera de 0.5"	5
<b>Acoples</b>	Para conectar las mangueras a los nebulizadores, así como al resto de mangueras	10
<b>Teflón</b>	para sellar las uniones	1

*Elaborado por: El investigador*

### **Etapa de monitoreo del pH**

Para la etapa de monitoreo del pH se utilizará un sensor el cual estará mostrando constantemente un valor de pH entre 0 y 14 pH. Cabe recalcar que el pH que se va a monitorear es el del agua. Los materiales a utilizar se detallan en la tabla 10.

**Tabla 10. Materiales para el monitoreo de pH.**

Material	Descripción	Cantidad
<b>Sensor de pH del agua</b>	Sensor de pH de 0 a 14pH	1

*Elaborado por: El investigador*

### **Etapa de monitoreo de iluminación**

En la etapa de monitoreo de la iluminación se pretende ayudar a que la plantación reciba la suficiente energía proveniente de la radiación para que pueda trabajar correctamente durante el día, llegando así a su máximo nivel de producción, ya que la cantidad de radiación que recibe está directamente relacionada al proceso denominado fotosíntesis que realizan las plantas, cuánto mejor sea el proceso de fotosíntesis y mayor número de horas permanezca realizándolo sin llegar a estresar a la planta la producción incrementará.

En base a lo mencionado, se pretende instalar dos pequeños paneles solares para medir la radiación recibida por las plantas en vatios hora y poder procesar esa información para encender la iluminación artificial.

Por tanto, los materiales que se pretenden utilizar en esta etapa son los que se observan

en la tabla 11:

**Tabla 11. Materiales para la etapa de monitoreo de iluminación**

<b>Material</b>	<b>Descripción</b>	<b>Cantidad</b>
<b>Cable para conexión eléctrica</b>	Cable de cobre de calibre 14 que soporte al menos 15 amperios (por metros)	200
<b>Boquillas para foco</b>	Boquillas para 17 focos, que trabajen a 220Vac	17
<b>Focos</b>	Focos de alta potencia, al menos de 60W	17
<b>Cinta adhesiva (Tape)</b>	Cinta adhesiva para empalmar los cables y distintas conexiones	3
<b>Panel solar</b>	Paneles solares que hagan la función de sensores	2
<b>Cable de red</b>	Cable para conexiones con bajo flujo de corriente (por metros)	5

**Elaborado por:** *El investigador*

Las cuatro etapas mencionadas forman las cuatro áreas en las cuales va a trabajar el sistema de control y monitoreo con el fin de incrementar la producción del invernadero, pero es necesario mencionar que estas cuatro etapas deben trabajar juntamente con el resto de los equipos y elementos de la etapa de control, es decir los tableros de control donde se encuentran los diferentes dispositivos que harán que el sistema trabaje de forma inalámbrica. Cabe mencionar que se pretende instalar dos tableros, el primero estará instalado fuera del invernadero en la zona donde el propietario del invernadero tiene instalado la bomba de agua y el segundo tablero se pretende instalar en el interior del invernadero. Por tanto, a continuación, se mencionan los diferentes elementos y dispositivos necesarios para los tableros de control.

### **Tablero de control del interior del invernadero**

El tablero de control que se pretende instalar dentro del invernadero almacenara en el interior a cuatro ESP32, relés, módulo Lora entre otros elementos para poder controlar los actuadores colocados en el interior del invernadero. A continuación, en la tabla 12, se detallan los materiales para dicho tablero.

**Tabla 12. Materiales para el tablero de control del interior del invernadero**

<b>Material</b>	<b>Descripción</b>	<b>Cantidad</b>
<b>Tablero eléctrico</b>	Tablero eléctrico de 30x40	1
<b>Rieles</b>	Riel para contactores o relés (por metros)	1
<b>Cable</b>	Cable de cobre, calibre 14 (por metros)	3

<b>Cinta adhesiva (tape)</b>	Cinta adhesiva para empalmar cables	1
<b>Convertidor</b>	Transformador de 220Vac a 12Vac	1
<b>Relés de 10A</b>	Que soporten al menos 10A	2
<b>Relés de 5A</b>	Que soporten al menos 5A	3
<b>Fuente</b>	Trabaje con 3A y convierta 220Vac en 12Vdc	1
<b>Placa para nodo sensor</b>	Microcontrolador de 32-bits, con comunicación inalámbrica Wifi	4
<b>Placa para módulo Lora</b>	Módulo que trabaja con tecnología IoT a grandes distancias	1
<b>Caja para placa nodo sensor</b>	Caja de plástico para almacenar la placa del nodo sensor	4
<b>Caja para placa módulo Lora</b>	Caja de plástico para almacenar la placa del módulo LoRa	1

*Elaborado por: El investigador*

### **Tablero de control fuera del invernadero**

El tablero de control que se encuentra fuera del invernadero está enfocado en controlar la bomba de agua, así como tener un control manual del riego cuando el propietario así lo requiera, además cuenta con un paro de emergencia. dentro de este tablero se encuentran 3 relés, una fuente, dos transformadores, así como la placa del módulo Lora. A continuación, en la tabla 13, se especifica de mejor manera todos los materiales necesarios para dicho tablero.

*Tabla 13. Materiales para el tablero de control fuera del invernadero*

<b>Material</b>	<b>Descripción</b>	<b>Cantidad</b>
<b>Tablero eléctrico</b>	Tablero eléctrico de 30x30	1
<b>Rieles</b>	Riel para contactores o relés (por metros)	1
<b>Cable</b>	Cable de cobre, calibre 14 (por metros)	3
<b>Cinta adhesiva (tape)</b>	Cinta adhesiva para empalmar cables	1
<b>Convertidor</b>	Transformador de 220Vac a 12Vac	2
<b>Relés de 10A</b>	Que soporten al menos 10A	1
<b>Relés de 5A</b>	Que soporten al menos 5A	2
<b>Fuente</b>	Trabaje con 3A y convierta 220Vac en 12Vdc	1
<b>Placa para módulo Lora</b>	Módulo que trabaja con tecnología IoT a grandes distancias	1
<b>Caja para placa módulo Lora</b>	Caja de plástico para almacenar la placa del módulo LoRa	1
<b>Breaker</b>	Interruptor automático que corte el paso de corriente de 2 fases (220Vac)	1

<b>Breaker</b>	Interruptor automático que corte el paso de corriente de 1 fase (110Vac)	1
<b>Sensor de corriente</b>	Medidor de corriente consumida	2
<b>Sensor de voltaje</b>	Medidor de voltaje consumido	1

*Elaborado por: El investigador*

### **Placas del módulo nodo sensor**

Las placas de los nodos sensores realizan la función de tomar los datos de los sensores ubicados en el interior del invernadero, al dividir el invernadero en cuatro secciones para un mejor manejo también se utilizan cuatro nodos sensor, por tanto, si se pretende instalar cuatro sensores de humedad del suelo, cuatro sensores de temperatura de ambiente y cuatro sensores de humedad del aire cada uno de estos cuatro sensores ira enlazado a cada uno de estos cuatro módulos nodo sensor. A continuación, en la tabla 14, se menciona los materiales necesarios para desarrollar las placas de los módulos nodo sensor, cabe recalcar que las cuatro placas van a ser iguales.

*Tabla 14. Materiales para la placa del nodo sensor*

<b>Material</b>	<b>Descripción</b>	<b>Cantidad</b>
<b>Tarjeta de desarrollo</b>	E/S analógicas. E/S digitales. Alimentación de entre 5 y 12 Vdc. Corriente de alimentación de 0.5 a 2A.	1
<b>MOC</b>	Conexión entre elementos electrónicos y triacs de potencia para trabajar a 110Vac	2
<b>TRIAC</b>	Controlador de carga en corriente alterna	2
<b>Diodos</b>	Permitir el paso de corriente en un solo sentido	2
<b>Dip switch</b>	4 vías	1
<b>Potenciómetro de precisión</b>	Resistencia variable	2
<b>Amplificador operacional</b>	Amplificar la señal sin afectar a otras variables	1
<b>Borneras</b>	Regleta de conexión aislada (varias vías)	5
<b>Espadines</b>	Conexión de elementos (por regleta)	4
<b>Baquelita electrónica</b>	Base para conexión de elementos electrónicos	1
<b>Estaño</b>	Material para soldar elementos electrónicos (por metro)	0.25
<b>Ácido para baquelita</b>	Corroer cobre sobrante de baquelita	1

*Elaborado por: El investigador*

Como se mencionó anteriormente se pretende fabricar cuatro placas para los módulos del nodo sensor, la tabla anterior se realiza en base a una placa, para obtener la lista de materiales final dicha tabla se debe multiplicar por cuatro en todos los materiales mencionados.

### **Placas del módulo LoRa**

Las placas de los módulos Lora tienen como función conectarse a las ESP32 y poder conectarse al Gateway Lora, con esto se forma la red LoRaWAN, contando con dos nodos un Gateway y un servidor. A continuación, en la tabla 15, se describe los materiales para desarrollar las placas de los nodos Lora.

**Tabla 15. Materiales modulo Lora**

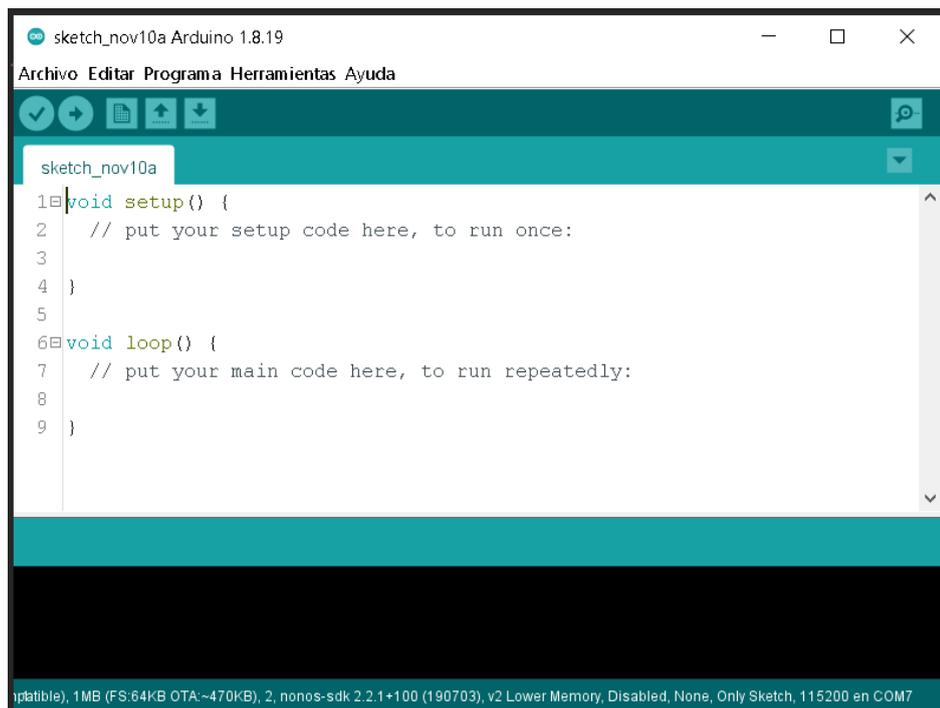
<b>Material</b>	<b>Descripción</b>	<b>Cantidad</b>
<b>Tarjeta de desarrollo LoRa</b>	E/S analógicas. E/S digitales. Alimentación de entre 5 y 12 Vdc. Corriente de alimentación de 0.5 a 2A.	1
<b>MOC</b>	Conexión entre elementos electrónicos y triacs de potencia para trabajar a 110Vac	4
<b>TRIAC</b>	Controlador de carga en corriente alterna	4
<b>Diodos</b>	Permitir el paso de corriente en un solo sentido	4
<b>Potenciómetro de precisión</b>	Resistencia variable	4
<b>Amplificador operacional</b>	Amplificar la señal sin afectar a otras variables	2
<b>Borneras</b>	Regleta de conexión aislada (varias vías)	3
<b>Espadines</b>	Conexión de elementos (por regleta)	3
<b>Baquelita electrónica</b>	Base para conexión de elementos electrónicos	1
<b>Estaño</b>	Material para soldar elementos electrónicos (por metro)	0.25
<b>Ácido para baquelita</b>	Corroer cobre sobrante de baquelita	1

*Elaborado por: El Investigador*

### **Software**

Para el desarrollo del sistema de control y monitoreo es necesario contar con software de programación que sea de licencia libre o gratuita, después de haber analizado las diferentes opciones se escogen dos softwares de programación, los cuales se definen a continuación.

**Arduino IDE:** consiste en un software de programación de código abierto, es decir, es un entorno de desarrollo integral formado por un conjunto de herramientas de programación, puede utilizarse solo con un lenguaje de programación o con varios. Dentro del programa el usuario puede editar el código, compilarlo, depurarlo y cargarlo, además de tener un constructor de interfaz gráfica. Los programas que se realizan en Arduino IDE están compuestos de un solo fichero el cual tiene una extensión “.ino”, por otro lado, también cuenta con un monitor serial el cual le permite al usuario realizar las pruebas necesarias.



**Figura 25.** *Arduino IDE*

**Fuente:** *El investigador*

**Node-RED:** consiste en una herramienta potente de programación visual que se implementa en dispositivos controladores de hardware. Se caracteriza por mostrar de forma visual las relaciones y funciones de manera que se pueda programar sin la necesidad de escribir. Dentro de su panel se pueden conectar nodos que se comuniquen entre ellos y puede instalarse también diferentes equipos como por ejemplo computadoras o servidores en la nube. Esta forma de programar puede facilitar la visualización del flujo de la información.



<b>Corriente de CC para 3.3V Pin</b>	-	40 mA	150 mA
<b>Memoria flash</b>	4 MB	4 MB	48 KB
<b>SRAM</b>	160 KB	520 KB	6 KB
<b>EEPROM</b>	512 bytes	512 bytes	256 bytes
<b>Velocidad de reloj</b>	80Mhz (hasta 160 Mhz)	160 MHz (hasta 240 MHz)	20 MHz
<b>Longitud</b>	48 mm	52 mm	69 mm
<b>Ancho</b>	26 mm	31 mm	53 mm
<b>WIFI</b>	sí	sí	sí
<b>Bluetooth</b>	no	sí	no
<b>Sensor táctil</b>	no	10	no
<b>CAN</b>	no	sí	no
<b>Interfaz MAC Ethernet</b>	no	sí	no
<b>Sensor de temperatura</b>	no	sí	no
<b>Sensor de efecto de pared</b>	no	sí	no
<b>Toma de corriente</b>	no	no	sí
<b>conexión USB</b>	sí	sí	sí
<b>Precio</b>	\$12,00	\$12,50	\$ 68,00

*Elaborado por: El investigador*

Después de haber realizado la tabla comparativa de tres módulos WiFi los cuales podrían servir para el proyecto se puede apreciar diferencias entre cada 1 de ellos por tanto se selecciona el módulo ESP32 el cual cuenta como una serie de ventajas respecto a los otros dos módulos wifi por ejemplo la velocidad a la que trabaja es 240 MHz, la SRAM con la que cuenta es de 512KB, cuenta con dos núcleos, el consumo de corriente en sueño profundo es de 2.5  $\mu$ A, y por último el número de pines de entrada analógica es de 15. Por estas características y ventajas respecto a los otros dos módulos se selecciona el módulo ESP32.

*Tabla 17. Comparación entre módulos LoRa*

	<b>ESP32 LoRa - SX1276 868MHz-915MHz</b>	<b>Modelo Lora32 V2 Lilygo 868/915MHz</b>	<b>LILYGO® TTGO T- Beam V0.7 ESP32 868/915Mhz</b>
<b>Ilustración</b>			
<b>Maestro chip</b>	ESP32	ESP32 PICO-D4	TTGO T-Beam V0.7 ESP32
<b>Voltaje de funcionamiento</b>	3,3 - 7V	1,8 - 3,7V	1,8 - 3,7V

<b>Pantalla</b>	OLED 0,96 pulgadas	OLED 0,96 pulgadas	NO
<b>Corriente de transmisión</b>	120mA @ + 20dBm	120mA @ + 20dBm 90mA @ + 17dBm 29mA @ + 13dBm	120mA @ + 20dBm 90mA @ + 17dBm 29mA @ + 13dBm
<b>Distancia de comunicación</b>	2.6 Km (área abierta)		
<b>Flash</b>	4MB	4MB	4MB
<b>Frecuencia de funcionamiento</b>	868MHz-915MHz	868M/915MHz	433/470MHz (versión 433HMz) /868M/915M (versión 868MHz)
<b>Potencia de transmisión:</b>	19.5dBm @ 11b, 16.5dBm @ 11g, 15.5dBm @ 11n	+20dBm	+20dBm
<b>Sensibilidad de recepción</b>	-139DBM (SF12, 125KHZ)	-139dBm @ LoRa & 62,5 Khz & SF = 12 & 146bps  -136dBm @ LoRa & 125Khz & SF = 12 & 293bps  -118dBm @ LoRa & 125Khz & SF = 6 & 9380bps  -123dBm @ FSK & 5Khz & 1,2 Kbps	-139dBm@LoRa &62.5Khz&SF=12&146b ps -136dBm@LoRa &125Khz&SF=12&293b ps -118dBm@LoRa &125Khz&SF=6&9380b ps - 123dBm@FSK&5Khz&1 .2Kbps
<b>Velocidad de datos</b>	150Mbps @ 11n HT40, 72Mbps @ 11n HT20, 54Mbps @ 11g, 11Mbps @ 11b	1,2 K ~ 300Kbps @ FSK	1.2K~300Kbps@FSK 0.018K~37.5Kbps@LoRa
<b>Modo de modulación</b>	FSK, GFSK	FSK, GFSK, MSK, GMSK, LoRa TM, OOK	FSK,GFSK,MSK,GMSK, LoRa TM, OOK
<b>Micro SD</b>	NO	SI	NO
<b>Corriente de sueño</b>	0.2 µA	0.2 µA	0.2uA@SLEEP 1.5uA@IDLE
<b>Temperatura de funcionamiento</b>	-40°C hasta +90°C	-40°C hasta + 85 °C	-40°C hasta + 85 °C
<b>Corrección de frecuencia</b>	NO	Automática	Automática
<b>Control de ganancia</b>	NO	Automático	Automático
<b>Activación y salto de frecuencia</b>	NO	Rápido	Rápido

<b>Manejador de paquetes de datos</b>	NO	Altamente configurable	Altamente configurable
<b>Antena</b>	Arduino Smart	SMA	SMA
<b>PRECIO</b>	\$51.19	\$45,00	\$81.39

*Elaborado por: El investigador*

Después de haber realizado la tabla 17 donde se compara 3 módulos Lora se selecciona el Modelo Lora32 V2 Lilygo 868/915MHZ debido a las diferencias y ventajas que presenta con respecto a los otros dos módulos una de las ventajas es la corriente de transmisión y su respectiva ganancia la potencia de transmisión, la sensibilidad de recepción, así como la velocidad de datos. Es posible apreciar que el módulo seleccionado presenta características técnicas similares al módulo LILYGO® TTGO T-Beam V0.7 ESP32 868/915Mhz, pero se puede apreciar una gran diferencia en el precio, por tanto, en base a lo mencionado se escogió el módulo Modelo Lora32 V2 Lilygo 868/915MHZ.

*Tabla 18. Comparación entre microordenadores*

	<b>Raspberry Pi 3B</b>	<b>Raspberry Pi 3B+</b>	<b>Raspberry Pi 4B</b>
<b>Ilustración</b>			
<b>Procesador</b>	Broadcom BCM2837	Broadcom BCM2837B0	Broadcom BCM2711B0
<b>CPU</b>	1.2GHz Quad Core	1.4GHz Quad Core	1.5GHz Quad Core
<b>RAM</b>	1GB	1GB	1GB/2GB/4GB
<b>USB</b>	4 USB 2.0	4 USB 2.0	2 USB 2.0 / 2 USB 4.0
<b>Video</b>	Jack, HDMI	Jack, HDMI	Jack, 2 micro HDMI
<b>Audio</b>	4 USB 2.0	4 USB 2.0	2 USB 2.0 / USB 4.0
<b>Memoria</b>	Micro SD	Micro SD	Micro SD
<b>Wireless</b>	WiFi 2.4 GHz Bluetooth 4.1BLE	WiFi 2.4 y 5 GHz Bluetooth 4.1BLE	WiFi 2.4 G y 5 Hz Bluetooth 4.1BLE
<b>Ethernet</b>	Ethernet 10/100	Gigabit Ethernet 300	Gigabit Ethernet 1000
<b>Alimentación</b>	5V – 2.5A	5V – 2.5A	5V – 3A
<b>GPIO</b>	40 pines	40 pines	40 pines
<b>Dimensiones</b>	85x56x17 mm	85x56x17 mm	85x58x19 mm
<b>PRECIO</b>	\$155.00	\$205.00	\$305.00

*Elaborador por: El investigador*

Tras comparar 3 modelos de microordenadores en la tabla 18, se decide seleccionar la Raspberry Pi 4B con 4GB de memoria RAM ya que el procesamiento de datos debe ser rápido dentro del sistema. Además de este modelo presenta diferentes ventajas respecto a los otros como la velocidad del CPU, el procesador, los USB, así como la conectividad al wifi, debido a estas diferencias se realiza la selección.

**Tabla 19.** Comparación entre sensores de temperatura y humedad relativa

	<b>DHT11</b>	<b>DHT21</b>	<b>DHT22</b>
<b>Ilustración</b>			
<b>Alimentación</b>	3Vdc ≤ Vcc ≤ 5Vdc	3,5Vdc ≤ Vcc ≤ 5,5Vdc	3Vdc ≤ Vcc ≤ 5Vdc
<b>Tipo</b>	Humedad relativa & Temperatura	Humedad relativa & Temperatura	Humedad relativa & Temperatura
<b>Rango de medida de humedad</b>	De 20% a 90% RH	De 0% a 100% RH	De 0% a 100% RH
<b>Precisión Humedad</b>	4% RH	3% RH	2% RH
<b>Resolución Humedad</b>	1% RH	0,1% RH	0,1% RH
<b>Rango de medida de temperatura</b>	De 0 a 50 °C	De -40 a 80 °C	De -40 a 80 °C
<b>Precisión temperatura</b>	±2 °C	≤±0,5 °C	≤±0,5 °C
<b>Resolución Temperatura</b>	0,1 °C	0,1 °C	0,1 °C
<b>Señal de salida</b>	Digital	Digital	Digital
<b>Tiempo de sensado</b>	1s	2s	2s
<b>Tamaño</b>	12 x 15,5 x 5,5 mm	60 x 28 x 13mm	14 x 18 x 5,5 mm
<b>Precio</b>	\$4,00	\$9,50	\$7,00

*Elaborador por: El investigador*

En el presente proyecto no se van a medir temperaturas por debajo de los 0 °C ni tampoco por encima de los 50 °C, la humedad con la cual se va a trabajar no alcanzará valores superiores al 90%, por tanto, el sensor seleccionado es el DHT11, ya que en la tabla 19 se observa que cumple con las especificaciones necesarias y además su valor económico es inferior al resto de sensores con los cuales se lo compara.

**Tabla 20. Comparación entre sensores de humedad del suelo**

	SHT10	FC-28	FST100-2006
<b>Ilustración</b>			
<b>Alimentación</b>	5V	3 V a 5.5 V	5~24 V
<b>Tipo</b>	Temperatura y Humedad del suelo	Humedad del suelo	Humedad del suelo
<b>Rango de medida</b>	0-100%	0% - 94%	0 % de HR~100 % de HR
<b>Precisión</b>	±5%	±4%	±3%
<b>Señal de salida</b>	Digital	Analógica / Digital	Analógica
<b>Consumo</b>	30mA	35mA	167mA
<b>Tipo de comunicación</b>	Analógica	Analógica	Rs485 (protocolo Modbus)
<b>Posee sonda</b>	SI	SI	SI
<b>Tamaño</b>	49mm x 14mm	60x20x5 mm	129,5x44,5x14 mm
<b>Precio</b>	\$35,00	\$4,00	\$60,00

**Elaborador por:** El investigador

La humedad del suelo que se mide en este proyecto se encuentra en el rango del 0 y 90%. al observar la tabla 20 se puede ver que los tres sensores cumplen con el rango de medición requerido, además los tres sensores tienen una precisión similar, los tres cuentan con sonda, pero donde podemos observar una diferencia notoria es en el costo del sensor, por lo que se selecciona el sensor FC-28, ya que tiene las características necesarias para el proyecto y representa un ahorro económico considerable a las adquirir cuatro unidades. Por otro lado, la disponibilidad del sensor en el mercado es mucho mayor a comparación de los otros dos modelos comparados.

**Tabla 21. Comparación entre sensores de pH**

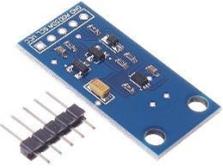
	RK500-02	SEN0161	SPR500-02
<b>Ilustración</b>			
<b>Alimentación</b>	5VDC, 12-24VDC	3.3~5.5V	5VDC, 12-24VDC
<b>Tipo</b>	pH	pH	pH
<b>Rango de medida</b>	0-14	0-14	0-14
<b>Precisión</b>	± 0.3pH	±0.1@25°C	±0.02pH
<b>Resolución</b>	0,01PH	No especificada por el fabricante	0.01pH
<b>Consumo</b>	4-20mA	10 mA	4-20 mA
<b>Protección contra agua</b>	IP68	SI	IP65

<b>Posee sonda</b>	SI	SI	SI
<b>Tipo de comunicación</b>	RS485	Analógica	RS485
<b>Tiempo de respuesta</b>	<10s	≤ 1min	<10s
<b>Precio</b>	\$138,00	\$50,00	\$300,00

*Elaborador por: El investigador*

Tras comparar las características técnicas de tres sensores de pH en la tabla 21, se seleccionó el sensor SEN0161, puesto que como se observa en la tabla anterior cuenta con características similares a los otros dos sensores con los que se comparan su rango de medición desde 0-14 pH, y además tiene protección contra el agua, uno de los factores importantes y donde destaca es en el costo indisponibilidad en el mercado.

**Tabla 22.** Comparación entre sensores de luminosidad

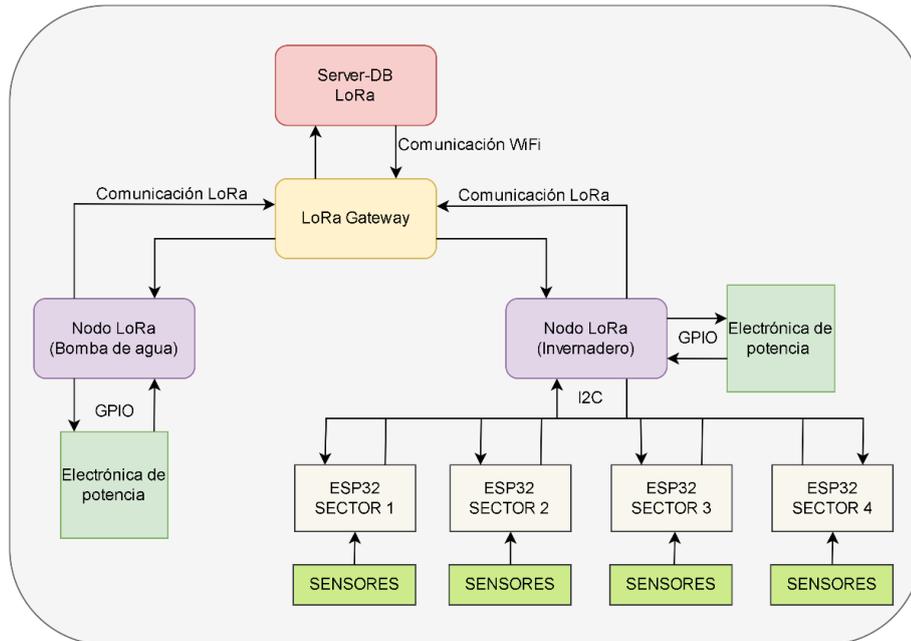
	<b>GY-30 BH1750FVI</b>	<b>LM358 ATQ6G4 43K</b>	<b>ML8511</b>
<b>Ilustración</b>			
<b>Alimentación</b>	4,5V	3-5V	3.3V-5V
<b>Tipo</b>	Sensor de luz	Sensor de luz	Sensor radiación solar
<b>Corriente de suministro</b>	120uA~190uA	0.5-3mA	300uA
<b>Fotorresistor</b>	BH1750FVI	GL5528	-
<b>Resolución</b>	0-65535 lx	0- 1023 lx	No especificada por el fabricante
<b>Tiempo de respuesta</b>	15-25 s	20-30 s	20-30 s
<b>Señal de salida</b>	Analógica	Analógica	Analógica
<b>Longitud de onda máxima</b>	No especificada por el fabricante	540 nm	280-390nm
<b>Tamaño</b>	32.6 x 15.1 mm	23,92x20,39x1,59 mm	18 x 14 x 3 mm
<b>Precio</b>	\$7,50	\$10,60	\$3,49

*Elaborador por: El investigador*

Para el presente proyecto se requiere medir la radiación que reciben las plantas en  $Wh/m^2$ , por tanto, después de comparar los 3 sensores expuestos en la tabla 22 se selecciona el sensor ML8511, ya que es un sensor creado específicamente para medir la radiación solar UV, el cual cuenta con una salida analógica que está relacionada linealmente con la intensidad UV  $mW/cm^2$ , por lo que cumple con los requerimientos necesarios para el sistema.

### 3.1.2.3 Diseño del sistema de control y monitoreo

Para realizar el diseño del sistema de control y monitoreo se procede a dividir en 3 etapas, para que el diseño sea más fácil de realizar. Las 3 etapas en las cuales se divide el sistema son adquisición de datos y control de actuadores, comunicación inalámbrica, procesamiento de datos y visualización. En la figura 27 se puede observar un esquema general del sistema.



*Figura 27. Esquema general del sistema*

*Fuente: El investigador*

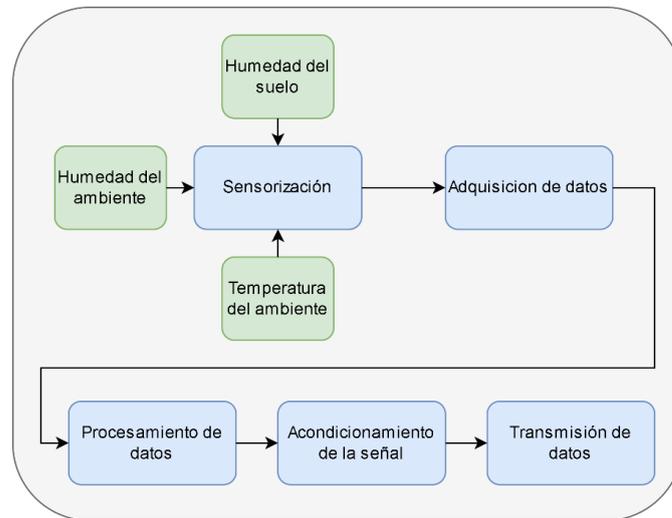
#### 3.1.2.3.1 Adquisición de datos y control de actuadores

En la adquisición de datos y control de actuadores se utiliza un módulo de desarrollo ESP32, módulo que se conectará de forma alámbrica a los sensores y de forma inalámbrica al nodo LoRa. Por tanto, el módulo ESP32 está siendo de enlace entre los sensores y el nodo Lora para que éste se conecte al Gateway del sistema, para que posteriormente los datos sean procesados y el usuario pueda tomar decisiones.

#### Nodo ESP32

El nodo ESP32 está compuesto por los sensores ubicados en cada sector, dichos sensores son: un sensor de humedad de suelo, un sensor de DHT11 (sensor de humedad y temperatura de ambiente), un módulo ESP32. El módulo ESP32 se conectará inalámbricamente vía wifi, al nodo Lora que se encuentra en el interior del

invernadero, dicho nodo LoRa se conectará inalámbricamente al Gateway Lora a través de comunicación SPI, para que éste posteriormente se conecte al servidor y poder procesar la información, visualizarla y tomar decisiones. A continuación, se puede observar en la figura 28 un diagrama del circuito de adquisición de datos.



**Figura 28.** *Nodo sensor*

**Fuente:** *El investigador*

### **Etapa de sensorización**

En la etapa de sensibilización lo que se pretende es poder medir 3 variables los cuales son temperatura del ambiente, humedad del ambiente y humedad del suelo. Para lo cual se instalan los sensores en puntos estratégicos del invernadero para poder hacer un promedio más real de dichas variables de esta manera se estará sensando ya parámetros físicos los cuales son relevantes dentro de un invernadero para que el cultivo esté en mejores condiciones.

### **Etapa de adquisición de datos**

Para la etapa de adquisición de datos se utilizan dos sensores, un sensor higrómetro FC-28 (sensor de humedad de suelo) y un sensor DHT11 el cual permite medir la temperatura y humedad del ambiente, dichos sensores cuentan con salidas digitales las cuales están conectadas a la placa del módulo ESP32.

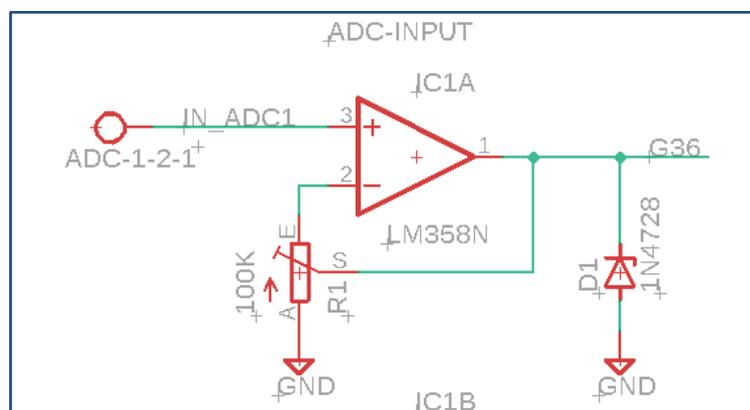
### **Etapa de procesamiento de datos**

En esta etapa se pretende adquirir y procesar los datos que envían los sensores, para el

procesamiento de estos datos se realizan algoritmos, fórmulas matemáticas, y librerías que se encuentran disponibles en Arduino IDE. La humedad del suelo se obtiene a través de una salida analógica que tiene el sensor higrómetro FC-28, los valores que se obtienen de este sensor van desde 0 sumergido en agua, hasta 1023 en aire (o en un suelo muy seco), estos valores son procesados a través de fórmulas matemáticas, para que el usuario pueda observar una humedad de suelo entre 0% y 100%. la humedad de ambiente y temperatura de ambiente es medida a través de el sensor DHT11 el cual envía una señal digital y que a través de una librería disponible en Arduino denominada DHT.h, se puede separar la señal de temperatura y humedad mediante código el void loop con las sentencia de `dht.readTemperature` y `dht.readHumidify`.

### **Etapa de acondicionamiento de la señal**

Para el acondicionamiento de la señal es necesario el diseño de un circuito en el cual consten un amplificador operacional, LM358N, un diodo, 1N4728 y un potenciómetro de 100k $\Omega$ . Como se puede observar en el siguiente gráfico de la figura 29, la señal analógica entra por el pin 3 del amplificador operacional, en el pin número 2 se conecta el potenciómetro de 100 k $\Omega$  el cual está conectado hacia GND y su pin central se conecta a la salida del amplificador operacional, en el pin número 1 o salida del amplificador operacional se obtiene ya la señal digital la cual proviene de los sensores, en dicho pin se conecta además un diodo el cual está conectado a su vez a GND, para cruzar a la señal por cero o GND. Una vez la señal haya pasado por este proceso y hacen cuenta acondicionada y es posible ingresarla a la ESP32 para su posterior procesamiento.

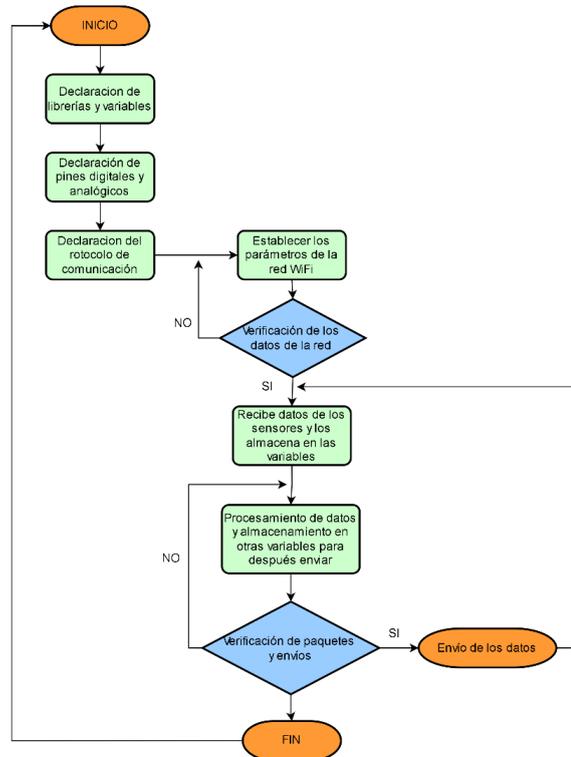


**Figura 29.** Circuito de acondicionamiento de la señal

**Fuente:** El investigador

## Etapa de transmisión de datos

Para la etapa de transmisión de datos de los nodos sensores hacia el nodo LoRa se utiliza la lógica representada en el diagrama de flujo de la figura 30.

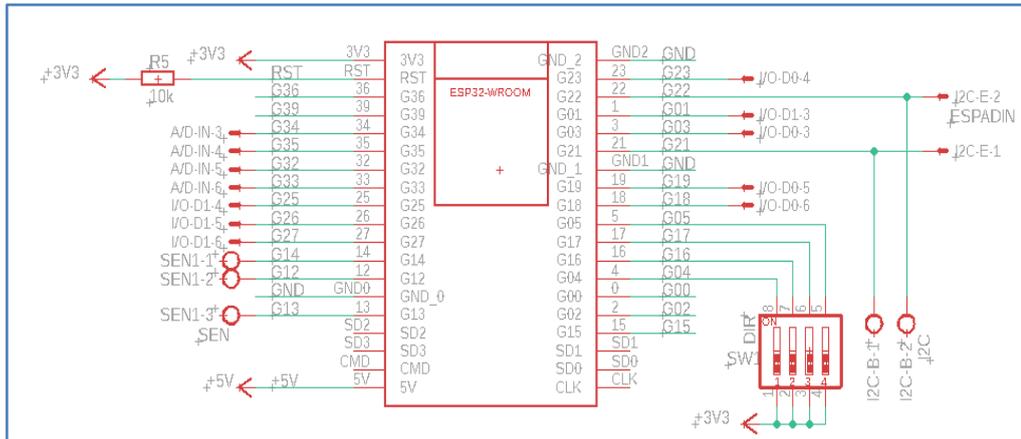


**Figura 30.** Diagrama de flujo del envío de datos

**Fuente:** El investigador

La etapa de transmisión de datos consta de 2 fases. La primera fase se trata de la comunicación entre el módulo ESP32 y el nodo Lora. Para la comunicación entre estos dos dispositivos se utiliza la comunicación conocida como I2C. La comunicación I2C permite el enviar la información hacia el módulo Lora, así como recibir órdenes del módulo Lora hacia los módulos ESP32 para que éstos activen o desactiven a los triacs los cuales están controlando los actuadores.

Se coloca un dip switch de 4 vías para asignar de forma local una dirección en la comunicación I2C, lo cual funciona leyendo las entradas digitales las cuales se asignan a un array de 4 bits el mismo que se convierte en un número binario y de ese número en binario se pasa a decimal que se le asigna a la dirección del I2C. Como se puede observar en el gráfico siguiente el dip switch se conecta a los pines 5, 17, 16, 4, de la ESP32, pines que están configurados como entradas digitales. En la figura 31 se observa el circuito de la ESP32.



**Figura 31.** Conexiones de la ESP32.

**Fuente:** El investigador

La trama de bits que se está utilizando en la comunicación I2C es una trama dinámica y donde se trasmite en formato *string*, el maestro manda una señal de *request* y el esclavo responde con los datos de los sensores con puntero t, h, s con los valores de antes de cada puntero. Y para el control de los triac el maestro envía un mensaje *string* con cabecera A y B con un valor 0/1 para su activación.

En la figura 32 se puede observar el código utilizado en el maestro para lectura de datos, donde *dir\_sect\_i2c* es un *string* con las direcciones de los módulos I2C detectados.

```

for (int cont_1 = 0; cont_1 < n_dir_sect; cont_1++) {
  String datoI2Caux1 = "";
  Wire.requestFrom(dir_sect_i2c[cont_1], uint8_t(18));
  while (Wire.available()) { // Verifica que el puerto serial virtual recibe datos
    char c = Wire.read(); // Serial.println("int -> " + String(int(c)) + " char -> " + c);
    if ((int(c) >= 32) && (int(c) <= 126)) {
      datoI2Caux1 = datoI2Caux1 + c;
    }
  }
  //Serial.println(datoI2Caux1);
  datoI2Caux1 = datoI2Caux1 + char(33 + cont_1);
  byte pos1 = datoI2Caux1.indexOf('t');
  byte pos2 = datoI2Caux1.indexOf('h');
  byte pos3 = datoI2Caux1.indexOf('s');
  tmp_s[cont_1] = (datoI2Caux1.substring(0, pos1)).toFloat();
  hum_s[cont_1] = (datoI2Caux1.substring(pos1 + 1, pos2)).toFloat();
  hums_s[cont_1] = (datoI2Caux1.substring(pos2 + 1, pos3)).toFloat();
  tmp_p = tmp_p + tmp_s[cont_1];
  hum_p = hum_p + hum_s[cont_1];
  hums_p = hums_p + hums_s[cont_1];
  datoI2Creceive = datoI2Creceive + datoI2Caux1;
  datoI2CtoLora = datoI2Creceive;
}
tmp_p = tmp_p / n_dir_sect;
hum_p = hum_p / n_dir_sect;
hums_p = hums_p / n_dir_sect;
oled_line[3] = "Tmp: " + String(int(tmp_p)) + " *C";
oled_line[4] = "H.s: " + String(int(hum_p)) + " %";
oled_line[5] = "H.a: " + String(int(hums_p)) + " %";
}

```

**Figura 32.** Maestro para lectura

**Fuente:** El investigador

### Etapa de transmisión de datos entre los nodos LoRa y el Gateway

Para la transmisión de información desde los nodos Lora, el primero dentro del invernadero y el segundo fuera del invernadero en el sector de la bomba de agua, hasta el Gateway Lora se utiliza el tipo de comunicación SPI. Para llevar a cabo esta comunicación se utiliza la frecuencia 915MHz, que es la permitida para ISM en Ecuador.

Se puede observar en la figura 33 la declaración de variables referentes a la comunicación SPI donde se establece la banda de frecuencia a utilizar, así como varios ajustes de la pantalla del módulo LoRa.

```
#define BAND 915E6
#define OLED_RST -1
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define resistor 10000
#define volt 3.3
```

**Figura 33.** Configuración banda y pantalla Nodos LoRa

**Fuente:** El investigador

Para inicializar el bus SPI se utiliza la función SPI.begin() donde además se está configurando los pines SCK, MISO, MOSI y SS como salida. Además de las funciones que se pueden ver en la figura 34 para que establecer la comunicación LoRa.

```
SPI.begin(SCK, MISO, MOSI, SS);
LoRa.setPins(SS, RST, DIO0);
if (!LoRa.begin(915E6)) {
  while (1);
}
LoRa.setSyncWord(EEPROM.read(2));
LoRa.enableCrc();
LoRa.onReceive(onReceive);
LoRa.onTxDone(onTxDone);
LoRa_rxMode();
```

**Figura 34.** Inicialización comunicación SPI.

**Fuente:** El investigador

Para que la comunicación LoRa sea posible es necesario establecer las funciones para el envío y recepción de datos, así como la función para el envío de mensajes, en la figura 35 se puede apreciar que para la recepción de datos se utiliza la función LaRa\_rxMode(), Para la transmisión de datos se utiliza la función LoRa\_txMode(), y para el envío de mensajes se utiliza la función LoRa\_sendMessage(String messahe).

```

void LoRa_rxMode() {
  LoRa.enableInvertIQ(); // invertir activamente las señales I y Q
  LoRa.receive(); // establecer el modo de recepción
}
void LoRa_txMode() {
  LoRa.idle(); // establece el modo de espera
  LoRa.disableInvertIQ(); // modo normal
}
void LoRa_sendMessage(String message) {
  LoRa_txMode(); // establece el modo tx
  LoRa.beginPacket(); // inicia el paquete
  LoRa.print(message); // añade la carga útil
  LoRa.endPacket(true); // termina el paquete y lo envía
  //Serial.println(message);
}
}

```

**Figura 35.** Configuración para comunicación LoRa

**Fuente:** El investigador

Una vez que la comunicación entre los nodos Lora y el Gateway es la esperada se procede a enviar la información a través del siguiente código. Con el propósito de ordenar mejor el código se utiliza la función switch case la cual permite organizar en bloques de código, de forma que éstos se ejecuten cuando se cumple una cierta condición, con ello se logra visualizar y ordenar mejor el código, tal como se observa en la figura 36.

```

lora_r = " ";
byte q_id, q_st;
switch (fun) {
  case 0: // no hace nada
    break;
  case 1: // envia valor del los sensores al gateway LoRa
    datoLoRasend = (
      String(EEPROM.read(2)) + "gr" +
      String(EEPROM.read(0)) + "ie" +
      String(1) + "f" +
      datoI2CtoLora + //255gr16ie3f16nrge2f
      String(pW).c_str() + "w" +
      String(vr_ph).c_str() + "p" +
      String(pc).c_str() + "c"); // Serial.println(datoLoRasend);
    LoRa_sendMessage(datoLoRasend);
    fun = 0;
    break;
}

```

**Figura 36.** Envío de datos entre nodo LoRa y Gateway

**Fuente:** El investigador

Como se pudo observar en la figura 36 el case 0 no hace nada, el case 1 tiene la función de enviar los valores sensados hacia el gateway Lora, el case 2 de la figura 37, tiene la

función de activar o desactivar los contactores los cuales de la misma manera activan o desactivan los actuadores o electroválvula.

```
case 2: // confirma que se ejecutaron las funciones de control
  q_id = LoRa_receive.substring(0, LoRa_receive.indexOf("q")).toInt();
  q_st = LoRa_receive.substring(LoRa_receive.indexOf("q") + 1, LoRa_receive.length()).toInt();
  if (q_id < 4) {
    if (q_id == 3) {
      contactores_locales_stado[3] = 1;
      digitalWrite(q3, (contactores_locales_stado[3]));
      delay(100);
      contactores_locales_stado[3] = 0;
      digitalWrite(q3, (contactores_locales_stado[3]));
    } else {
      contactores_locales_stado[q_id] = q_st;
    }
  } else {
    contactor_i2c_stado[q_id - 4] = q_st;
  }

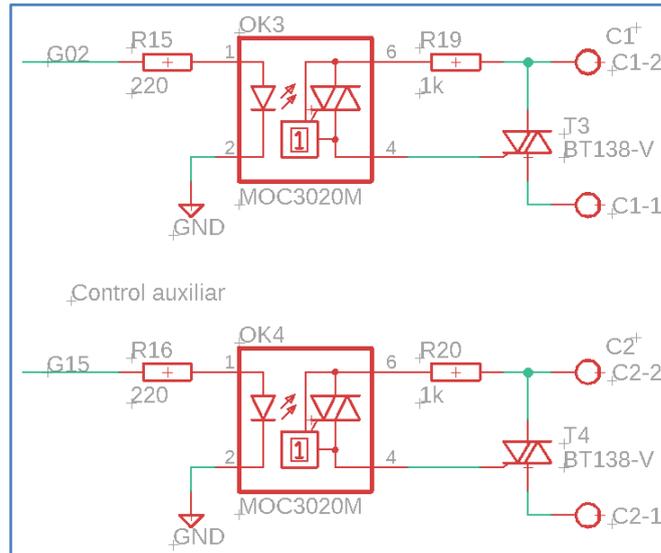
  datoLoRasend = (
    String(EEPROM.read(2)) + "gr" +
    String(EEPROM.read(0)) + "ie" +
    String(4) + "f");
  LoRa_sendMessage(datoLoRasend);
  fun = 0;
  break;
```

**Figura 37.** Control de actuadores

*Fuente: El investigador*

### **Etapa de control de actuadores y electroválvulas**

Para la etapa de encendido o apagado de los actuadores o electroválvulas se diseña un circuito el cual está constituido de un diodo de arseniuro de galio, o más conocido como MOC3020M, el cual se encarga de emitir infrarrojos ópticamente acoplado a un interruptor de 2 lados de silicio en un paquete de inmersión de tipo 6-lead. A la entrada de este modo se conecta a una resistencia de 220 ohmios en el pin número 1 en el pin número dos se conecta directamente GND, a la salida del MOC se conecta una resistencia de 1 k $\Omega$  y un triac BT138, el cual se utiliza para el control y conmutación de corrientes alternas nominales. El triac funciona como un conmutador en corriente alterna. Tras el diseño de este circuito ya es posible activar o desactivar los diferentes actuadores o electroválvulas que funcionan con corriente alterna. La figura 38, muestra el circuito mencionado.



**Figura 38.** Circuito control de actuadores y electroválvulas

**Fuente:** El investigador

Una vez realizado el diseño del circuito para la activación y desactivación de los actuadores o electroválvulas, es necesario un código de programación que permita activar o desactivar los triac, en la figura 39, se muestra el código del maestro para envío de activación de los triac.

```
void control_sectores_i2c() {
  for (int i = 0; i < n_dir_sect; i++) {
    if ((contactor_i2c_stado[i] != aux_c_s[i]) || (contactor_i2c_stado[i + n_dir_sect] != aux_c_s[i + n_dir_sect])) {
      Wire.beginTransmission(dir_sect_i2c[i]);
      Wire.write({
        String(contactor_i2c_stado[i] + "a" +
              String(contactor_i2c_stado[i + n_dir_sect] + "b").c_str());
      });
      Wire.endTransmission();
      aux_c_s[i] = contactor_i2c_stado[i];
      aux_c_s[i + n_dir_sect] = contactor_i2c_stado[i + n_dir_sect];
    }
    delay(50);
  }
}
```

**Figura 39.** Código del maestro para envío de activación de los triac.

**Fuente:** El investigador

La función escritura solo se activa cuando el maestro envía un *request* con la función **Wire.requestFrom()**, la función escritura como su nombre lo dice es la que escribe los datos de la temperatura ambiente, humedad ambiente y la humedad del suelo. La función **leer\_sensores** se encuentra ejecutándose todo el tiempo, esta función permite que se esté leyendo constantemente las variables de temperatura de ambiente, humedad de ambiente y humedad del suelo. Como se puede observar en la figura 40 para leer la

humedad de ambiente y temperatura de ambiente se está utilizando una librería específica para el sensor DHT.

```
void escritura() {
    // int s_hs = analogRead(sensor_hums);
    String x = "";
    x = (String(temp) + "t" + String(hum) + "h" + String(hums) + "s");
    Wire.write(x.c_str());
    // Serial.print("valor envio de datos: ");
    // Serial.println(x);
}

void leer_sensores() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    int s_hs = analogRead(sensor_hums);
    hums = 100 - (s_hs * 100 / 4096);
    // float hs = s_hs;
    if (isnan(h) || isnan(t)) {
        // Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }
    temp = t;
    hum = h;
    // Serial.println(analogRead(sensor_hums));
    // hums = hs;
}
```

*Figura 40. Código de escritura y lectura de los valores sensados.*

*Fuente: El investigador*

Según lo que el maestro envíe, las salidas digitales se activan o desactivan, como se visualiza en la siguiente función de la figura 41.

```
void eventoRecepcion(int event) {
    message = Wire.readString();
    Serial.println("Mensaje RX");
    Serial.println(message);
    q[0] = message.substring(0, message.indexOf('a')).toInt();
    q[1] = message.substring(message.indexOf('a') + 1, message.indexOf('b')).toInt();
    Serial.println(q[0]); Serial.println(q[1]);
}
```

*Figura 41. Código para la activación y desactivación de las salidas digitales.*

*Fuente: El investigador*

### **Monitoreo de la radiación solar**

Como se mencionó en apartados anteriores, el sensor seleccionado para poder medir la radiación solar que está llegando a las plantas es el ML8511, el cual cuenta con una

salida analógica que está relacionada linealmente con la intensidad UV ( $mW/cm^2$ ). Por tanto, la programación para este sensor es sencilla. En la figura 42 se declara una variable flotante *s\_radiación* la cual sea igual a 0.

```
float s_luz = 0,
```

**Figura 42.** Declaración de variable flotante para sensor de radiación.

**Fuente:** El investigador

A continuación, en la figura 43, se lee el sensor el cual envía una señal analógica. este sensor cuando es alimentado por 3.3V llega a un rango máximo de 669, pero se necesita llegar a un rango máximo de 1023, por tanto, el valor medido se multiplica por 1023 y se divide entre 3,3, que corresponde al voltaje de alimentación. Al necesitar los datos en  $W/m^2$ , es necesario multiplicar por 10 al valor final tal como se puede apreciar en la siguiente figura.

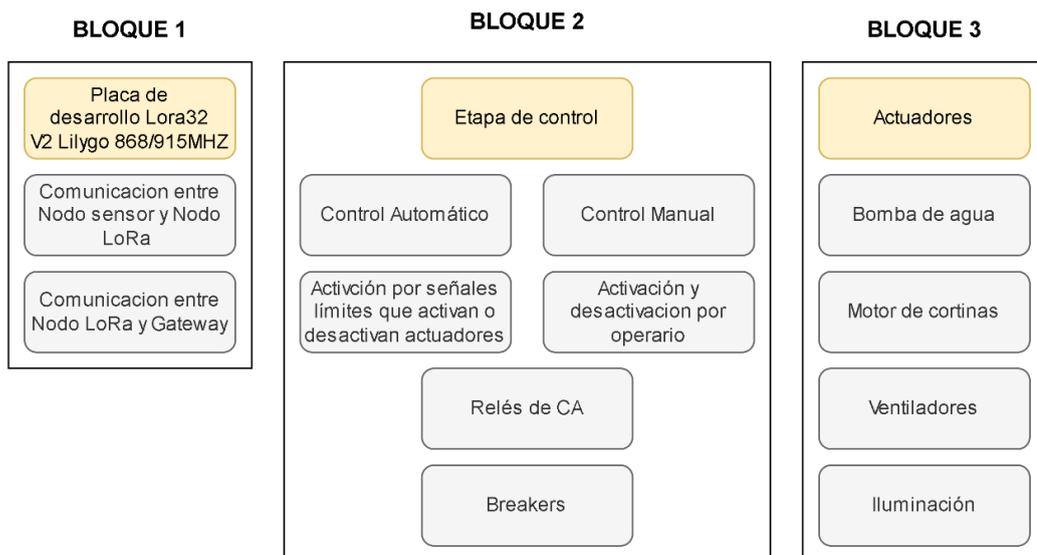
```
s_luz = s_luz + ((analogRead(pin_sensor_luz)));
vr_luz = (((s_luz*1023)/3.3)*10);
```

**Figura 43.** Declaración de variable flotante para sensor de radiación.

**Fuente:** El investigador

### Control del sistema

Dentro de la etapa del control de sistema se puede dividir en 3 bloques, placa de desarrollo, etapa de control y actuadores.



**Figura 44.** Etapas del Control del sistema

**Fuente:** El investigador

En el bloque 1 de la figura 44, se considera la placa de desarrollo que se encuentra en cada nodo Lora programado para controlar a los actuadores y la comunicación entre el nodo sensor y el nodo Lora, y el nodo Lora con el gateway. Como ya se mencionó anteriormente, la comunicación entre los nodos sensores y los nodos Lora se realiza a través de la comunicación I2C, y la comunicación entre los nodos Lora y el Gateway se realiza utilizando la comunicación Lora, la frecuencia utilizada es 915 MHz.

En el segundo bloque 2, se encuentra la etapa de control, el sistema cuenta con dos formas de uso el primero automático y el segundo manual. en el modo automático los autores serán activados y desactivados a través de valores limites superiores e inferiores que se declaran en la programación, por ejemplo, si la temperatura supera los 26°C los ventiladores se encenderán y las cortinas se abrirán, si la temperatura es inferior a 26°C las cortinas se cierran, es decir el operario no debería estar pendiente de la temperatura del invernadero debido a que el sistema tomará decisiones por sí solo. En el modo manual el operario deberá estar continuamente revisando las diferentes variables que mide el sistema para poder tomar decisiones adecuadas para la plantación encendiendo y apagando los actuadores de forma manual. Para encender y apagar los actuadores se utiliza el circuito electrónico anteriormente descrito con el MOC3020M y a su vez con relés de corriente alterna. Además, se utilizan breakers para prevenir que haya algún daño en los elementos de electrónicos y eléctricos en el caso de que hubiera una subida o bajada de tensión brusca.

El bloque 3, está constituido por los actuadores, actuadores como bomba de agua, motor de cortinas, ventiladores, iluminación. Dichos actuadores se encargarán de dar unas mejores condiciones climáticas en el interior del invernadero para que la plantación incremente su producción.

#### **3.1.2.3.2 Comunicación inalámbrica entre Nodos LoRa y Gateway**

Para realizar la comunicación inalámbrica el primer paso es establecer la ubicación exacta de los nodos Lora y el Gateway. para ello se utiliza un software denominado Google Earth, el cual permitirá observar la ubicación exacta, así como la distancia que existe entre dichos nodos, A continuación, se muestra tabla 23, con las coordenadas geográficas, así como una imagen satelital de los puntos a tomar en cuenta.

**Tabla 23. Coordenadas geográficas de los nodos.**

Nodo	Coordenadas		Imagen Satelital
<b>Nodo LoRa Invernadero</b>	Latitud	1°19'2.73"S	
	Longitud	78°38'3.84"O	
<b>Nodo LoRa Bomba</b>	Latitud	1°19'2.81"S	
	Longitud	78°38'2.56"O	
<b>Gateway</b>	Latitud	1°19'2.78"S	
	Longitud	78°38'4.79"O	

*Elaborado por: El investigador*

Para realizar los diferentes cálculos también es necesario conocer la distancia que existen entre los diferentes nodos, a continuación, se detallan en la tabla 24.

**Tabla 24. Distancia Nodos – Gateway**

Enlace	Distancia
<b>Nodo LoRa Invernadero - Gateway</b>	132,95 m
<b>Nodo LoRa Bomba - Gateway</b>	114,26 m

*Elaborado por: El investigador*

Para el diseño de la red inalámbrica es necesario calcular diferentes parámetros tales como:

- Frecuencia
- Zona de Fresnel
- Pérdidas del espacio libre

- Margen de desvanecimiento
- Potencia de recepción disponible
- Margen de umbral

A continuación, se detallan los cálculos para que la comunicación LoRa sea posible.

### **Enlace: Nodo LoRa Invernadero - Gateway**

#### **Frecuencia**

$$f = \frac{f_{\min} + f_{\max}}{2}$$

Donde:

Fmin= Frecuencia mínima (MHz)

Fmax= Frecuencia máxima (MHz)

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

$$f = 915MHz$$

$$f = 0.915GHz$$

#### **Zona de Fresnel**

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

Donde:

d= distancia (km)

f= frecuencia (GHz)

$$r = 17.32 \sqrt{\frac{0.13295 Km}{4 \cdot 0.915GHz}}$$

$$r = 9.301m$$

$$r_1 = 0.6 \cdot r$$

$$r_1 = 0.6 \cdot 9.301m$$

$$r_1 = 1.981m$$

#### **Pérdidas del espacio libre**

$$F_{SL}(dB) = 20\log(d) + 20\log(f) + 32.4$$

Donde:

d= distancia (km)

f= frecuencia (MHz)

$$F_{SL}(dB) = 20\log(0.13295Km) + 20\log(915MHz) + 32.4$$

$$F_{SL}(dB) = 74.102 dB$$

### Margen de desvanecimiento

$$FM = 30\log(d) + 10\log(6 \cdot a \cdot b \cdot f) - 10\log(1 - R) - 70$$

Donde:

d: distancia (km)

f: frecuencia (GHz)

R: confiabilidad del sistema la cual normalmente es del 99.99%, por tanto, el valor de R es 0.9999.

a: factor de rugosidad de la trayectoria en este caso vale 1.

b: factor para convertir la probabilidad del peor de los meses en probabilidad, en este caso vale 0.25.

$$FM = [30\log(0.13295Km) + 10\log(6 \cdot 1 \cdot 0.25 \cdot 0.915GHz) - 10\log(1 - 0.9999) - 70]$$

$$FM = -54.914 dB$$

### Potencia de recepción disponible

$$P_{R_x} = P_{T_x} - AL_{T_x} + G_{T_x} - F_{SL} + G_{R_x} - AL_{R_x}$$

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.

$$P_{R_x} = 20dB - 0.5dB + 2dB - 74.102 dB + 2dB - 0.5dB$$

$$P_{R_x} = -51.102dB$$

### Margen de Umbral

$$MU = P_{R_x} - S_r$$

Donde:

$S_r$ : sensibilidad del receptor

$$MU = -51.102dB - (-138dB)$$

$$MU = 86.898dB$$

Nota: Para que la señal llegue del transmisor al receptor se debe cumplir que el margen de umbral debe ser mayor o igual al margen de desvanecimiento.

$$MU \geq FM$$

Para este caso si se cumple puesto que:

$$86.898dB \geq -54.914 dB$$

Por tanto, la transmisión de señal se realizará correctamente.

### Enlace: Nodo LoRa Bomba - Gateway

#### Frecuencia

$$f = \frac{f_{\min} + f_{\max}}{2}$$

Donde:

$f_{\min}$ = Frecuencia mínima (MHz)

$f_{\max}$ = Frecuencia máxima (MHz)

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

$$f = 915MHz$$

$$f = 0.915GHz$$

#### Zona de Fresnel

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

Donde:

d= distancia (km)

f= frecuencia (GHz)

$$r = 17.32 \sqrt{\frac{0.11426 \text{ Km}}{4 \cdot 0.915 \text{ GHz}}}$$

$$\mathbf{r = 3.06m}$$

$$\mathbf{r_1 = 0.6 \cdot r}$$

$$r_1 = 0.6 \cdot 3.06m$$

$$\mathbf{r_1 = 1.836m}$$

### **Pérdidas del espacio libre**

$$\mathbf{F_{SL}(dB) = 20\log(d) + 20\log(f) + 32.4}$$

Donde:

d= distancia (km)

f= frecuencia (MHz)

$$F_{SL}(dB) = 20\log(0.11426\text{Km}) + 20\log(915\text{MHz}) + 32.4$$

$$\mathbf{F_{SL}(dB) = 72.786 dB}$$

### **Margen de desvanecimiento**

$$\mathbf{FM = 30\log(d) + 10\log(6 \cdot a \cdot b \cdot f) - 10\log(1 - R) - 70}$$

Donde:

d: distancia (km)

f: frecuencia (GHz)

R: confiabilidad del sistema la cual normalmente es del 99.99%, por tanto, el valor de R es 0.9999.

a: factor de rugosidad de la trayectoria en este caso vale 1.

b: factor para convertir la probabilidad del peor de los meses en probabilidad, en este caso vale 0.25.

$$FM = [30\log(0.11426\text{Km}) + 10\log(6 \cdot 1 \cdot 0.25 \cdot 0.915\text{GHz}) - 10\log(1 - 0.9999) - 70]$$

$$\mathbf{FM = -56.888 dB}$$

### Potencia de recepción disponible

$$P_{R_x} = P_{T_x} - AL_{T_x} + G_{T_x} - F_{SL} + G_{R_x} - AL_{R_x}$$

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.

$$P_{R_x} = 20dB - 0.5dB + 2dB - 72.786 dB + 2dB - 0.5dB$$

$$P_{R_x} = -49.786dB$$

### Margen de Umbral

$$MU = P_{R_x} - S_r$$

Donde:

$S_r$ : sensibilidad del receptor

$$MU = -49.786dB - (-138dB)$$

$$MU = 88.214dB$$

Nota: Para que la señal llegue del transmisor al receptor se debe cumplir que el margen de umbral debe ser mayor o igual al margen de desvanecimiento.

$$MU \geq FM$$

Para este caso si se cumple puesto que:

$$88.214dB \geq -56.888 dB$$

Por tanta la transmisión de señal se realizará correctamente.

### Simulación en software

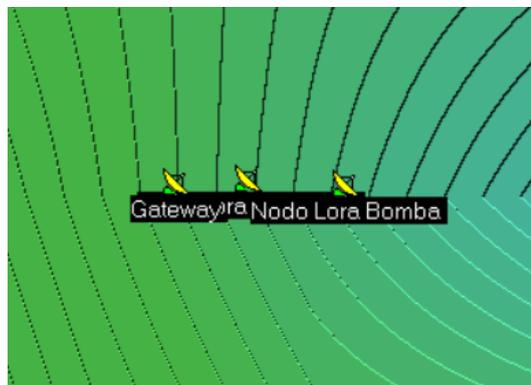
Una vez concluido los cálculos matemáticos es necesario verificar la comunicación a través de un software para poder verificar que la comunicación se está dando entre los

diferentes nodos y quién los cálculos matemáticos sean correctos. para la simulación se utiliza el software.

En el software mencionado se ingresan las ubicaciones de cada uno de los nodos, así como las características de cada uno de los módulos Lora que se utilizan, los cuales están haciendo la función de antena.

### 3.1.2.3.3 Configuración de los módulos LoRa

Para la configuración de los módulos Lora es necesario tener presente el plan de frecuencias de la ICM, para Ecuador que se encuentra en la región dos las frecuencias que le corresponde están entre 902MHz y 928MHz, que son frecuencias asignadas para el uso industrial, para el presente proyecto se utiliza una frecuencia de 915MHz. Debido a que en el presente proyecto se va a realizar una red LoRaWAN es necesario la utilización de 2 módulos Lora emisores y receptores y un módulo Lora que hará la función de Gateway, como se observa en la figura 45, con ello se está formando la red LoRaWAN. la configuración de cada 1 de los módulos Lora es diferente ya que cumplen funciones distintas, a continuación, se describe cómo trabaja cada una de ellas.



**Figura 45.** Ubicación de los nodos

**Fuente:** El investigador

Es necesario aclarar que el nodo de la bomba y el nodo del invernadero se encuentran más cerca de lo que se puede apreciar en la figura anterior, sin embargo, al haber poca distancia entre ellos el software lo mueve hacia un lado para que se pueda visualizar correctamente los tres nodos. La posición real de los nodos se observa en la figura 46.



**Figura 46.** Ubicación de nodos vista satelital

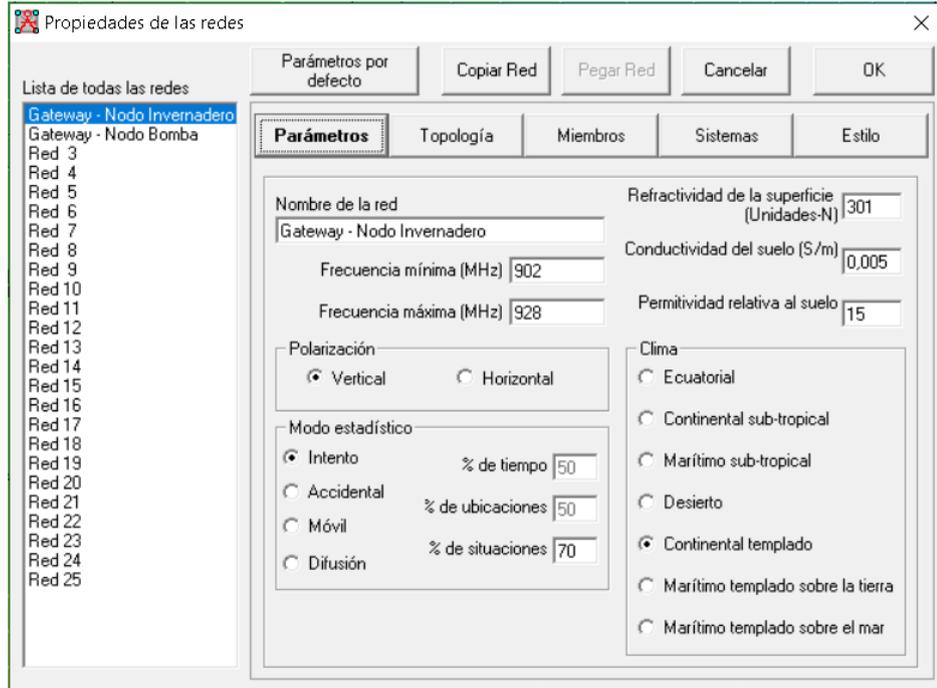
*Fuente: El investigador*

### **Configuración de los nodos en el software Radio Mobile.**

Para poder configurar los nodos es necesario tener en cuenta las características técnicas de cada uno de para poder ingresar en el software. Sus características son:

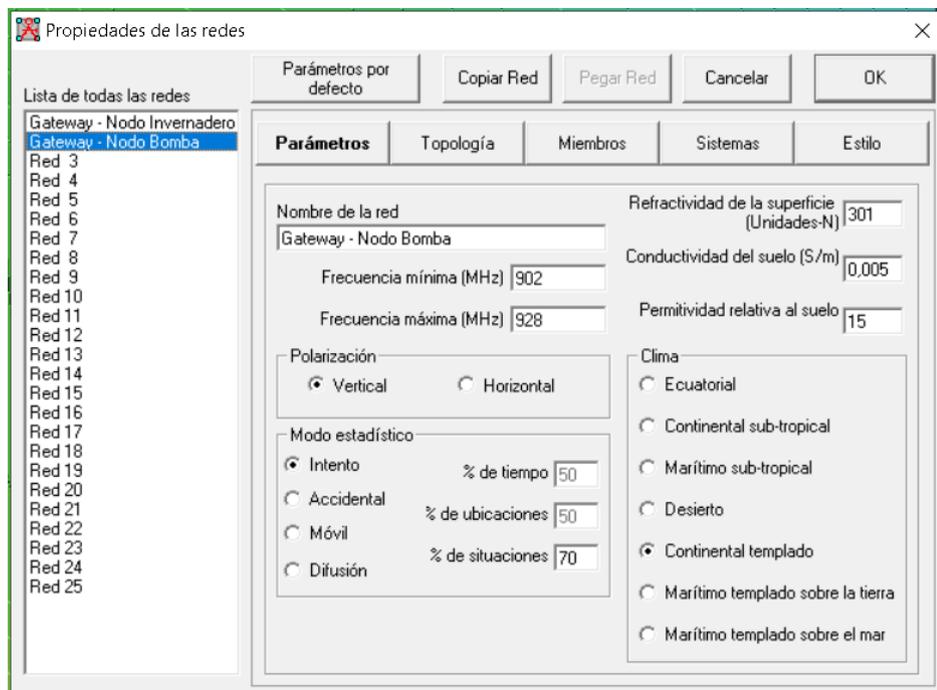
- Perdidas de la línea = 0.5dB
- Ganancia de la antena = 2dB
- Frecuencia máxima= 928MHz
- Frecuencia mínima= 902MHz
- Potencia del transmisor = 20dBm
- Umbral del receptor = -138dBm
- Altura de la antena = 12m
- Tipo de antena = Omnidireccional

El siguiente paso es configurar la red en el software, lo cual se realiza en la ventana de “Propiedades de las Redes”. En dicha ventana se ingresa las características técnicas de las antenas, así como se menciona entre que antenas debe existir la comunicación. En las figuras 47 y 48 se puede observar la configuración.



**Figura 47.** Red Gateway – Nodo Invernadero

*Fuente: El investigador*

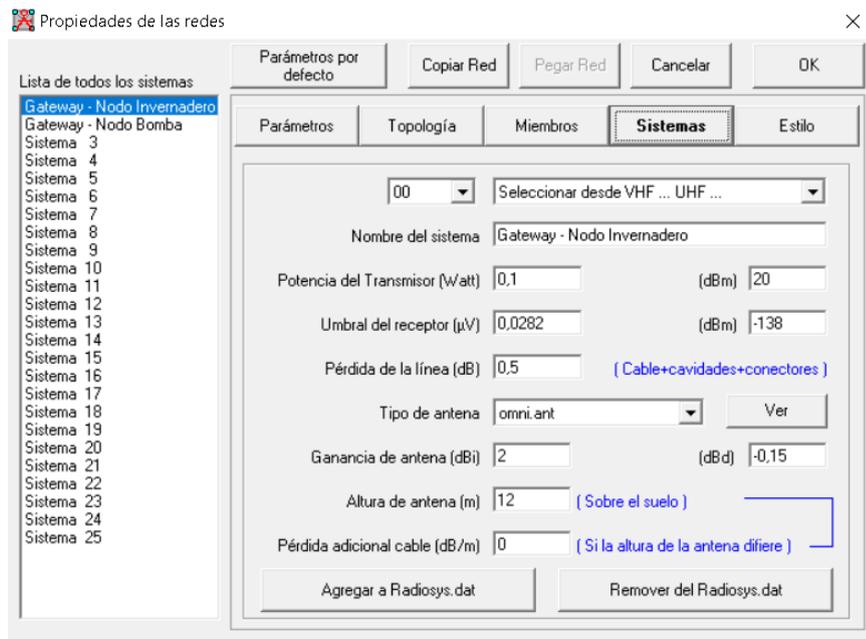


**Figura 48.** Red Gateway – Nodo Bomba

*Fuente: El investigador*

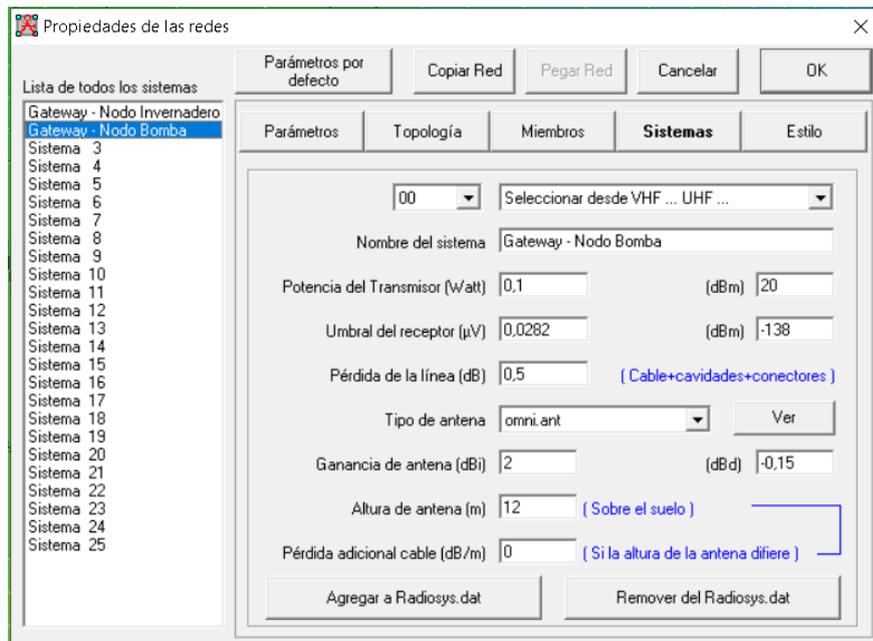
La frecuencia máxima y mínima que se utiliza para simular el enlace es de 928 MHz y 902 MHz respectivamente como se observa en las gráficas anteriores.

El siguiente paso es introducir las características técnicas de cada una de las antenas y proceder a simular, la altura que se utiliza para la antena del Gateway es de 12 m. La configuración se observa en las figuras 49 y 50.



**Figura 49.** Propiedades de las antenas Gateway – Nodo Invernadero

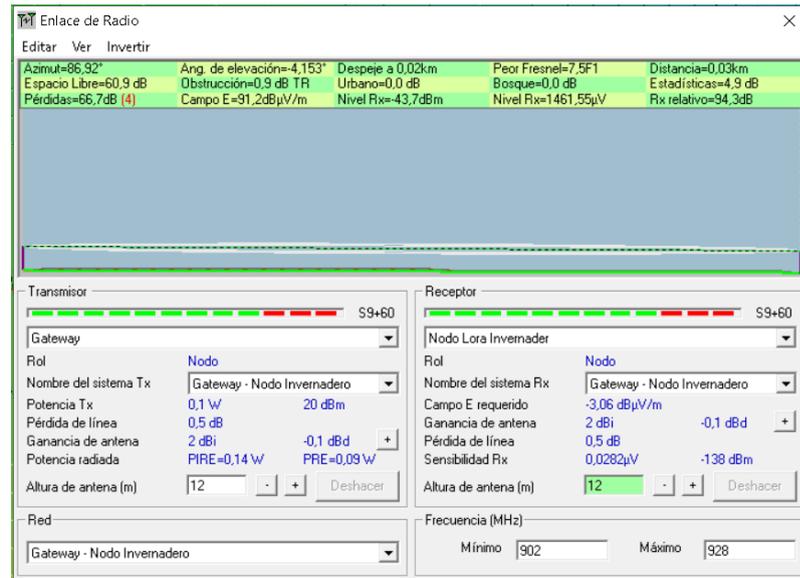
**Fuente:** El investigador



**Figura 50.** Propiedades de las antenas Gateway – Nodo Bomba

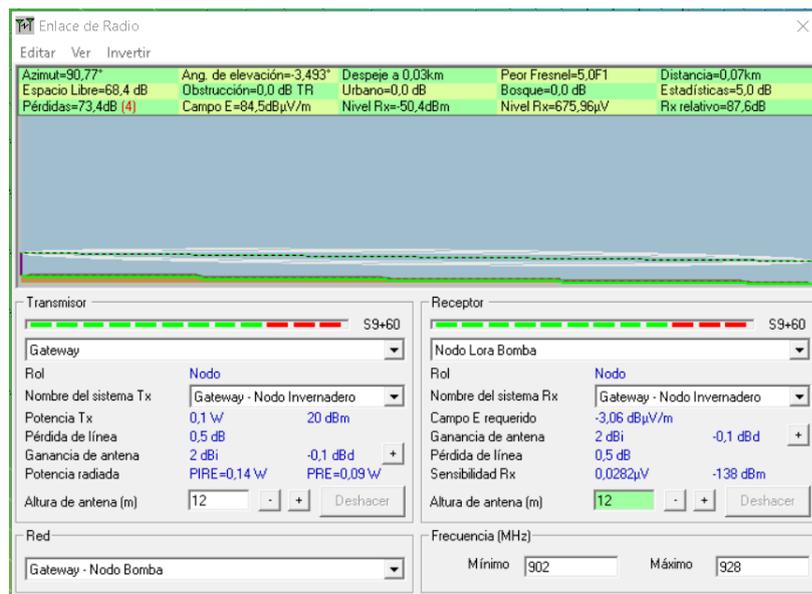
**Fuente:** El investigador

En las figuras 51 y 52 se puede observar los detalles de la comunicación entre los diferentes nodos. la comunicación es adecuada puesto que entre los nodos hay línea de vista directa y la distancia no es excesiva entre los nodos, por tanto, se puede observar que las condiciones se cumplen para que la comunicación sea posible, los valores obtenidos están dentro del rango adecuado en la comunicación entre los tres nodos. en las figuras en la parte superior de color verde se puede observar diferentes datos como la peor zona de Fresnel, pérdidas, espacio libre, ángulo de elevación, entre otros.



**Figura 51.** Detalles de la comunicación Gateway – Nodo Invernadero

*Fuente: El investigador*

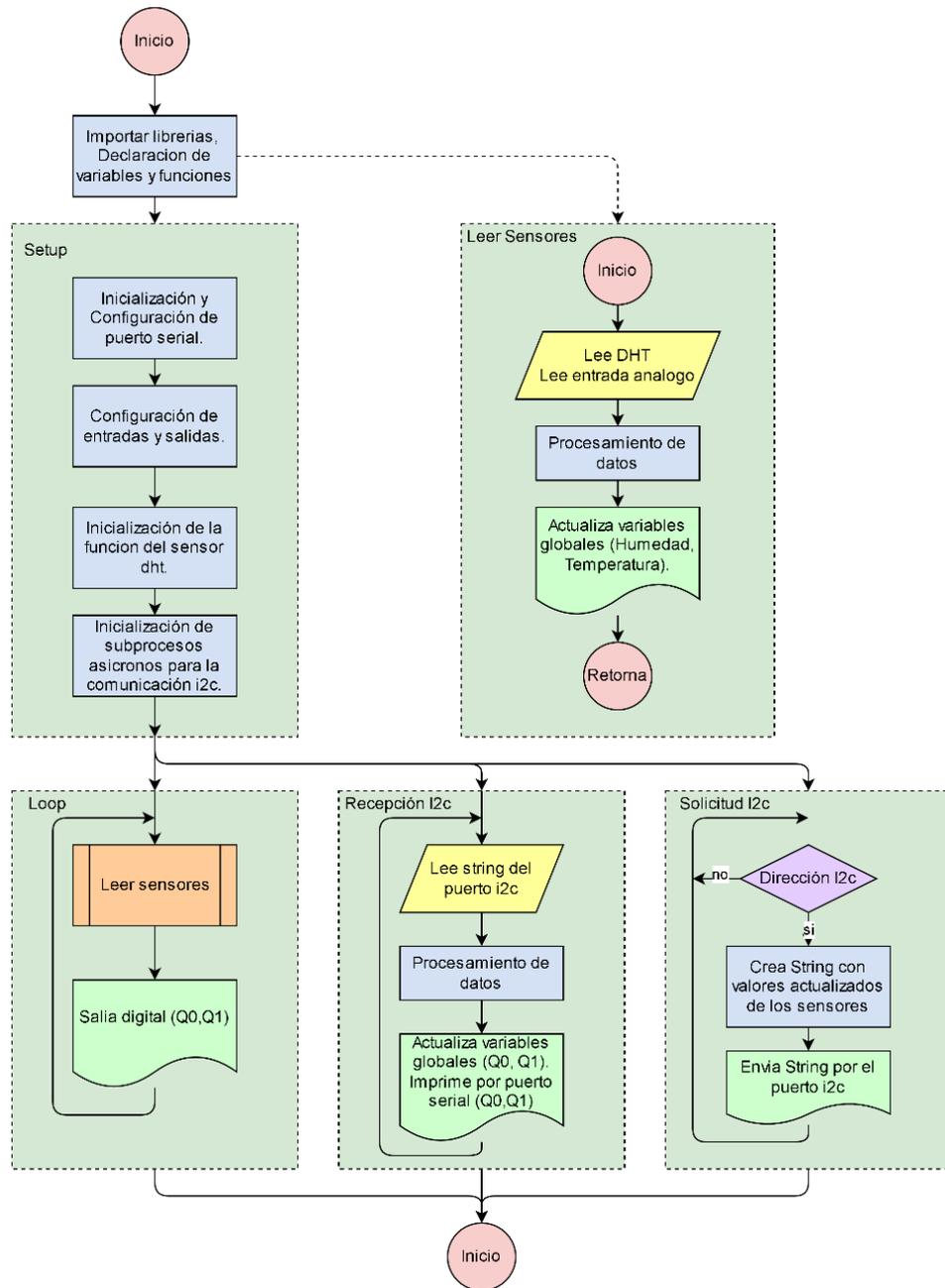


**Figura 52.** Detalles de la comunicación Gateway – Nodo Bomba

*Fuente: El investigador*

## Programación de los nodos sensores

Los nodos sensores son los encargados de adquirir los datos de cada uno de los sensores que se encuentran monitoreando el invernadero, los sensores están continuamente sensando y enviando los datos hacia los nodos sensores (ESP32), para que los datos sean procesados y se puedan enviar hacia el nodo Lora invernadero, la lógica utilizada para la programación de los nodos sensores es la que se puede ver en la figura 53.

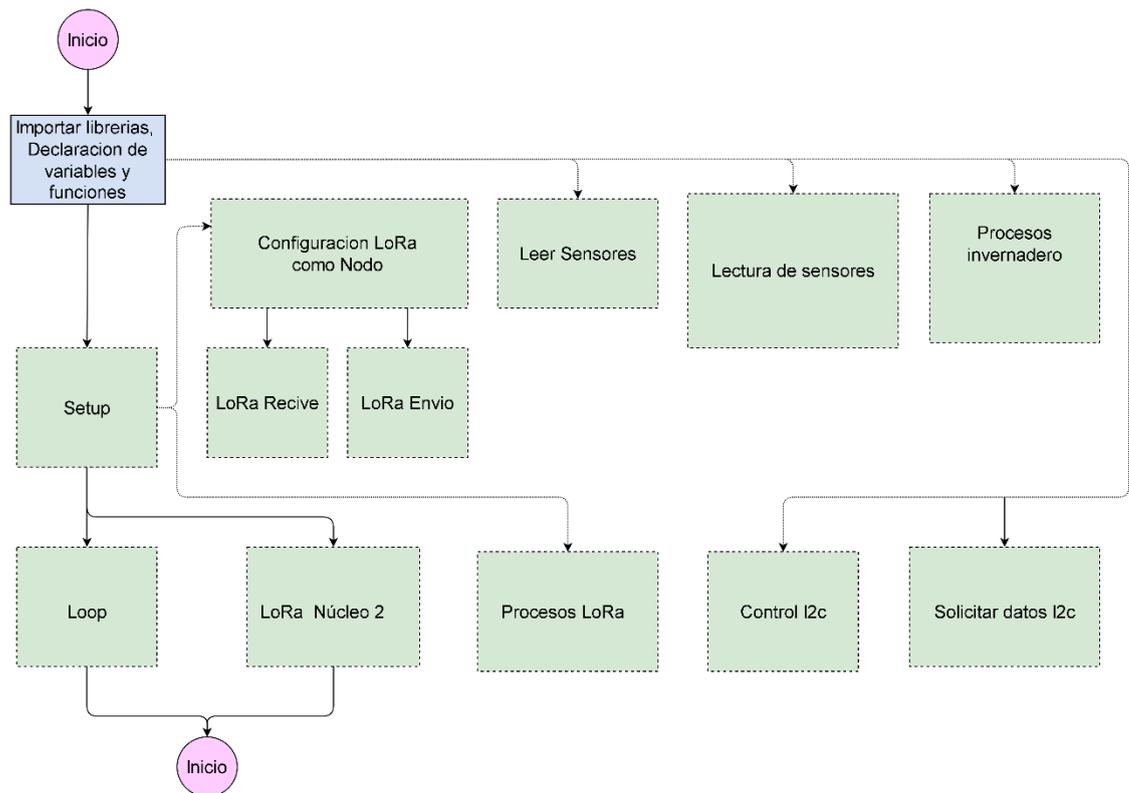


**Figura 53.** Diagrama de flujo de los nodos sensores

**Fuente:** El investigador

## Programación de los Nodos LoRa Invernadero

Para la configuración del nodo que se encuentra dentro del invernadero se utiliza la lógica de programación que se puede apreciar en la figura 54 (la lógica de programación de cada bloque se puede observar en el Anexo 17), el nodo del invernadero se encarga de procesar los datos que recibe de los nodos sensores y enviarlos hacia el Gateway, además debe recibir órdenes por parte del operario a través del Gateway lo que implica que debe tener una comunicación half dúplex. La programación se detallará en el Anexo 18.

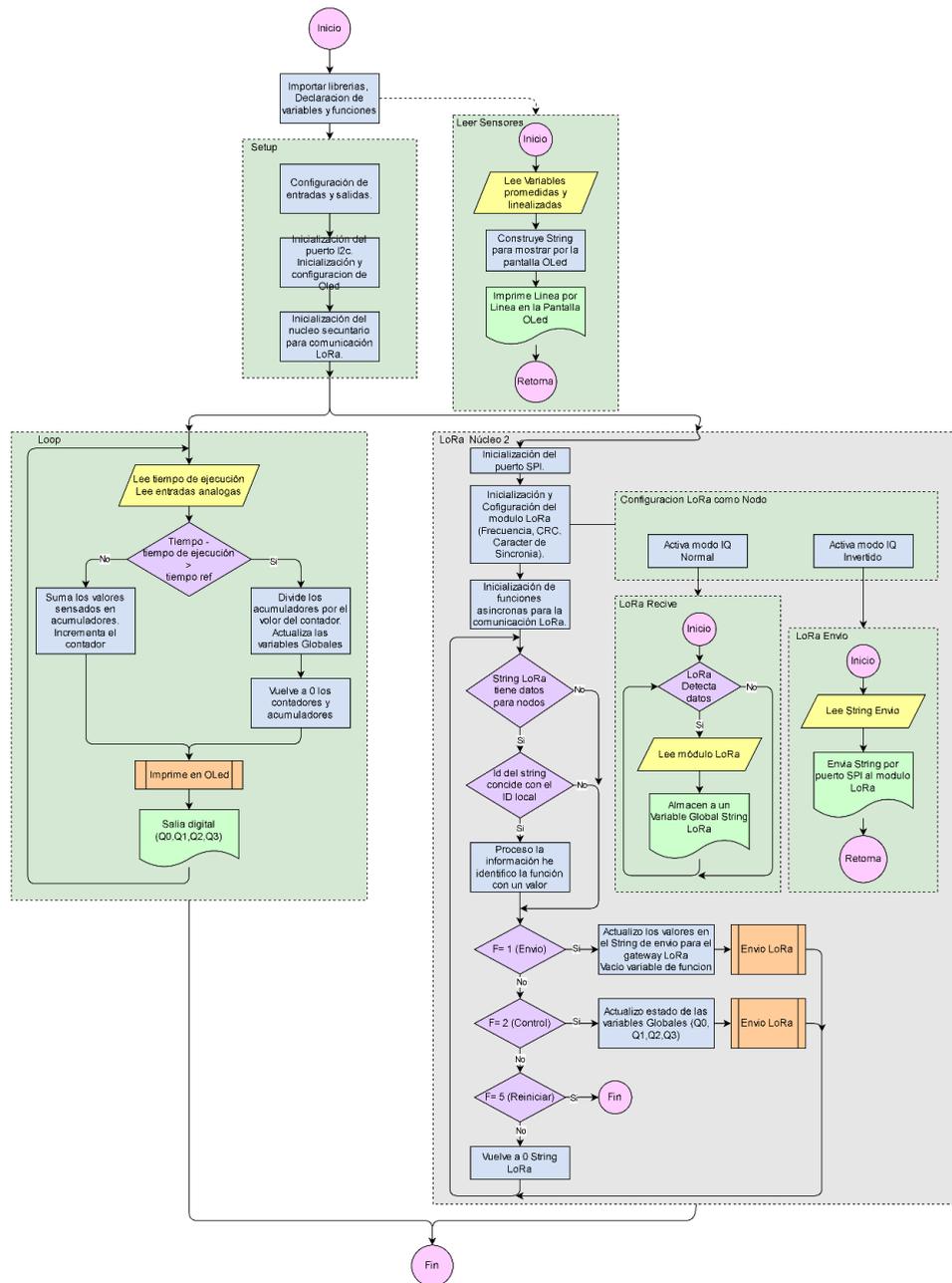


**Figura 54.** Diagrama de flujo nodo LoRa Invernadero

**Fuente:** El investigador

## Programación de los Nodos LoRa Bomba

Para la configuración del nodo que se encuentra en la ubicación de la bomba de agua se utiliza la lógica de programación que se puede apreciar en la figura 55, el nodo de la bomba se encarga de procesar los datos y enviarlos hacia el Gateway, además debe recibir órdenes por parte del operario a través del Gateway lo que implica que debe tener una comunicación half dúplex. La programación se detallará en el Anexo 19.

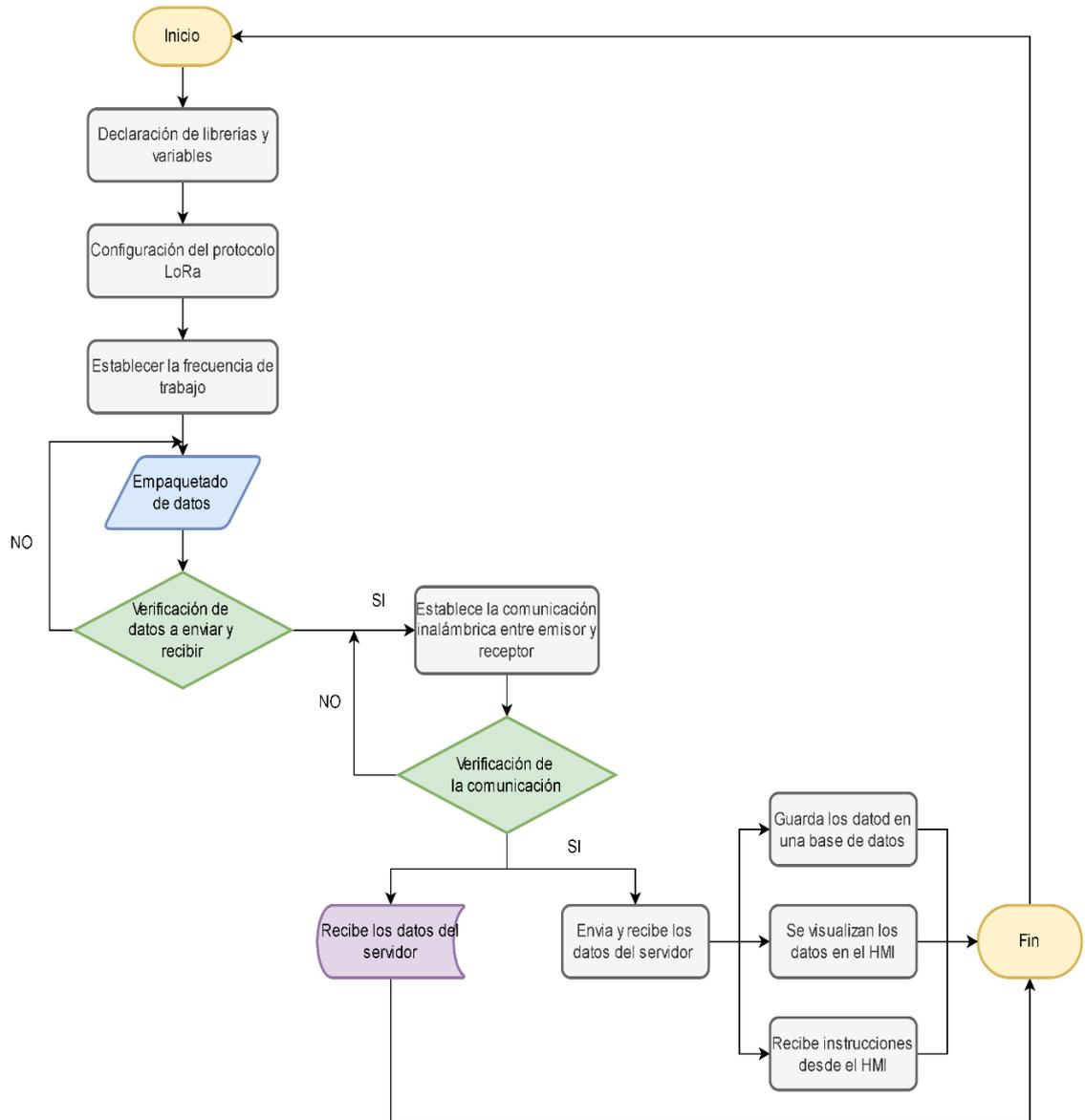


**Figura 55. Diagrama de flujo nodo LoRa Bomba**

**Fuente:** El investigador

### Programación del nodo Gateway

La función que desarrollará en nodo Gateway es comunicar al servidor con los dos nodos Lora que se encuentran dentro de la red, por tanto, el gateway recibe la información de los dos nodos, envía la información hacia el servidor, y envía las órdenes hacia los nodos Lora para realizar el control del sistema. Para poder comprender de mejor manera la programación se realiza el diagrama de flujo de la figura 56. La programación se detallará en el Anexo 20.



**Figura 56.** Diagrama de flujo del nodo Gateway

**Fuente:** El investigador

### Configuración del servidor

Como ya se había mencionado anteriormente para el servidor se utiliza una raspberry Pi 4b de 4GB de memoria RAM, ya que cuenta con las especificaciones y características necesarias para que el sistema operativo trabaje de manera fluida y rápido.

En la raspberry se instala el sistema operativo raspbian, mosquito, Node-RED, LAMP, MySQL, phpMyAdmin, y Arduino IDE. A continuación, se detalla el proceso de instalación.

#### 3.1.2.3.4 Instalación de software utilizados

La instalación de los diferentes softwares utilizados para el sistema de control y monitoreo se detalla en la sección de anexos. La instalación del protocolo MQTT se encuentra en el anexo 10, la instalación del servidor LAMP en el anexo 11, la instalación de Node-RED en el anexo 12, la instalación de Arduino IDE en el anexo 13 y la instalación de Ngrok en el anexo 14.

#### 3.1.2.3.5 Configuración del Gateway

Para poder realizar las configuraciones del Gateway se crea una página web para poder realizar cualquier variación de forma sencilla y rápida para ello se utiliza un servidor web asíncrono utilizando las bibliotecas de ESPAsyncWebServer, debido a que es fácil manejar más de una conexión al mismo tiempo, la respuesta es inmediata al enviar una respuesta ya que el servidor se encarga de enviar la respuesta en segundo plano, en la figura 57 se observa la utilización del servidor web.

```
AsyncWebServer server(80);  
AsyncWebsocket ws("/ws");  
WiFiClient ESP_LoRa_Gateway;  
PubSubClient client(ESP_LoRa_Gateway);
```

*Figura 57. AsyncWebServer*

*Fuente: El investigador*

Primero se utiliza una variable string para asignarle una ip al mqtt, como se observa en la figura 58, en este caso es:

```
String mqtt_ip = "192.168.1.8";
```

*Figura 58. Ip del servidor mqtt*

*Fuente: El investigador*

Esta variable string después se reemplazará por los datos que estarán guardados en la memoria EEPROM. Como se puede observar en la figura 59, primero se inicia la EEPROM para posteriormente grabar los datos en ella.

```
void EEPROM_init() {  
  EEPROM.begin(EEPROM_SIZE);  
  hostname = "GWY LoRa " + String(EEPROM.read(0));  
}
```

*Figura 59. EEPROM*

*Fuente: El investigador*

En el caso que el usuario pierda la conexión a internet de la red general de su vivienda o la red que esté utilizando para manejar el sistema, se podrían presentar cambios en las direcciones IP del gateway o del servidor mqtt, por lo que es necesario diseñar una forma fácil y rápida de corregir dichos cambios y que el sistema vuelva a estar en funcionamiento de manera correcta, por ende se crea el servidor web que se puede observar en la figura 60, donde es posible modificar el id del gateway, la IP del servidor mqtt y el puerto que se está utilizando.



**Figura 60.** Servidor web

**Fuente:** El investigador

Para la programación y diseño de la página web se utiliza la Visual Studio debido a que es un IDE completo que permite entre otras cosas desarrollar páginas web.

Para programar la comunicación se utiliza WebSocket, que consiste en un protocolo basado en TCP, el cual permite la comunicación entre dos puntos finales de manera fiable y eficiente, de manera que el intercambio de datos pueda ser en ambas direcciones, en este caso entre los nodos LoRa y el servidor. Como se visualiza en la figura 61.

```
function initWebSocket() {  
    console.log("Trying to open a WebSocket connection...");  
    websocket = new WebSocket(gateway);  
    websocket.onopen = onOpen;  
    websocket.onclose = onClose;  
    websocket.onmessage = onMessage;  
}
```

**Figura 61.** WebSocket

**Fuente:** El investigador

La función `submit_guardar1()` permite al usuario modificar y guardar el id del nodo en el que se encuentra, así como una cabecera para almacenar los datos según el id, como se observa en la figura 62 y 63.

```
function submit_guardar1() {
  var nodo_id = parseInt(document.getElementById("n_ga").value) - 1;
  console.log(nodo_id);
  websocket.send("id" + nodo_id + "&");
}
```

**Figura 62.** Función `submit_guardar1()`

**Fuente:** El investigador

```
if (message1.indexOf("id") >= 0) {
  EEPROM.write(0, message1.substring(2, message1.indexOf("&")).toInt());
  EEPROM.commit();
  notifyClients("id" + String(EEPROM.read(0)) + "#");
}
```

**Figura 63.** Modificación de id en el gateway

**Fuente:** El investigador

La función `submit_guardar2()` de la figura 64 permite al usuario modificar y guardar la IP del servidor mqtt y el puerto, cabe mencionar que la IP debe ser una ipv4.

```
function submit_guardar2() {
  const ip = document.getElementById("ipv4").value;
  console.log(ip);
  const port = document.getElementById("port").value;
  console.log(port);
  websocket.send("ipv4" + ip + "port" + port);
}
```

**Figura 64.** Función `submit_guardar2()`

**Fuente:** El investigador

```
if (message1.indexOf("ipv4") >= 0) { //
  EEPROM.writeString(2, message1.substring(4, message1.indexOf("port")));
  // Serial.println(message1.substring(message1.indexOf("port") + 4, message1.length()));
  EEPROM.writeInt(18, message1.substring(message1.indexOf("port") + 4, message1.length()).toInt());
  EEPROM.commit();
  notifyClients("ip" + EEPROM.readString(2) + "port" + String(EEPROM.readInt(18)) + "=");
}
```

**Figura 65.** Modificación de la ip y del puerto en el gateway

**Fuente:** El investigador

En la figura 66, se observa el puerto que por default se está utilizando es el puerto 1883, el cual se encuentra declarado en la carpeta con la dirección: `nano /etc/mosquitto/mosquitto.conf`.

```
invernadero@invernadero: ~
File Edit Tabs Help
GNU nano 5.4 /etc/mosquitto/mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
allow_anonymous true

[ File '/etc/mosquitto/mosquitto.conf' is unwritable ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line
```

**Figura 66.** Declaración del puerto en la configuración de mosquitto  
*Fuente: El investigador*

### 3.1.2.3.6 Diseño y configuración del HMI

Una vez instalado el servicio de Node-RED y en Ngrok se procede al diseño y configuración del HMI para poder visualizar y enviar ordenes al sistema de una forma remota y fácil, para ello se procede a programar en Node-RED, para poder acceder a Node-RED se copia el URL que se puede apreciar en la figura 67, primero se inserta el comando “ngrok http 1880” acceder a la siguiente ventana, ahí se podrá copiar la URL de acceso.

```
invernadero@invernadero: ~
File Edit Tabs Help
ngrok (Ctrl+C to quit)
Add Single Sign-On to your ngrok dashboard via your Identity Provider: https://n

Session Status online
Account Freddy Perez (Plan: Free)
Version 3.1.0
Region South America (sa)
Latency 140ms
Web Interface http://127.0.0.1:4040
Forwarding https://...-78-89.sa.ngrok.io -> http://lo...

Connections      ttl   opn   rt1   rt5   p50   p90
                  152   1     0.01  0.17  0.47  1.28

HTTP Requests
-----
GET /red/tours/welcome.js                200 OK
GET /red/images/subflow_tab.svg          200 OK
GET /icons/node-red-contrib-ui-actions/button.png 200 OK
GET /ui_base/gs/gridstack.min.js        304 Not Modified
GET /icons/node-red/rpi.svg             200 OK
GET /flows                               200 OK
```

**Figura 67.** URL para acceder a Node-RED  
*Fuente: El investigador*

### 3.1.2.3.7 Configuración de la seguridad del sistema

Con el fin de que el sistema cuente con un método de seguridad fiable para que el sistema no presente vulnerabilidades se hace uso del parámetro SyncWord, que consiste en un byte definido por los dos últimos Chirps ascendentes del preámbulo, es utilizado para diferenciar redes LoRa que utilicen las mismas bandas de frecuencia. Por tanto, se descartará cualquier transmisión de cualquier dispositivo aun trabajando en la misma banda de frecuencia si la palabra de sincronización es diferente. Para este caso se utiliza el carácter de sincronización 255. En la siguiente figura se puede observar el parámetro de sincronización para el carácter de sincronización.

Para recuperar datos que se pueden perder se hace uso del parámetro CRC, que permite corregir errores. En la figura 68 se puede observar cómo se habilita el parámetro mencionado.

```
108   LoRa.setPins(SS, RST, DI00);
109   if (!LoRa.begin(915E6)) {
110     while (1);
111   }
112   LoRa.setSyncWord(sync_byte);
113   LoRa.enableCrc();
114   LoRa.onReceive(onReceive);
115   LoRa.onTxDone(onTxDone);
116   LoRa_rxMode();
117   for (long m1 = 0; m1 < 600; m1++) {
```

**Figura 68.** SyncWord y CRC

**Fuente:** El investigador

### IQ enable IQ disable

El tráfico de datos es considerable en el presente sistema de control y monitoreo, debido a eso se utiliza la función Invert IQ Signals, la cual permite invertir las señales, logrando diferenciar el mensaje del emisor y del receptor. La intención es no saturar el tráfico de datos por ende una de las opciones es utilizar la función mencionada como se observa en la figura 69 y 70, con ello se logra que los dos nodos no se lean entre sí, sino que directamente se comuniquen al gateway y del mismo modo el que gateway no lea a otros gateways, si es que se encontrarán cerca, sino que lea solamente a los nodos, esto pasa por qué los gateway leen mensajes con InverterIQ desactivado y envían mensajes con InverterIQ activado, y por otro lado los nodos leen mensajes con InvertIQ activado y envían mensajes con InvertIQ desactivado.

```

68 void LoRa_rxMode() {
69     LoRa.enableInvertIQ();           // invertir activamente las señales I y Q
70     LoRa.receive();                 // establecer el modo de recepción
71 }
72 void LoRa_txMode() {
73     LoRa.idle();                     // establece el modo de espera
74     LoRa.disableInvertIQ();         // modo normal
75 }

```

**Figura 69.** *Invert IQ Signals en los Nodos LoRa*

**Fuente:** *El investigador*

```

void LoRa_rxMode() {
    LoRa.disableInvertIQ();
    LoRa.receive();
}

void LoRa_txMode() {
    LoRa.idle();
    LoRa.enableInvertIQ();
}

```

**Figura 70.** *Invert IQ Signals en el Gateway*

**Fuente:** *El investigador*

De esta forma el gateway solo leerá los mensajes de los nodos y nunca de otro gateway, y los nodos no leerán mensajes de otros nodos.

### Capa de aplicación del modelo OSI

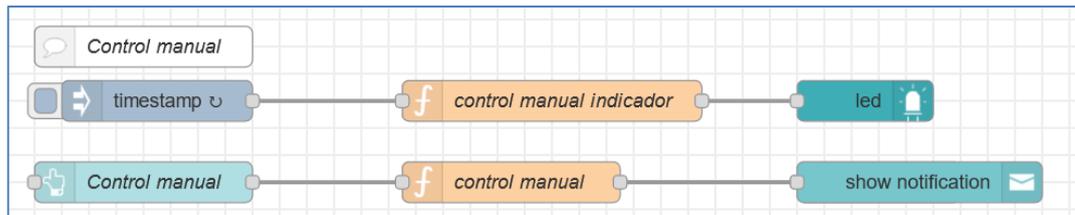
En la capa de aplicación se aplica otro método de seguridad para enviar las solicitudes de realizar diferentes acciones al sistema a través de una cadena de datos que solo el programador del sistema puede descifrar y comprender que quiere decir al enviar la cadena de datos. Estas peticiones pueden hacer solicitudes de solamente lectura de sensores o encendido o apagado de los actuadores. Una de las cadenas de datos que se envía puede ser 16irge1f, donde se hace la solicitud al nodo del invernadero que lea los sensores y envíe sus valores.

#### 3.1.2.3.8 Programación de Node-RED

##### Programación general del sistema

Una vez dentro Node-RED se procede a la programación del sistema, primero se configura el control manual que tendrá el sistema tal como se aprecia en la siguiente figura para ello se utiliza un nodo “inject” para inyectar un intervalo de tiempo, a continuación, se enlaza a una función “control manual indicador” el cual indicará si el

mensaje fue cargado o no, si el mensaje fue cargado se encenderá un LED del HMI. En el HMI también se coloca un botón el cual realizará la misma función de activar o desactivar el modo manual, de la misma forma encendiendo o apagando el LED. Si es que eso pasa se enviará un mensaje de “Control Manual Desactivado” o “Control Manual Activado”. En las figuras 71, 72 y 73 se puede observar la programación mencionada.



**Figura 71. Programación gráfica del Control Manual**

**Fuente: El investigador**

```

Name: control manual indicador
On Message:
1 | var values = global.get(["cm"]);
2 | if (values[0] == 1){
3 |     msg.payload = 1;
4 | }else{
5 |     msg.payload = 0;
6 | }
7 |
8 | return msg;

```

**Figura 72. Código del control manual indicador**

**Fuente: El investigador**

```

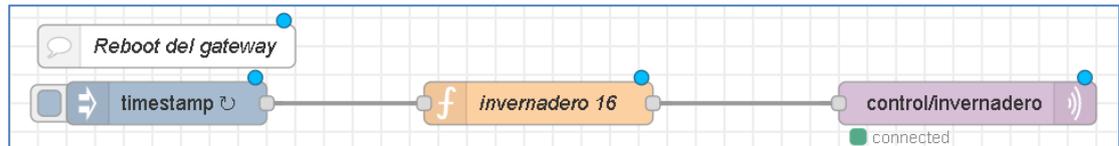
Name: control manual
On Message:
1 | var values = global.get(["cm"]);
2 | if (values==1){
3 |     values = 0;
4 |     msg.payload = "Control Manual Desactivado";
5 | }else{
6 |     values = 1;
7 |     msg.payload = "Control Manual Activado";
8 | }
9 | global.set("cm", values);
10 |
11 | return msg;

```

**Figura 73. Código del control manual**

**Fuente: El investigador**

Se programa un reinicio del gateway ya que en ocasiones es posible que se pueda perder la comunicación entre el Gateway y el servidor, como método de prevención y seguridad se deja programado un reinicio del Gateway, para ello se utiliza la función “invernadero 16” y un nodo “mqtt out” denominado “control/invernadero”, para que se produzca la conectividad y el módulo gateway proceda a reiniciarse cada intervalo de tiempo previamente establecido, en este caso se reiniciará todos los días a las 00:00h, como se aprecia en la figura 74, 75 y 76.



**Figura 74.** Programación gráfica para el reinicio del Gateway

*Fuente: El investigador*

**Figura 75.** Código del timestamp para el reinicio

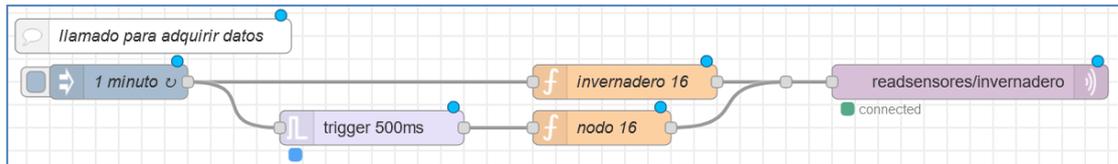
*Fuente: El investigador*

**Figura 76.** Código del “mqtt out” control/invernadero

*Fuente: El investigador*

Para la adquisición de datos de la misma manera se programa cada cierto lapso, en este caso el tiempo estimado es de un minuto, además se utiliza un “trigger” de 500ms para qué envíe dicha orden de adquirir los datos, en este caso se solicita los datos al módulo

Lora que se encuentra en el interior del invernadero, por medio de una función “invernadero 16” y “nodo 16”, finalmente se utiliza un “mqtt out” denominado “readsensores/invernadero” para proceder a leer los datos. A continuación, en la figura 77, 78, 79 y 80 se puede visualizar la programación de manera gráfica y sencilla, como se puede observar existe conectividad.



**Figura 77.** Programación gráfica para la adquisición de datos

*Fuente: El investigador*

```

1 msg.payload = "16irge1f";
2 return msg;

```

**Figura 78.** Código del invernadero 16

*Fuente: El investigador*

```

1 msg.payload = "16nrge1f";
2 return msg;

```

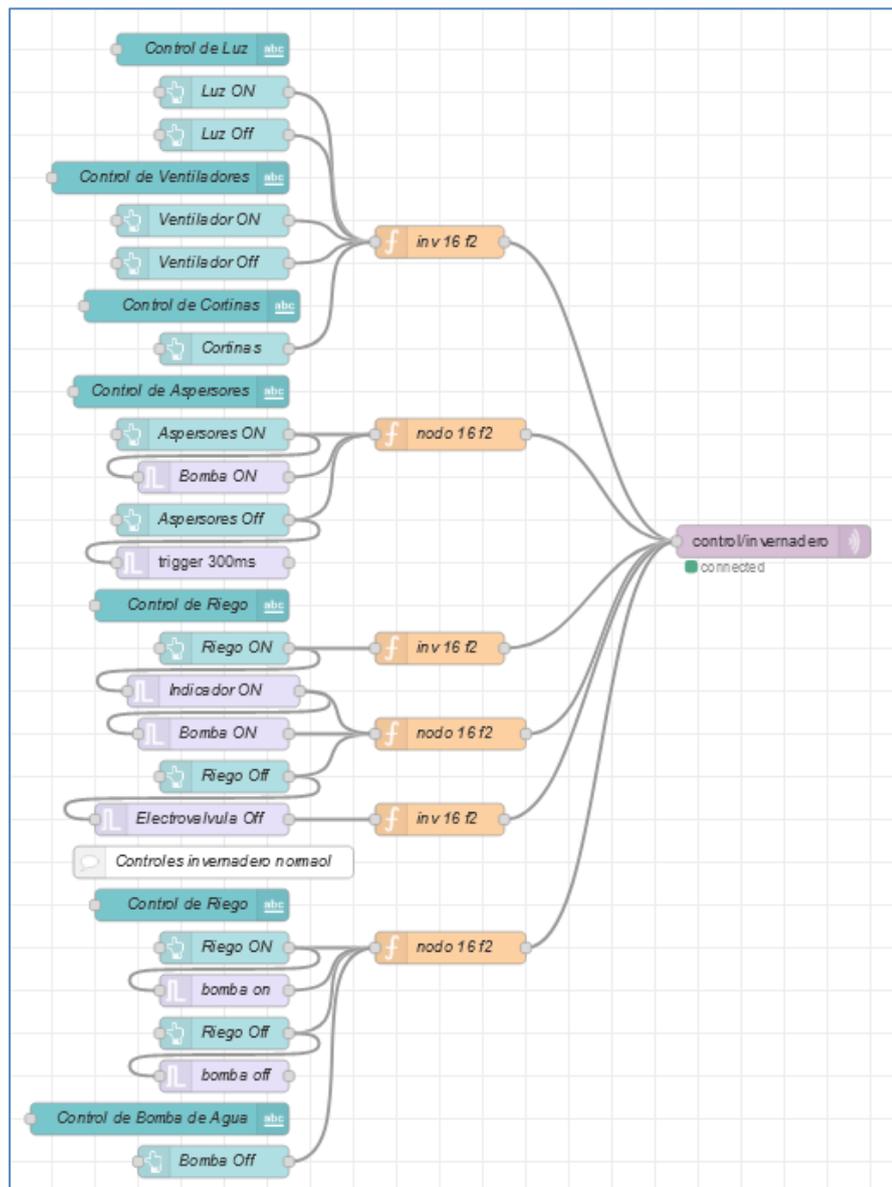
**Figura 79.** Código del nodo 16

*Fuente: El investigador*

**Figura 80.** Código del readsensores/invernadero

*Fuente: El investigador*

Una vez programada la adquisición de datos se procede a la programación del control de los diferentes actuadores, en la figura 81 se puede observar toda la programación en general desarrollada para el control. se utilizan nodos de texto, botones nodos desencadenantes, nodos de funciones y una salida MQTT para realizar la conexión. Como el propietario del invernadero cuenta con dos invernaderos los cuales trabajan como una sola bomba de agua, también se instaló una electroválvula en dicho invernadero para poder controlarlas desde el sistema y que no presente fallas en el momento de su funcionamiento.

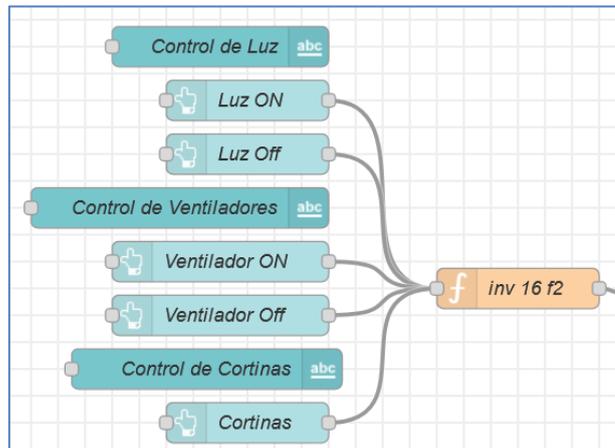


**Figura 81.** Programación visual del control de actuadores

**Fuente:** El investigador

## Control de luces, ventiladores y cortinas

La programación utilizada para el control de cada uno de los actuadores es muy similar tal como se puede apreciar en la figura 82, se utilizan nodos de texto, nodos de botones y nodos de funciones para poder conectar inalámbricamente.



**Figura 82.** Programación gráfica del control de luz y cortinas

**Fuente:** El investigador

En la figura 83 se puede observar el código que está en cada una de las funciones que se aprecia en la figura 81, el código es similar por lo que solo se explicara una de las funciones.

```

Name inv 16 f2
Setup
On Start
On Message
On Stop
1 var values = global.get(["cm"]);
2
3 if (values[0] == 1) {
4     msg.payload = "16irge2f" + msg.payload;
5 } else {
6     msg.payload = 0;
7 }
8
9 return msg;
```

**Figura 83.** Código de programación del nodo inv 16 f2

**Fuente:** El investigador

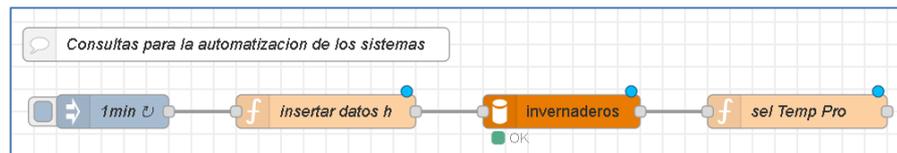
En la figura 83 se puede observar la programación del nodo función “inv 16 f2”, donde es posible apreciar que se envía una cadena de datos, en este caso en específico es “16irge2f”, el significado es el siguiente. Desde el inicio hasta la r está indicando para quien es el mensaje, en este caso es el identificador del nodo de la bomba, en otras palabras, el número es la dirección y a continuación está el id (i: nodo invernadero, n: nodo bomba, g: gateway), desde la r hasta la e, indica quien envía el mensaje, lo mismo

para el número y la letra de esa sección, desde la *e* a la *f* indica que tipo de acción envía, 1 para solicitar valores de los sensores, 2 para funciones de control.

### Programación de la base de datos

Para almacenar los datos adquiridos por los sensores del sistema de control y monitoreo se recurre a una base de datos, como se mencionó anteriormente en el presente informe se utiliza una base de datos de MySQL ya que es gratuita y de código abierto.

Es necesario hacer la conexión de nuestro sistema hacia la base de datos, por ende, se realiza la programación en node-red. Como se puede apreciar en la gráfica de la figura 84 el intervalo de tiempo para consultar los datos para el modo automático será de un minuto de la base de datos llamada invernadero.

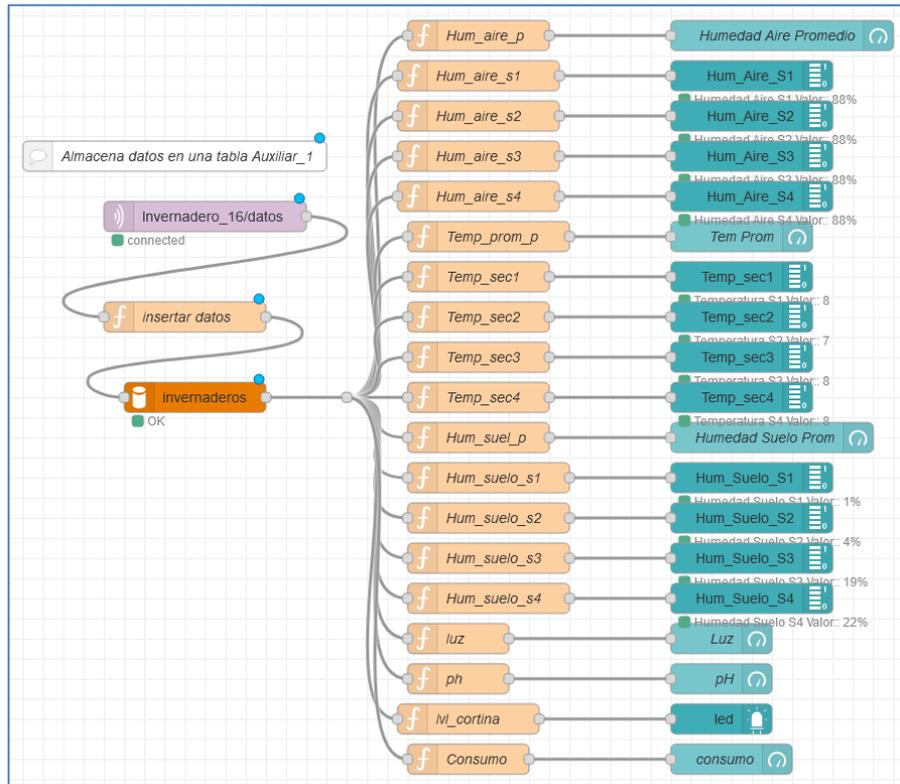


**Figura 84.** Programación gráfica de la consulta de datos

**Fuente:** El investigador

Para poder almacenar los datos de cada uno de los sensores se crea una tabla llamada auxiliar 1, donde se van guardando los valores sensados de cada sector del invernadero, como se puede apreciar en la figura 85 se saca un promedio de la humedad del suelo, humedad del aire, temperatura de ambiente, para poder visualizar a través de gráficas de una mejor manera y que el operario pueda tomar decisiones en base a un promedio de lo que sucede en el interior del invernadero. Además, la cantidad de radiación solar que reciben las plantas se puede visualizar también de forma gráfica en  $Wh/m^2$ , así como el pH y el consumo energético del sistema desarrollado.

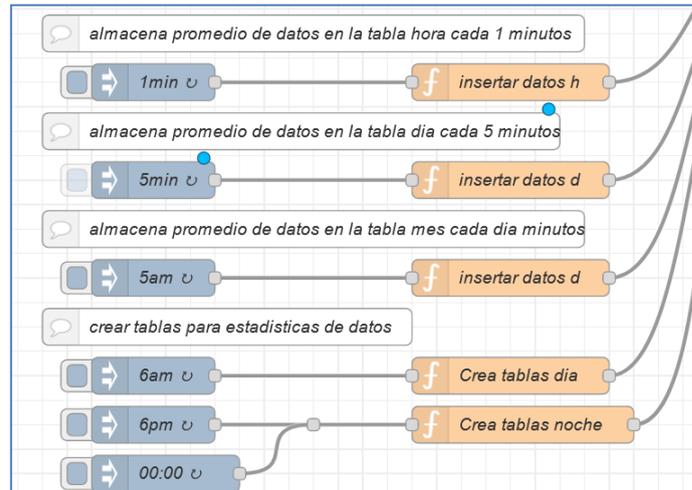
Con este diseño del sistema el operario podrá saber qué es lo que pasa en el interior de su invernadero en tiempo real, así como visualizar y analizar tablas y promedios de lo que va sucediendo gracias a la utilización de la base de datos, con ello se consigue realizar un mejor seguimiento a la plantación y poder tratarla en base a sus necesidades.



**Figura 85.** Programación gráfica del almacenamiento de datos y su graficado

**Fuente:** El investigador

Debido a la cantidad de información que recogen los sensores de cada sector es necesario crear tablas en las cuales se almacenen datos solamente cada 5 minutos ya que las condiciones dentro de un invernadero no cambian bruscamente, sino que su cambio es progresivo, debido a esto no es necesario almacenar datos en intervalos de tiempo más cortos. Otro aspecto muy importante a tener en cuenta es la necesidad de separar los datos en tablas diferentes tanto para el día como para la noche, ya que se pretende realizar un análisis minucioso de lo que ocurre en el interior del invernadero tanto en el día como en la noche, ya que si se promediarán los datos recogidos por los sensores durante las 24 horas los resultados serían erróneos puesto que en el interior de un invernadero durante el día las temperaturas son altas pero durante la noche tienden a bajar considerablemente debido a esto se realizan tablas diferentes como se puede apreciar en el siguiente gráfico. La tabla del día es desde las 6:00AM hasta las 5:59PM y la tabla de la noche es desde las 6:00PM hasta las 5:59AM, como se observa en la figura 86.

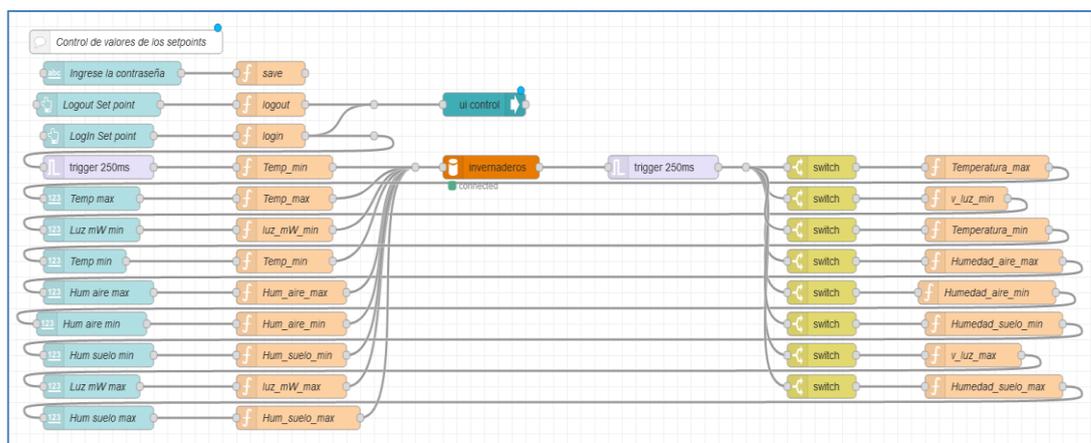


**Figura 86.** Almacenamiento de datos y creación de tablas para el día y la noche

**Fuente:** El investigador

Para el modo automático del sistema es necesario programar un *set point* el cual permita modificar los valores límites para cada variable sensada, esto se debe a que cada plantación necesita unas condiciones climáticas diferentes en el interior del invernadero, por tanto, el operario debe tener a disposición algo que le permita modificar dichos valores, para ello se procede a diseñar un *set point*, figura 87.

Lo primero que se tiene en cuenta es que no todos deben poder modificar dichos valores por tanto como método de seguridad se bloquea esta parte y solamente se permitirá el acceso a las personas que cuenten con el usuario y contraseña, con esto el operario se asegura que el modo automático estará funcionando de la manera que él lo decida.



**Figura 87.** Controles límites del Setpoint

**Fuente:** El investigador

### 3.1.2.3.9 Control, monitoreo y recepción de notificaciones a través de Telegram Creación de un Bot de Telegram

Para poder controlar el sistema de control y monitoreo desde Telegram primero se inicia creando un Bot, este Bot permitirá que el sistema reciba órdenes de parte del operario además de enviar alertas o notificaciones cuando las variables medidas en el interior del invernadero estén sobrepasando los límites o se produzcan cambios notorios.

El primer paso para crear el Bot es acceder a BotFather, para ello se debe escribir en el buscador de Telegram BotFather Y acceder a él, una vez dentro se envía el comando /start, BotFather mostrará una serie de comandos con las que es posible trabajar, para crear un Bot es necesario enviar el comando /newbot, una vez enviado el comando BotFather solicita ingresar un nombre para el bot en este caso es Invernadero Bot. Todo lo mencionado se puede apreciar en la figura 88.

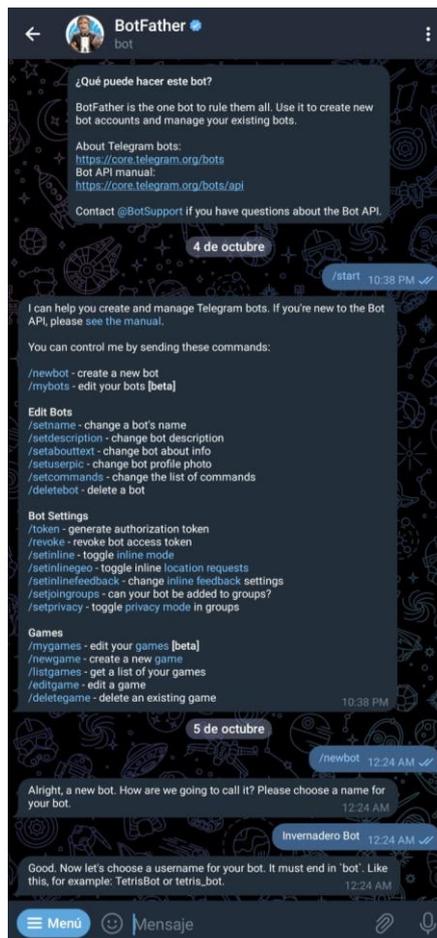
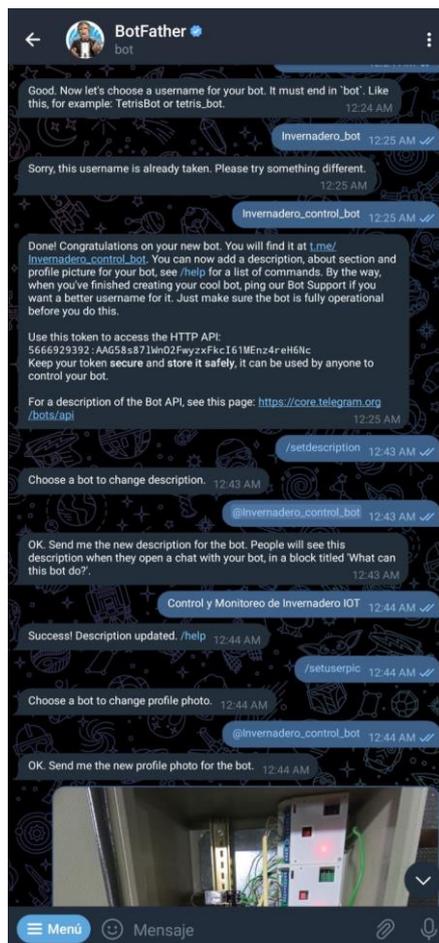


Figura 88. Creación de un nuevo bot

Fuente: El investigador

El siguiente paso es un nombre de usuario para el Bot, el nombre de usuario es Invernadero\_control\_bot, después solicita que se ingrese una descripción, primero es necesario seleccionar el bot en el cual se desea cambiar la descripción, por tanto, se selecciona el Bot @Invernadero\_control\_bot, una vez seleccionado se cambia la descripción a Control y monitoreo de invernadero y IOT. Para poder identificar mejor al Bot se procede a cambiar la imagen de perfil, El comando utilizado es /setuserpic, se vuelve a seleccionar el Bot y se envía la imagen para que se establezca como foto de perfil, tal y como se observa en la figura 89.



**Figura 89.** Cambio de descripción e imagen de perfil

**Fuente:** El investigador

El siguiente paso es ingresar los comandos que se enviarán para que el sistema realice diferentes acciones como encender o apagar los diferentes actuadores o poder visualizar datos en tiempo real del interior del invernadero. Todos los comandos utilizados se observan en la figura 90.



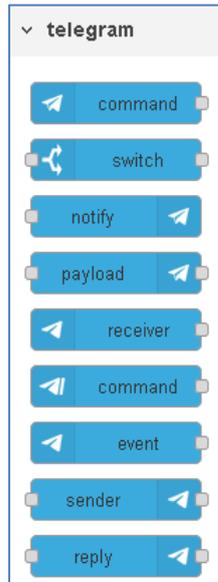
*Figura 90. Comandos de control*

*Fuente: El investigador*

Una vez creado el Bot el siguiente paso conectar Telegram con la programación de Node-RED.

### **Programación en Node-RED para control desde Telegram**

Node-RED permite realizar una programación en la cual se conecta a Telegram, la siguiente figura se puede observar las diferentes funciones y acciones que se permite realizar para poder programar como un Bot de Telegram, como se observa en la figura 91 los nodos disponibles son: dominio, switch, notificar, carga útil, receptor, dominio (que activa la salida cuando se recibe un comando específico del chat), evento, remitente, respuesta.



**Figura 91.** *Nodos de Node-RED para Telegram*

**Fuente:** *El investigador*

Para poder iniciar el Bot es necesario utilizar un nodo comando el cual se denomina start, figura 92. El propósito de este comando es iniciar al Bot de Telegram para poder empezar a trabajar en él, por tanto, se conecta a las funciones 24 y 25.



**Figura 92.** *Nodo Start*

**Fuente:** *El investigador*

Como se puede observar en la figura 93, la función 24 contiene el chatId proporcionado por Telegram, declara que va a ser un mensaje y el contenido será estado de funcionamiento: activo, en el caso que el usuario envíe el comando /start.

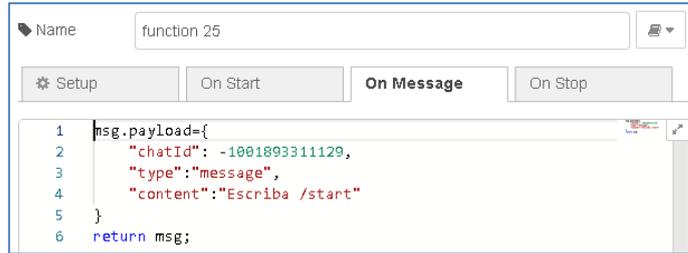
```

Name: function 24
Setup:
On Start:
On Message:
On Stop:
1 msg.payload={
2   "chatId": "-1001893311129",
3   "type": "message",
4   "content": "Estado de Funcionamiento: Activo"
5 }
6 return msg;
  
```

**Figura 93.** *Función 24*

**Fuente:** *El investigador*

Si el usuario envía un mensaje o comando diferente a /start el Bot responderá con Escriba /start como se puede observar en la figura 94.



```
1 msg.payload={
2   "chatId": "-1001893311129",
3   "type": "message",
4   "content": "Escriba /start"
5 }
6 return msg;
```

**Figura 94. Función 25**

**Fuente: El investigador**

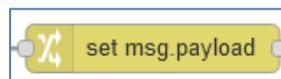
Finalmente, las dos funciones se enlazan a un nodo remitente el cual se denomina Telegram sender en el cual se menciona al Bot con el cual se va a trabajar en este caso es Invernadero\_control\_bot, tal y como se observa en la figura 95, con ello ya se está enviando el mensaje y existe la comunicación.



**Figura 95. Bot Invernadero\_control\_bot**

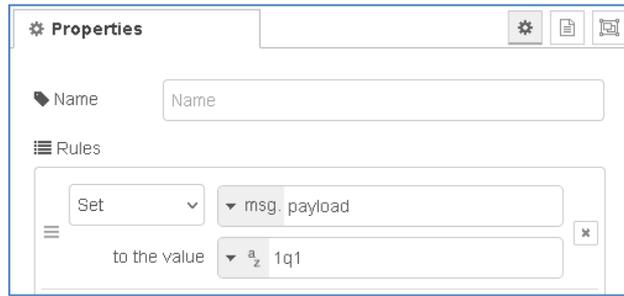
**Fuente: El investigador**

Para poder realizar el control de los diferentes actuadores es necesario utilizar los nodos switch (figura 96 ) estos permiten enviar datos de tipo string qué es lo que se requiere para poder encender o apagar los actuadores. En la figura 97 se observa que se utiliza una cadena de datos tal como 1q1, su significado es llamar al relé número 1 y ponerlo en estado 1, es decir encendido, en el caso del apagado se enviará un 1q0 y así sucesivamente para todos los casos de los diferentes actuadores ya que la lógica de programación que se utiliza es la misma.



**Figura 96. Nodo Switch**

**Fuente: El investigador**



**Figura 97.** Programacion del Nodo Switch

**Fuente:** El investigador

En el caso del control del riego, así como de los aspersores se utiliza un trigger, ya que es necesario activar primero la electroválvula y después encender la bomba de agua ya que si se encendiera al mismo tiempo podría causar daños de la bomba porque empezaría a enviar presión de agua sin que las electroválvulas se hayan activado, por ende, se utiliza un trigger de 300ms (figura 98). Una vez transcurrido este tiempo lo que hará es activar la bomba de agua.



**Figura 98.** Nodo trigger de 300ms

**Fuente:** El investigador

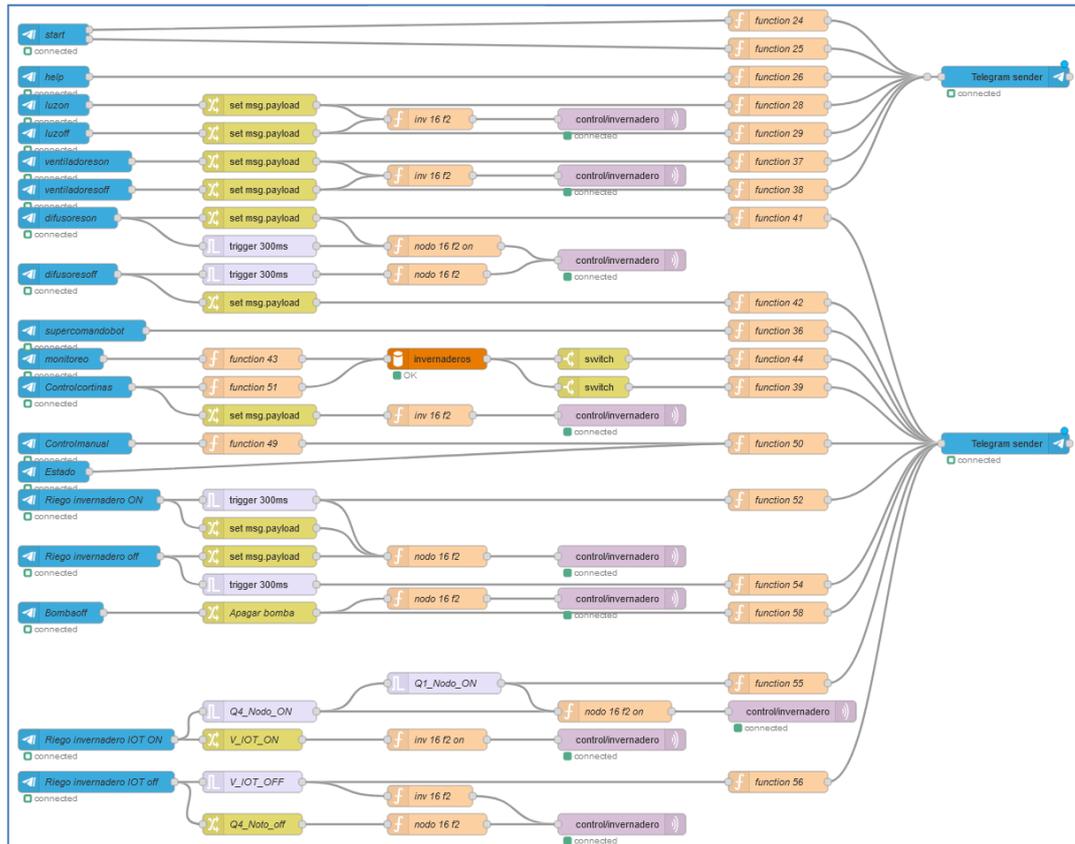
Para la función de monitoreo del invernadero donde el Bot debe enviar los promedios de los datos sensados en el interior del invernadero es necesario utilizar un nodo MySQL, como se observa en la figura 99, el cual se denomina invernaderos, con ello se tomarán los últimos promedios de la base de datos, para que el monitoreo sea en tiempo real.



**Figura 99.** Nodo MySQL, invernaderos

**Fuente:** El investigador

Como se mencionó anteriormente la lógica de programación para cada uno de los comandos utilizados dentro de la aplicación de Telegram es la misma por ende en la figura 100 se muestra toda la programación gráfica desarrollada en Node-RED para el control y monitoreo del invernadero a través de la aplicación de Telegram.



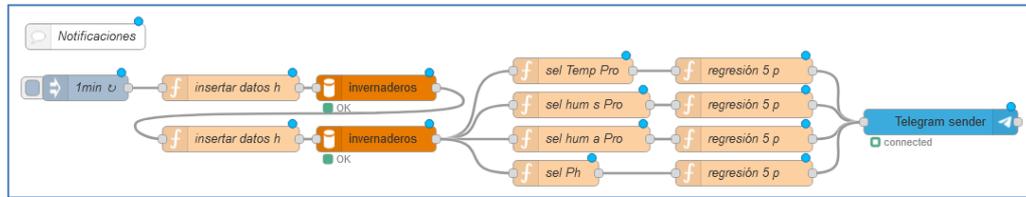
**Figura 100.** Programación gráfica para el control y monitoreo desde la app de Telegram

**Fuente:** El investigador

### Notificaciones de Telegram

Un factor importante dentro del sistema de control y monitoreo son las notificaciones automáticas cuando en el interior del invernadero se produzcan cambios notorios de las condiciones ambientales o a su vez las condiciones se encuentren por debajo o por encima de los límites estipulados, por ende, también es necesario realizar la programación para estas alertas.

Como se puede observar en la gráfica de la figura 101 cada minuto se estará monitoreando si es que se producen los cambios mencionados, para ello se consulta en la base de datos en cada uno de los promedios. Después de llamar a los promedios se utiliza una función en este caso denominada regresión 5 p, la cual se encarga de verificar si no se han producido cambios notorios, en el caso de que dichos cambios si se produzcan se enviará un mensaje o una notificación a través de Telegram.



**Figura 101.** Programación gráfica para el envío de notificaciones desde la app de Telegram

**Fuente:** El investigador

La programación de una de las funciones regresión 5 p se puede observar en la figura 102, dónde usando una sentencia if se realiza la comparación del valor promedio de la temperatura, en la cual se define que si la temperatura promedio sobrepasa los límites marcados se enviará una notificación de alerta a través de Telegram.

```

Name: regresión 5 p
On Message:
1  var rgtma = 28;
2  var rgtme = 16;
3  var y = [
4    msg.payload[0],
5    msg.payload[1]
6  ];
7
8  msg.payload = {
9    "chatId": "-1001893311129",
10   "type": "message",
11   "content": (
12     "Alerta" +
13     "\nTemperatura actual: " + y[1].toFixed(2)
14   )
15 }
16 var t_op = Number((y[1] >= rgtme) && (y[1] <= rgtma));
17
18 global.set("t_op",t_op);
19 if ((y[0] <= rgtma) && (y[1] > rgtma)) {
20   return msg;
21 } else if ((y[0] > rgtme) && (y[1] <= rgtme)) {
22   return msg;
23 } else if ((y[0] > 10) && (y[1] <= 10)) {
24   return msg;
25 }

```

**Figura 102.** Código del envío de notificaciones de alerta a través de Telegram

**Fuente:** El investigador

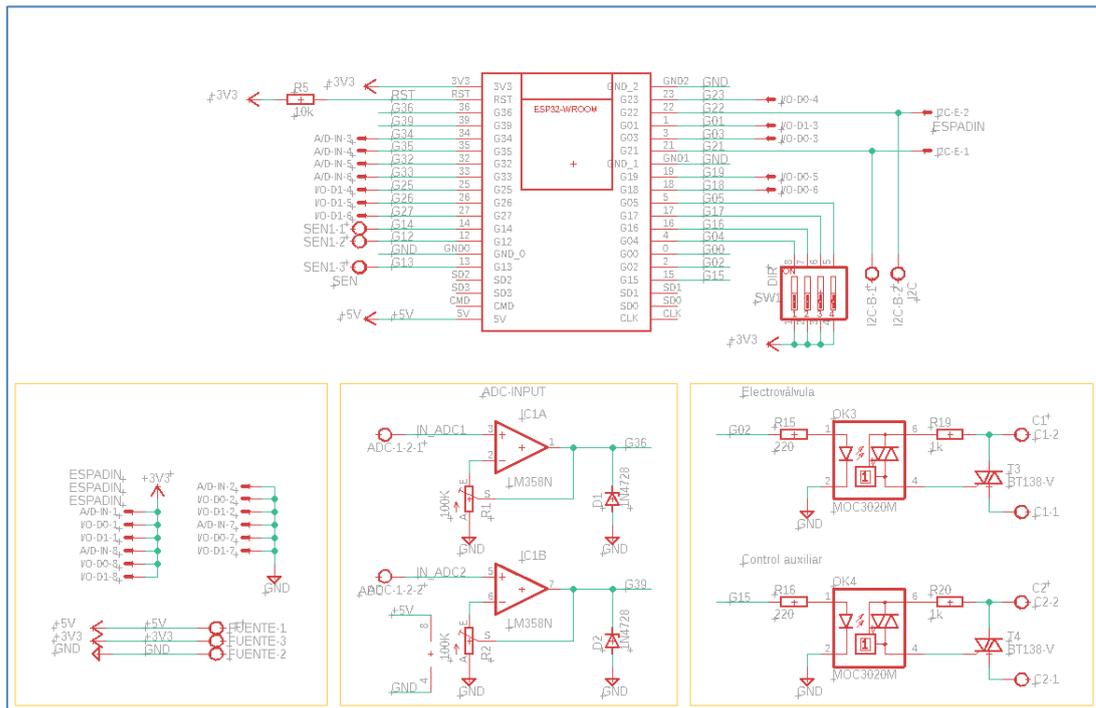
### 3.1.2.4 Implementación del sistema

#### 3.1.2.4.1 Diseño y armado de placas

La implementación del sistema permitirá probar ya de forma real todo el diseño realizado hasta este punto, con ello se pretende controlar de forma remota y automática diferentes actuadores dentro del invernadero con el fin de mejorar las condiciones climáticas para diferentes tipos de plantaciones, así como facilitar y agilizar el trabajo de los operarios.

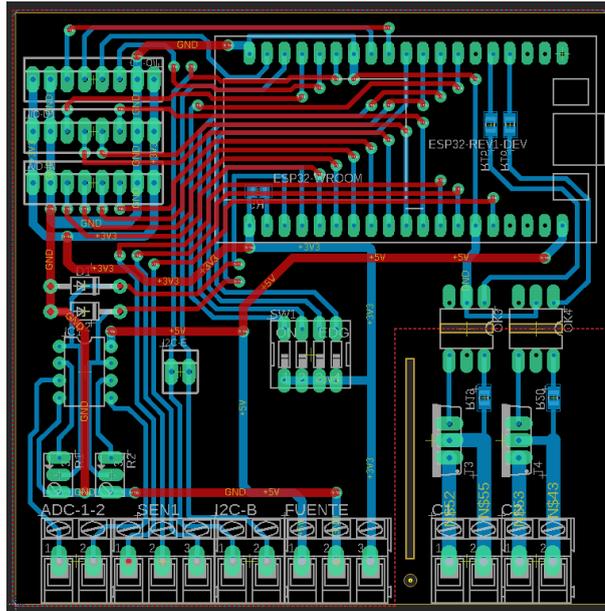
Se inicia con el diseño de 2 placas o baquelitas, una para los nodos sensores o ESP32 y la otra para los nodos Lora. Anteriormente ya se mencionó cómo está formada cada una de estas placas por donde a continuación se muestra gráficamente el diseño de la PCB.

Como se puede observar en la figura 103, 104 y 105 el circuito para los nodos sensores cuenta con una etapa para el acondicionamiento de la señal basado en el amplificador operacional LM358N. Además, cuenta con la etapa que permitirá controlar los actuadores donde se basa en la utilización del MOC3020M y del Triac BT138.



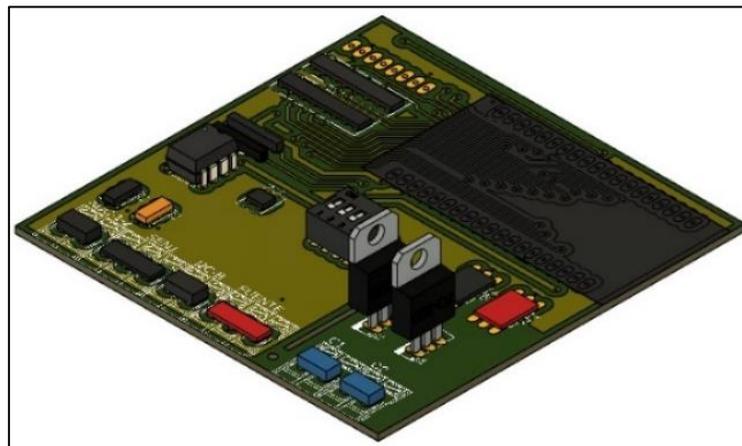
**Figura 103.** Circuito de los Nodos Sensores (ESP32)

**Fuente:** El investigador



**Figura 104.** PCB de los nodos sensores (ESP32)

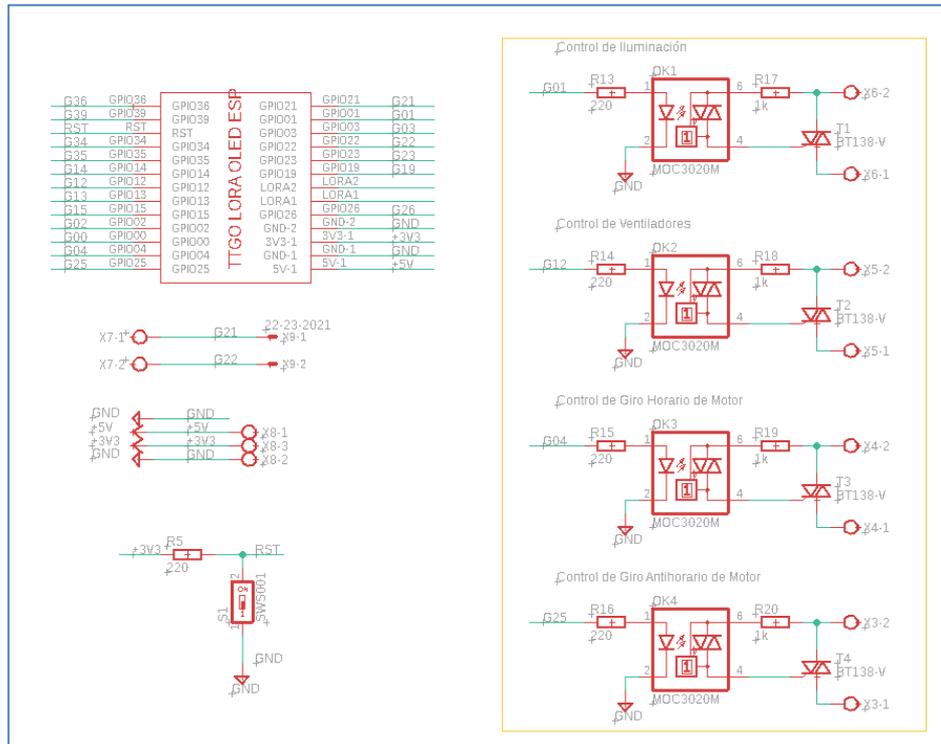
*Fuente: El investigador*



**Figura 105.** Vista 3D de la placa de los nodos sensores (ESP32)

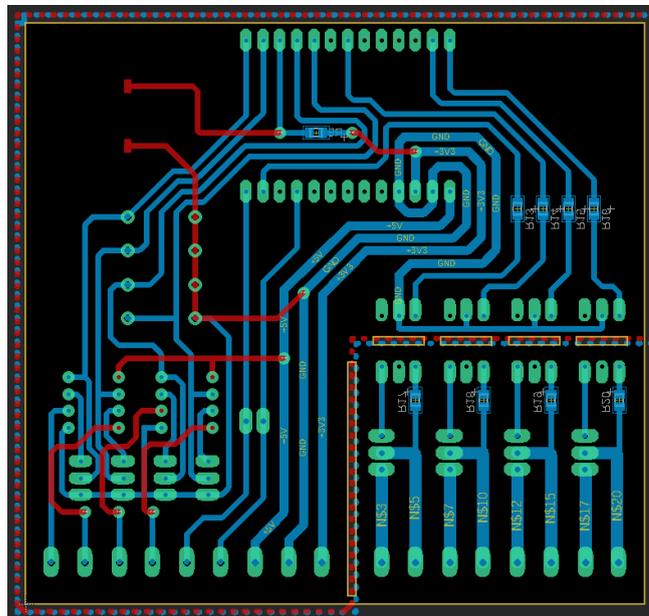
*Fuente: El investigador*

Para las placas de los nodos Lora simplemente se necesita las etapas donde se pueden accionar los diferentes actuadores por ende se puede observar en la figura 106, 107 y 108 el circuito donde se utiliza el MOC3020M y de Triac BT138.



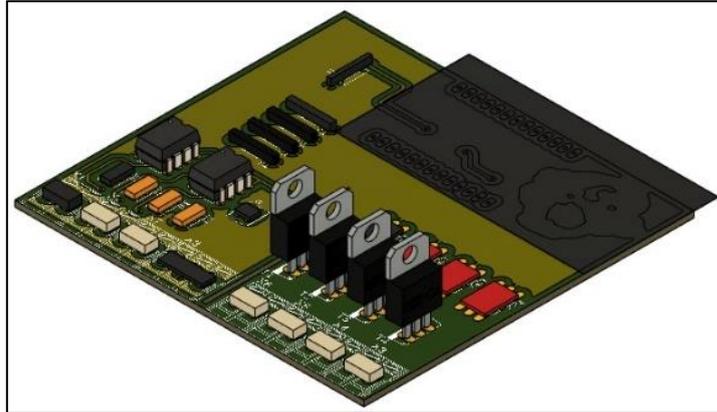
**Figura 106.** Circuito de los Nodos LoRa

*Fuente: El investigador*



**Figura 107.** PCB de los nodos LoRa

*Fuente: El investigador*



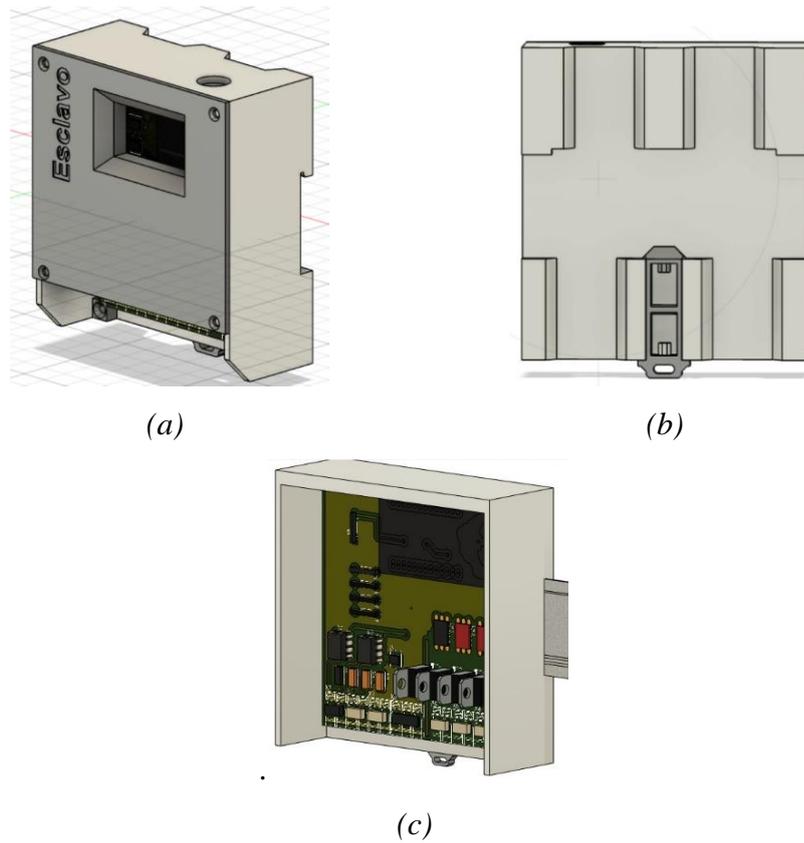
**Figura 108.** Vista 3D de la placa de los nodos LoRa

**Fuente:** El investigador

Cabe destacar que el sistema contará con cuatro nodos sensores y dos nodos Lora por ende las placas se reproducirán de manera similar cuatro y dos veces respectivamente.

Para proteger y cubrir las placas para los nodos sensores y nodos Lora, así como los sensores de pH, Sensores de humedad del suelo, sensores de voltaje y sensores de corriente, es necesario diseñar cajas de plástico para mayor seguridad y que su montaje tanto en tableros eléctricos cómo en campo sea sencillo y seguro. Para el diseño de cada una de estas cajas se utiliza el programa Autodesk Fusion 360.

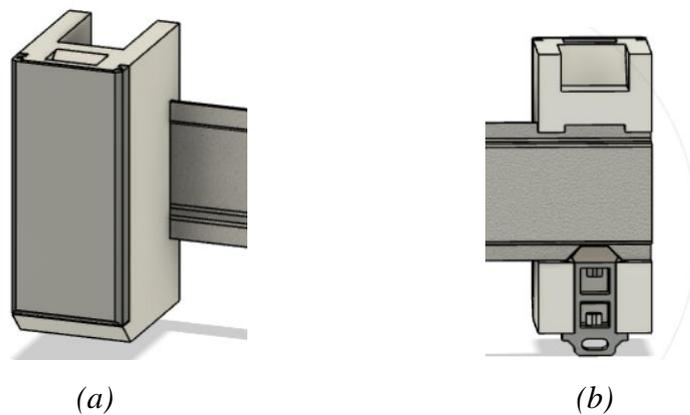
Para proteger los nodos sensores, donde se colocará las placas de las ESP32, y los nodos Lora donde se colocará los módulos Lora se diseña cajas cuadradas con un orificio central para poder visualizar la pantalla, además se deja un orificio en la parte superior para que pueda salir la antena y no pierda conectividad, como se puede observar en la figura 109 el diseño es de forma exacta para que encaje dichas placas y además se pueda fijar a los rieles que se colocan en el interior de los tableros del equipo.



**Figura 109.** Carcasa para placas de nodos sensores y nodos LoRa. (a) Vista frontal de la carcasa; (b) Vista trasera de la carcasa; (c) Carcasa con la placa

**Fuente:** El investigador

Para proteger los sensores de voltaje de la misma manera se diseña una caja rectangular (figura 110), la cual cuenta con orificios tanto en la parte superior como inferior para que el cableado pueda ingresar y salir, en la parte trasera se realiza un diseño el cual permita fijar la caja al riel.



**Figura 110.** Caja para sensores. (a) Vista frontal; (b) Vista trasera

**Fuente:** El investigador

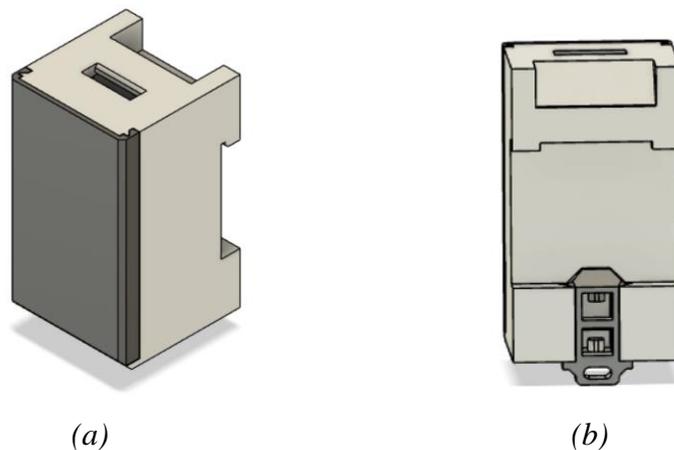
En la figura 111 se puede observar las cajas diseñadas para proteger a los sensores de humedad, estos se colocarán directamente en la tierra por ende el sensor debe quedar muy bien protegido, en la parte inferior tal y como se observa en la figura 111 se deja dos orificios necesarios para el sensor, así como en la parte frontal dos orificios cuadrados para poder observar los LED del sensor, con el fin de verificar si el sensor se encuentra conectado y trabajando de manera correcta.



**Figura 111.** Carcasa para sensores de humedad del suelo. (a) Vista frontal y lateral; (b) Vista lateral del interior

**Fuente:** El investigador

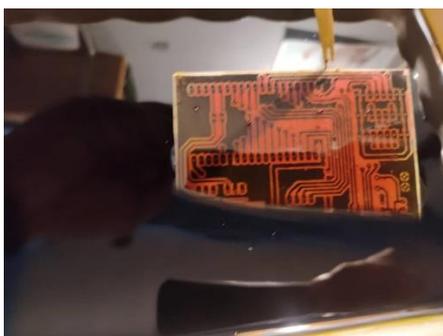
El sensor de pH cuenta con dos partes, una parte es la que ingresa al agua y tiene forma de esfera, y la otra parte es una tarjeta o placa electrónica la cual debe ser protegida del polvo o agua, en la figura 112 se puede observar el diseño de la caja.



**Figura 112.** Carcasa para sensor de pH. (a) Vista frontal y lateral; (b) Vista trasera

**Fuente:** El investigador

Una vez diseñadas las placas para los nodos se procede a imprimir y transferir a una baquelita, para ello se utilizó el calor de una plancha. Una vez transferido el circuito a la baquelita se introduce en agua caliente con ácido (cloruro férrico) para corroer el cobre excedente tal y como se observa en la figura 113.



**Figura 113.** Placa sumergida en agua con cloruro férrico

**Fuente:** El investigador

Una vez que se haya terminado de corroer el cobre excedente de cada una de las placas se seca y se limpia y el resultado es del que se puede observar en la figura 114, el siguiente paso es con algodón y tiñer (del inglés thinner) limpiar cada una de las placas para qué se pueda perforar las mismas.



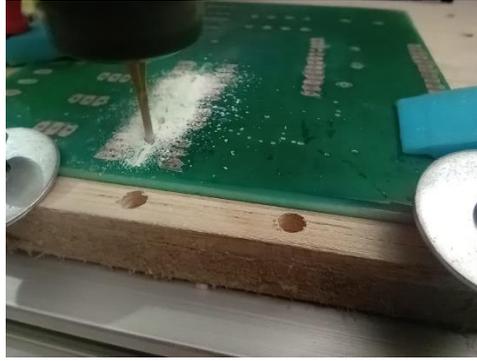
(a)

(b)

**Figura 114.** Placas corroídas. (a) Vista frontal; (b) Vista trasera

**Fuente:** El investigador

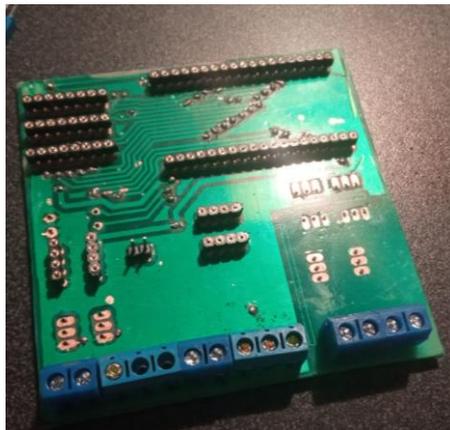
El siguiente paso es perforar cada una de las placas (figura 115), para ello se utiliza un taladro pequeño con una broca 0.8. es necesario tener precaución al momento de perforar debido a que se puede perforar en el lugar equivocado, por ello se recomienda utilizar un clavo y dar golpes suaves en cada uno de los lugares donde es necesario perforar con el fin de crear una pequeña guía para la broca.



**Figura 115.** *Proceso de perforación de placas*

**Fuente:** *El investigador*

Una vez que la placa ha sido perforada se procede a introducir los espadines, Borneras, resistencias y cada uno de los elementos que debe ser soldado directamente a la placa. Como se puede observar en figura 116, las placas ya están soldadas y se han colocado todos los elementos que forman parte de ella incluidos los módulos Lora y módulos ESP32.



(a)

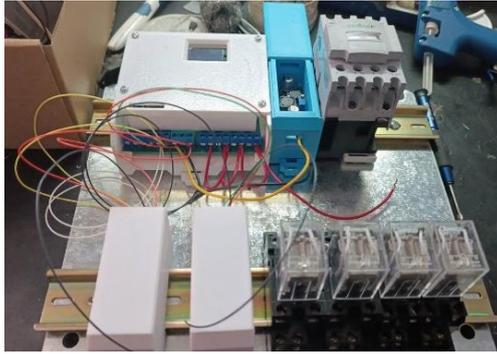


(b)

**Figura 116.** (a) *Placa lista para soldar; (b) Placas con todos los componentes*

**Fuente:** *El investigador*

Cuando ya todas las placas están soldadas se introduce cada una de ellas en sus respectivas cajas y se empieza a probar la posición adecuada que debe tener en el riel para que éstas encajen en los tableros eléctricos, como se puede observar en la figura 117 se intenta dar una colocación la cual permita cablear de forma sencilla ahorrando el mayor cable posible, para realizar unas conexiones eficientes.



(a)



(b)



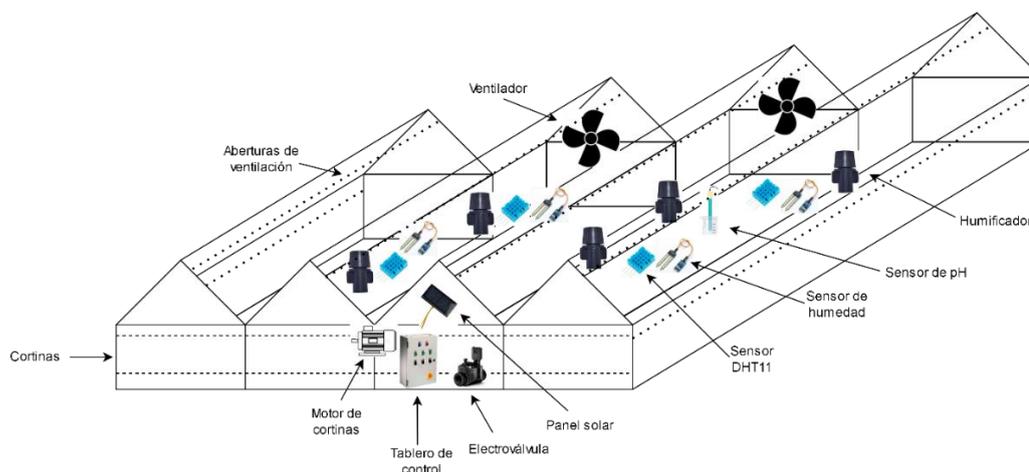
(c)

**Figura 117.** (a) Colocación de elementos en rieles 1; (b) Colocación de elementos en rieles 2; (c) Prueba de elementos en el tablero eléctrico

**Fuente:** El investigador

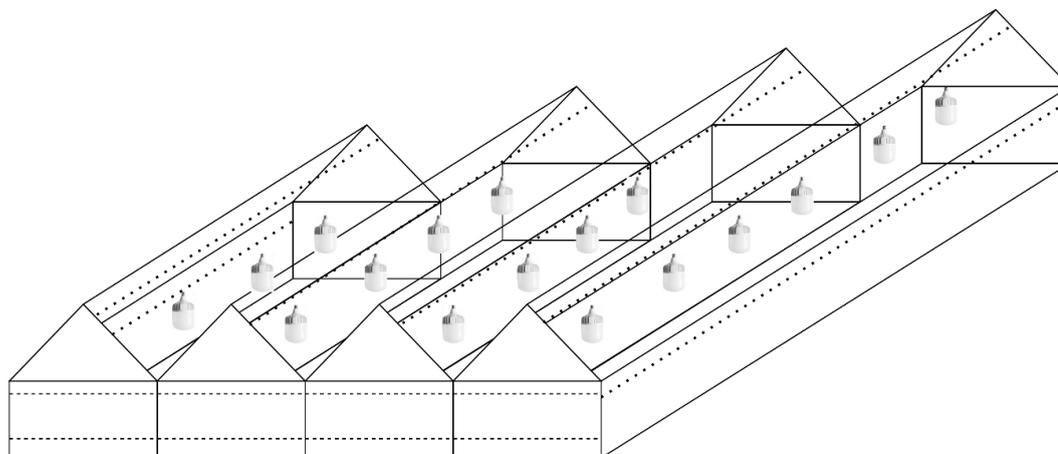
#### **3.1.2.4.2 Ubicación en el interior del invernadero**

Antes de iniciar con la instalación de cada uno de los elementos, dispositivos y nodos es necesario tener clara la ubicación de cada uno de eso, por ende, se realiza el gráfico de la figura 118, en el cual se determina su ubicación, con el propósito de cubrir toda el área del invernadero y que el sistema quede correctamente instalado mostrando datos reales para el operario. Como se puede apreciar en el siguiente gráfico se realiza la instalación de 2 ventiladores, 5 nebulizadores, cuatro sensores de humedad del suelo, cuatro sensores DHT11, un sensor de pH, un motor eléctrico, una electroválvula y un tablero eléctrico con sus diferentes elementos.



**Figura 118.** Ubicación de los diferentes elementos en el interior del invernadero  
**Fuente:** El investigador

En el gráfico de la figura 119 se puede observar la distribución de la iluminaria instalada en el invernadero, en total son 17 focos de alta potencia. Con la disposición que se puede observar en el gráfico se alcanza a cubrir toda el área del invernadero.

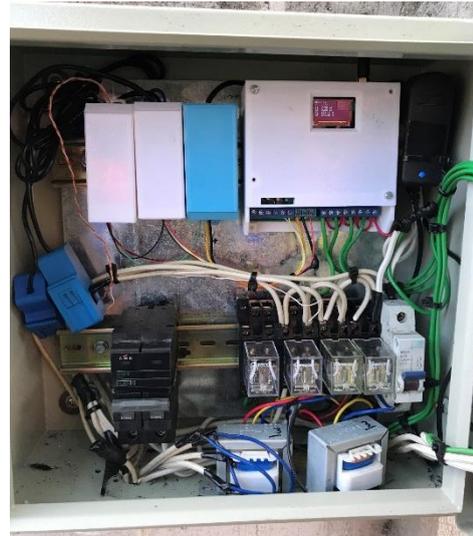


**Figura 119.** Ubicación de los focos de alta potencia  
**Fuente:** El investigador

El siguiente paso consiste en armar los dos tableros eléctricos para el sistema de control y monitoreo, el primer tablero es de unas proporciones de 40cm x 30cm, dicho tablero se colocará en el interior del invernadero, el segundo tablero es de unas proporciones de 30cm x 30cm este tablero se coloca en el área de la bomba, como se puede observar en la figura 120 todos los elementos necesarios se ajustan al tamaño de los tableros.



(a)



(b)



(c)

**Figura 120.** (a) Montaje de tablero eléctrico del invernadero; (b) Montaje de tablero eléctrico de la bomba; (c) Vista frontal del tablero de la bomba

**Fuente:** El investigador

Una vez armados los tableros es necesario realizar una serie de pruebas (figura 121) antes de alimentarlos con energía eléctrica, para ello se procede a medir conductividad en todas y cada una de las conexiones realizadas en el interior de los tableros, ya que si alguna conexión está mal podría ocasionar cortocircuitos y daños. Minuciosamente se realizan pruebas de conexión y ajuste en los diferentes nodos.



(a)



(b)

**Figura 121.** (a) Pruebas de conductividad en tablero de la bomba; (b) Pruebas de conductividad en tablero del invernadero

**Fuente:** El investigador

### **Instalación de elementos**

El invernadero en el cual se realiza el proyecto de investigación cuenta con cortinas cuya apertura es muy limitada por ende es necesario realizar una adaptación la cual permita una mayor apertura, con el fin de lograr que el invernadero tenga mayor ventilación y que el control de la temperatura en su interior sea eficiente. Como se puede observar en la siguiente figura las cortinas se cortan para que tengan una apertura de aproximadamente 2m, instalando un tubo de una pulgada de grosor en el cual las cortinas se enrollarán, debido a la longitud de la parte frontal del invernadero (30m) y al peso del plástico la adaptación de las cortinas se realiza por secciones, utilizan tubos g, chumaceras, rulimanes y varillas. Para poder enrollar y desenrollar las cortinas se utiliza un motor de portón eléctrico, ya que son motores que pueden desplazar bastante peso a muy poca velocidad qué es lo que requiere el sistema.

En la figura 122 se puede observar el trabajo de suelda de tubos en los cuales se va a enrollar la cortina, así como la suelda de los tubos G que van a servir de guía para las mismas.



(a)



(b)

**Figura 122.** (a) Proceso de suelda de tubos; (b) Proceso de suelda de tubos G

**Fuente:** El investigador

Para poder fijar el plástico al tubo de la figura 123 se utiliza pegamento de contacto y grapas, este proceso se realiza teniendo en cuenta que las medidas del plástico sean correctas ya que si hay errores el plástico al momento de enrollarse podría hacerlo de manera irregular provocando roturas en el mismo, ya que el motor al girar hará la suficiente fuerza como para romper el plástico en caso de no estar bien centrado.



**Figura 123.** Proceso de fijación del plástico al tubo.

**Fuente:** El investigador

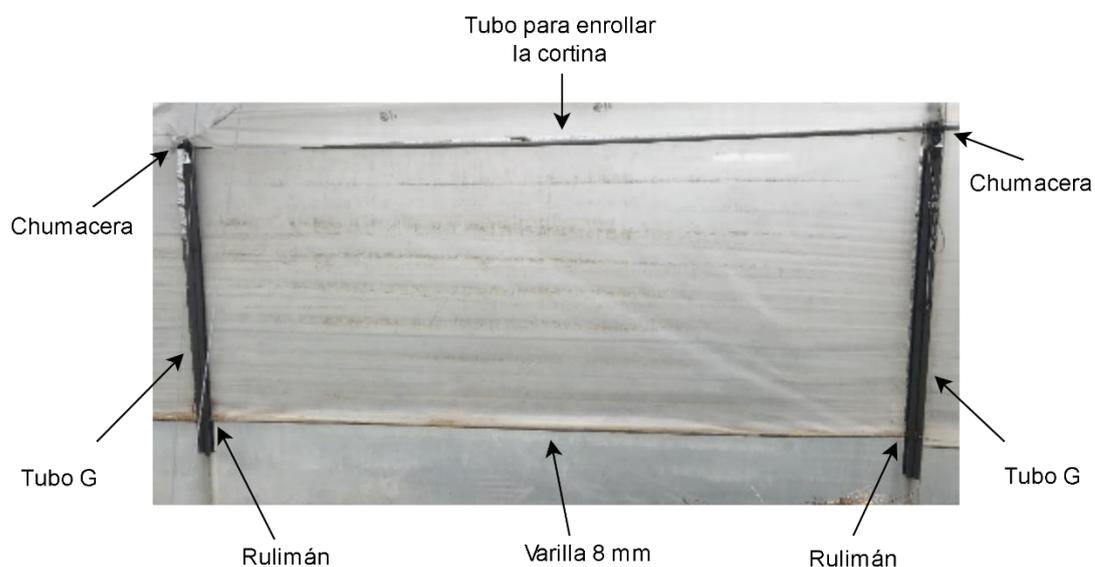
Una vez listas las cortinas se procede al montaje en el invernadero. La estructura del invernadero es metálica por tanto se procede a soldar y a fijar por medio de tornillos cada una de las chumaceras las cuales van a estar sujetando a los tubos donde se enrolla el plástico. En la figura 124 se puede observar el ajuste de las chumaceras.



**Figura 124.** Ajuste de chumaceras

**Fuente:** El investigador

En la figura 125 se puede observar cuáles son las partes de la cortina una vez ya adaptada, cómo es posible visualizar lleva diferentes elementos para que al momento de elevarla a través del motor no se produzcan roturas no deseadas del plástico.



**Figura 125.** Partes de la cortina

**Fuente:** El investigador

Al completar el montaje de las cortinas se pasa a la colocación del motor, primero es necesario crear una base en la cual se fija el motor y pueda realizar la fuerza necesaria para elevar las cortinas, la base tiene unas medidas de 30 cm de largo por 20 cm de

ancho, la base se suelda directamente a la estructura del invernadero para que tenga rigidez.

Una vez lista la base se procede al montaje del motor, para ello se adapta en el motor una polea de 2 in y se fija en el tubo una polea de 2 in y media, las poleas son fijadas tanto al motor como al tubo con un tornillo de 2 in con sus respectivas tuercas, como se observa en la figura 126.



(a)



(b)

**Figura 126.** Base para el motor; (b) Fijación del motor a la base

**Fuente:** El investigador

Al tener fijado el motor a la base se procede a realizar las conexiones de eléctricas del motor, figura 127. Las conexiones son:

- Alimentación a 110 voltios.
- Pulsador, el cual se utilizará para activar o desactivar el motor de forma remota.
- Finales de carrera, los cuales indican el estado de las cortinas, abierto o cerrado.



(a)



(b)

**Figura 127.** (a) Conexiones del motor; (b) Fijación de los finales de carrera

**Fuente:** El investigador

Se utiliza electroválvulas que funcionan con transformadores de 220Vac a 24Vac. Para la instalación de las electroválvulas (figura 128) es necesario tener en cuenta el sentido del flujo del agua, además se utilizan abrazaderas para fijarlas y que no pierda presión en ningún momento.

En total se utilizan 3 electroválvulas:

- Sistema de aspersores
- Sistema de riego del invernadero con el sistema de control y monitoreo
- Sistema de riego del otro invernadero con el que cuenta el agricultor.



(a)



(b)

**Figura 128.** (a) Instalación de electroválvula para el riego; (b) Instalación de electroválvula para aspersores

**Fuente:** El investigador

Para la instalación de los aspersores o nebulizadores se utiliza manguera de una pulgada y manguera de media pulgada, y como se puede apreciar en la figura 129 el nebulizador se coloca en el extremo de la manguera de media pulgada, para que quede a una altura de 1.70m. Además, se utiliza varilla para que el nebulizador quede de forma vertical. En el sistema se utilizan 5 en total.



(a)



(b)

**Figura 129.** (a) Instalación de nebulizadores. (b) Vista del nebulizador ya instalado

**Fuente:** El investigador

El sensor de pH (figura 130) se encargará de sensar en el agua que ingresa al invernadero a través del sistema de goteo, por ello se coloca un vaso plástico en la parte inferior de una de las mangueras de goteo para que cuando el sistema de riego se active dicho vaso almacene agua y el sensor pueda medir con facilidad, el vaso tiene unas perforaciones poco pronunciadas para que el agua se filtre y no se almacene durante intervalos de tiempo prolongados con el fin de que el valor sentido siempre sea del agua que está ingresando al invernadero en cada riego.



(a)



(b)

**Figura 130.** (a) *Instalación del sensor de PH;* (b) *Instalación de la placa del sensor de pH*

**Fuente:** *El investigador*

Los sensores de humedad del suelo se instalan a 5 cm del tallo de la planta, con el propósito de medir la humedad que tiene el área de las raíces de la planta que es por donde absorbe el agua, como se puede observar en la figura 131 el sensor se clava en el suelo. Se instalan 4 sensores de humedad de suelo en total, uno por sector.



**Figura 131.** *Instalación del sensor de humedad del suelo*

**Fuente:** *El investigador*

La instalación de los sensores de temperatura y humedad de ambiente se realiza a 1.60m del suelo. Ya que el invernadero cuenta con un sistema de microaspersión para generar humedad en el invernadero los sensores de humedad y temperatura deben ser cubiertos y protegidos como se aprecia en la figura 132. Se instalan 4 sensores de humedad y temperatura de ambiente en total, uno por sector.



**Figura 132.** *Instalación del sensor de humedad y temperatura de ambiente (DTH11)*

*Fuente: El investigador*

Para la iluminación del invernadero se utilizan 17 focos de alta potencia que funcionan a 220 voltios, se colocan a una altura de 2m para que su radio de alcance sea mayor y que las plantas no estén muy alejadas de dicho foco, con ello las plantas reciben la potencia de vatios necesarios para que su crecimiento sea adecuado y poder aumentar el tiempo de fotosíntesis. En la figura 133 se puede observar la instalación del cableado y de los focos.



(a)



(b)

**Figura 133.** (a) *Instalación del cableado para la iluminación;* (b) *instalación de iluminación*

**Fuente:** *El investigador*

La instalación de los ventiladores se realiza a 2.80m de altura (figura 134), los ventiladores trabajan a 220 voltios por lo que fue necesario cablear con cable de calibre 14 el cual permita trabajar correctamente a los ventiladores sin llegar a recalentarse. El total de ventiladores en el interior del invernadero son 2.



**Figura 134.** *Instalación de ventiladores*

**Fuente:** *El investigador*

Finalmente se procede a realizar el cambio de conexiones en el interior de la bomba de agua (figura 135) para que esta funcione a 220 voltios con el fin de reducir el consumo energético y que el sistema tenga una mayor eficiencia.



*Figura 135. Ajuste de conexiones en la bomba de agua*

*Fuente: El investigador*

### **3.1.3 Resultados y discusión**

Una vez finalizada la implementación de todo el sistema de control y monitoreo para el cultivo de hortalizas, se procede a realizar pruebas de funcionamiento en las que se ha obtenido los siguientes resultados.

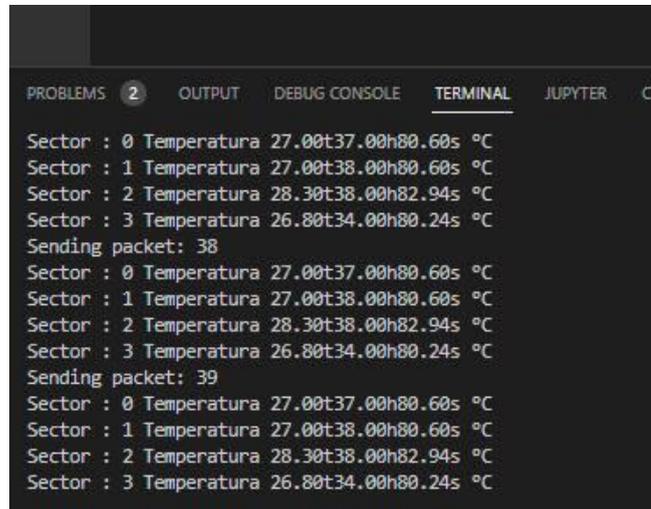
Para poder comprobar que el sistema está funcionando correctamente, es necesario probar cada una de las comunicaciones tanto alámbricas, como inalámbricas. Para ello se utiliza el monitor serie de Visual Studio, monitor serie de Arduino IDE y monitor de Node-RED.

#### **3.1.3.1.1 Comunicación entre nodos sensores y nodo LoRa**

Para poder verificar que la comunicación entre nodos sensores y nodo LoRa está funcionando correctamente y los datos están siendo enviados y recibidos adecuadamente se hace uso del monitor serie de Visual Studio. La comunicación entre los nodos sensores (ESP32's) y nodo LoRa del invernadero se realiza de forma alámbrica haciendo uso del protocolo I2C.

En la figura 136 es posible observar los datos que están siendo enviados desde los nodos sensores hacia el Gateway, se puede observar que se envían datos de los cuatro

sectores por separado. En el primer caso del sector 0 se está enviando 27°C de temperatura, 37,00% de humedad de ambiente y 80,60% de humedad del suelo. Para el proceso de pruebas de funcionamiento también se envían un Sending packet, que es un contador de paquetes enviados, para cerciorar que se están enviando correctamente.



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER CO
Sector : 0 Temperatura 27.00t37.00h80.60s °C
Sector : 1 Temperatura 27.00t38.00h80.60s °C
Sector : 2 Temperatura 28.30t38.00h82.94s °C
Sector : 3 Temperatura 26.80t34.00h80.24s °C
Sending packet: 38
Sector : 0 Temperatura 27.00t37.00h80.60s °C
Sector : 1 Temperatura 27.00t38.00h80.60s °C
Sector : 2 Temperatura 28.30t38.00h82.94s °C
Sector : 3 Temperatura 26.80t34.00h80.24s °C
Sending packet: 39
Sector : 0 Temperatura 27.00t37.00h80.60s °C
Sector : 1 Temperatura 27.00t38.00h80.60s °C
Sector : 2 Temperatura 28.30t38.00h82.94s °C
Sector : 3 Temperatura 26.80t34.00h80.24s °C
```

**Figura 136.** Comunicación entre nodos sensores y nodo LoRa del invernadero

**Fuente:** El investigador

### 3.1.3.1.2 Comunicación entre nodos LoRa y Gateway

Para poder probar la comunicación entre el nodo Lora del invernadero y el gateway se realiza mediante el programa visual Studio que permite correr la programación poder verificar que la comunicación se esté dando correctamente, como se puede observar en la gráfica de la figura 137 el nodo invernadero está recibiendo los datos de los nodos sensores a través del puerto I2C, se puede observar que es una trama de datos, en el nodo invernadero se clasifican los datos dice realiza una tabla, separando los datos de entre sectores, id, temperatura, humedad aire y humedad suelo, de modo que se puede observar que los datos lleguen correctamente al nodo invernadero, una vez aquí los datos se vuelven a agrupar en una trama para poder enviarlos hacia el nodo gateway.

```

Dato leído puerto I2C:
-> 24.40t38.00h24.49!24.20t40.00h75.56"24.50t36.00h76.10#24.00t34.00h75.20$
+-----+
|Sector| id |Temperatura|Humedad aire|Humedad suelo|
+-----+
| 0 | ! | 24.40 °C | 38.00 % | 24.49 % |
| 1 | " | 24.20 °C | 40.00 % | 75.56 % |
| 2 | # | 24.50 °C | 36.00 % | 76.10 % |
| 3 | $ | 24.00 °C | 34.00 % | 75.20 % |
+-----+
24.40t38.00h24.49!24.20t40.00h75.56"24.50t36.00h76.10#24.00t34.00h75.20$5.24w2.85p49.00c
Dato leído puerto I2C:
-> 24.40t38.00h25.10!24.20t40.00h75.56"24.50t36.00h76.10#24.00t34.00h75.20$
+-----+
|Sector| id |Temperatura|Humedad aire|Humedad suelo|
+-----+
| 0 | ! | 24.40 °C | 38.00 % | 25.10 % |
| 1 | " | 24.20 °C | 40.00 % | 75.56 % |
| 2 | # | 24.50 °C | 36.00 % | 76.10 % |
| 3 | $ | 24.00 °C | 34.00 % | 75.20 % |
+-----+
24.40t38.00h25.10!24.20t40.00h75.56"24.50t36.00h76.10#24.00t34.00h75.20$6.40w2.89p49.00c

```

**Figura 137.** Datos recibidos por el nodo LoRa

*Fuente:* El investigador

### 3.1.3.1.3 Comunicación entre Servidor, Gateway y nodos LoRa

Una vez que los datos se encuentran en el gateway es necesario verificar que haya comunicación entre el gateway y el servidor, para poder verificar esta comunicación se hace uso del software Node-RED y el monitor serie del Arduino IDE, para probar que la comunicación sea correcta se realiza una simulación entre el servidor, el Gateway y los nodos LoRa.

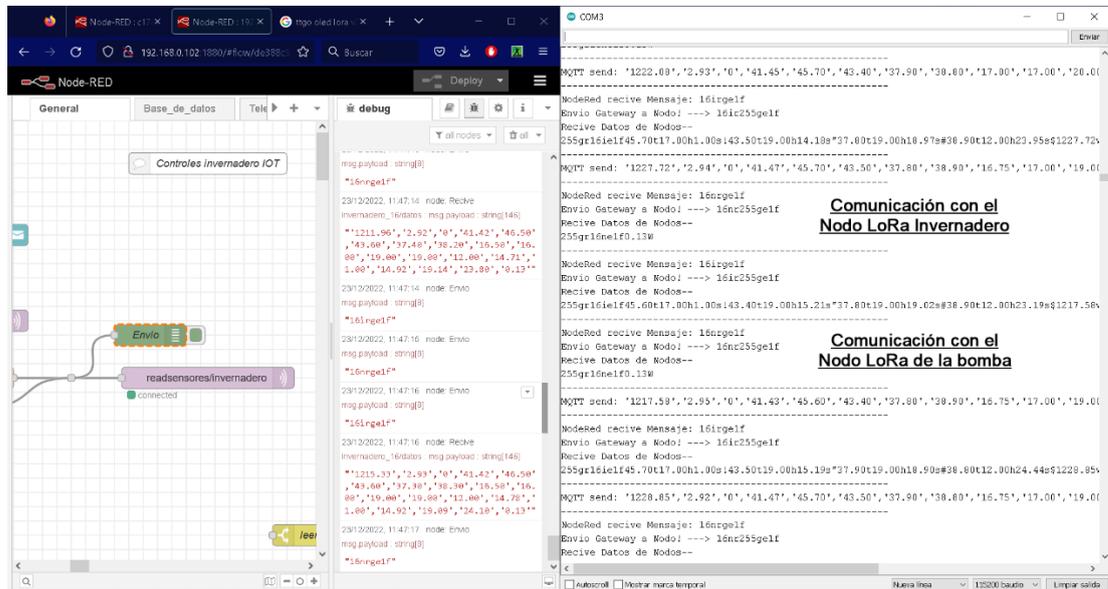
Como se puede observar en la figura 138 se realiza la simulación de enviar una solicitud de información desde el servidor hacia los nodos, para ello es necesario pasar a través del gateway.

En el primer caso se puede observar cómo desde el servidor se envía una cadena de datos “16irge1f”, en la cual se especifica que al nodo del invernadero se le está solicitando que realice la función de lectura de los sensores, como se observa en la gráfica Node-RED recibe el mensaje “16irge1f”, como se visualiza en el monitor serie del mismo, una vez aquí el mensaje es enviado desde el gateway hacia el nodo. Cuando el nodo Lora recibe esta solicitud a través de la cadena de datos de inmediato responde con los datos solicitados, tal y como se puede observar en el gráfico.

En el segundo caso se realiza la misma operación, pero solicitándole datos Al nodo

que se encuentra en la bomba para ello el mensaje de solicitud que se envía es “16nrgelF”, y como el caso anterior una vez recibida esta solicitud del nodo responder con los datos solicitados.

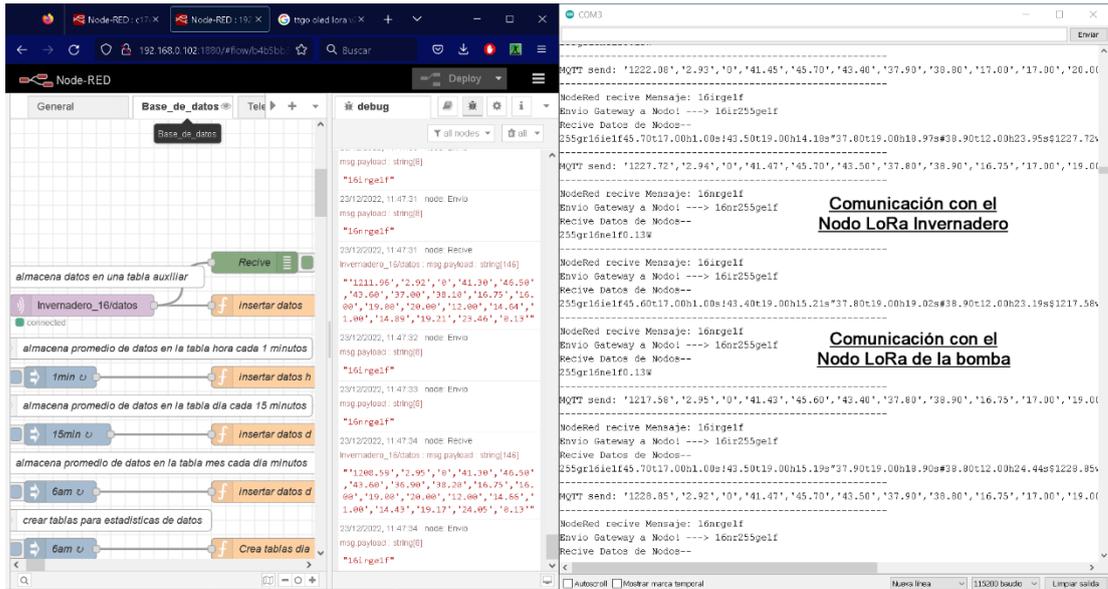
Además, en el gráfico es posible observar la trama de datos que se está enviando a través del protocolo MQTT para que estos puedan ser almacenados.



**Figura 138.** Comunicación entre Servidor – Gateway - Nodos LoRa (envía)

**Fuente:** El investigador

En la figura 139 se puede observar cómo se realiza la comunicación entre el servidor, Gateway y nodos Lora, pero en este caso desde la parte del nodo Lora del invernadero, es posible apreciar cómo el nodo recibe la solicitud de lectura de los sensores para posteriormente enviar los datos, se puede observar que todo el proceso de comunicación se realiza correctamente.

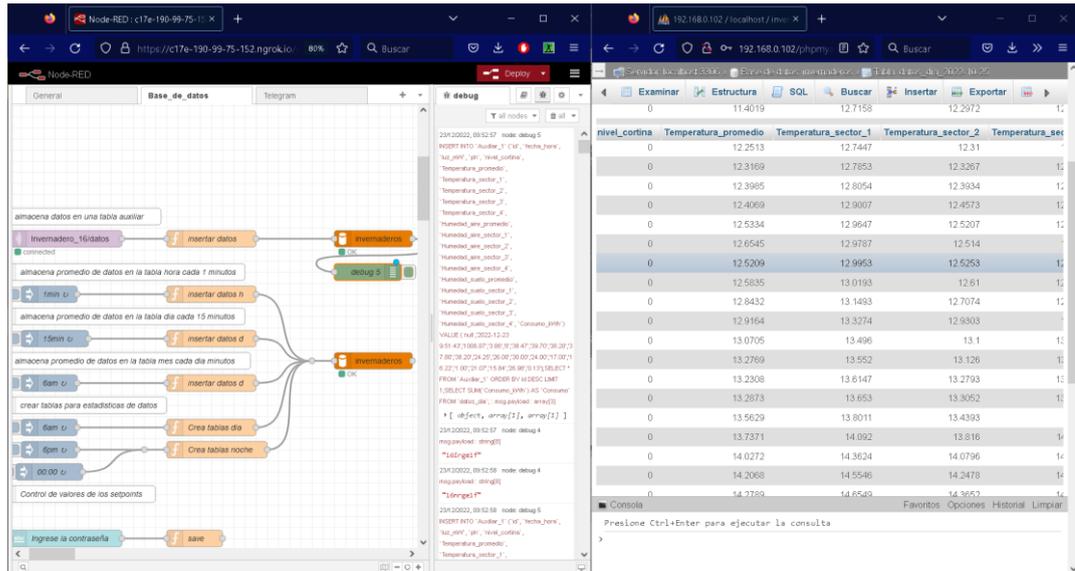


**Figura 139.** Comunicación entre Servidor – Gateway - Nodos LoRa (recive)

*Fuente: El investigador*

### 3.1.3.1.4 Comunicación entre Node-RED y base de datos

El sistema recopila datos del invernadero constantemente por ende es necesario almacenarlos en una base de datos, como se ha mencionado previamente en el presente informe se utiliza una base de datos en MySQL. Como se puede observar en la figura 140 los datos están siendo recopilados de los nodos sensores del invernadero, los cuales envían al nodo Lora del invernadero para que estos puedan ser almacenados en la base de datos, es posible apreciar como la comunicación está haciendo correcta al visualizar el monitor serie de Node-RED, los datos están siendo clasificados y enviados correctamente hacia la base de datos donde se están almacenando. Para poder ingresar a la base de datos se utiliza la dirección ip que brinda el Gateway, 192.168.0.102, Donde se le debe agregar /phpmyadmin, por tanto, la dirección para ingresar a la base de datos es: <http://192.168.0.102/phpmyadmin>. Cabe destacar que para que sea posible el ingreso es necesario estar conectado a la red local.



**Figura 140.** Comunicación entre Node-RED y MySQL

*Fuente: El investigador*

### 3.1.3.2 Pruebas y resultados

#### 3.1.3.2.1 HMI

Para el diseño del HMI se toman en cuenta 3 aspectos importantes, es necesario contar con un área donde se pueda monitorear el invernadero en tiempo real observando los promedios y lo que sucede en cada sector del invernadero, segundo el operario debe contar con un área desde la cual pueda controlar el invernadero tanto de forma manual como de forma automática y tercero debe existir una área en la cual se pueda observar estadísticas del invernadero, es decir donde se pueda analizar que ha pasado en los días previos u horas previas a la consulta de datos. Debido a lo mencionado el HMI se divide en 3 secciones cómo se puede visualizar en la figura 141.

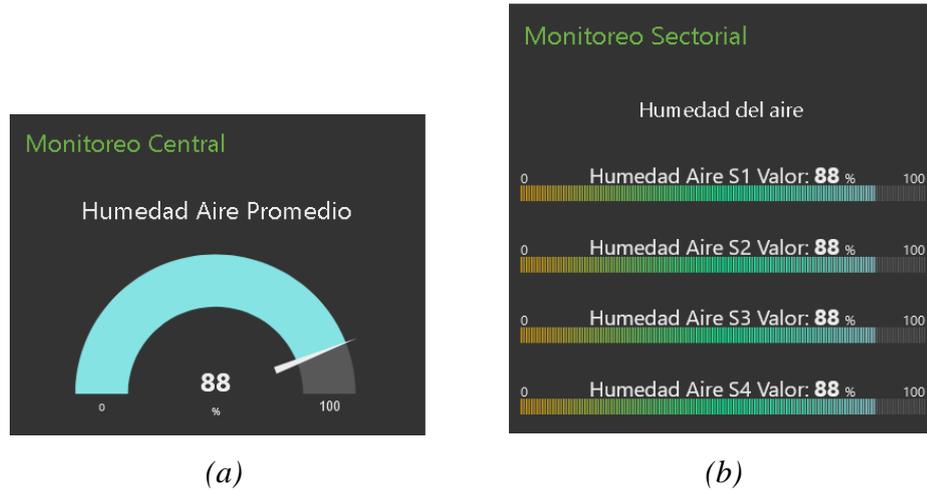


**Figura 141.** Secciones del HMI

*Fuente: El investigador*

### Monitoreo de invernadero

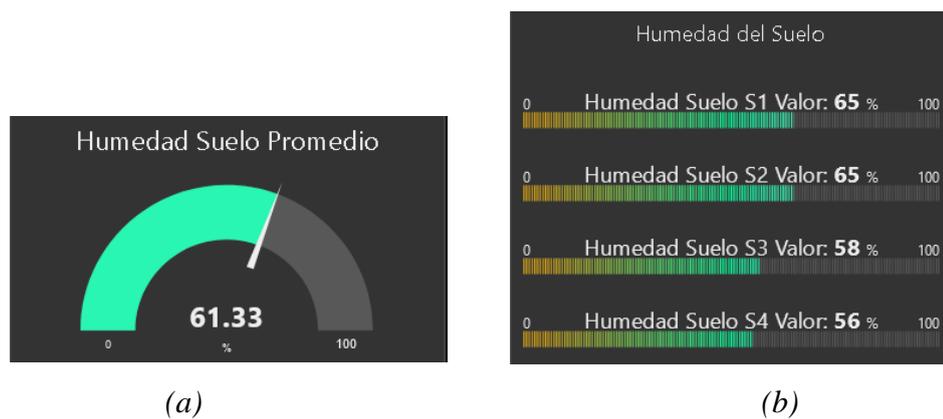
Dentro de la sección de monitoreo se diseña dos áreas diferentes, la primera será para observar los valores promedio y la segunda para observar los valores exacto de cada uno de los sectores del invernadero. En la figura 142 se puede observar un meter donde se observa el valor promedio de la humedad del aire, además de un gráfico de barras donde se observa el valor de la humedad del aire en cada sector.



**Figura 142.** (a) Monitoreo promedio de la humedad del aire. (b) Monitoreo sectorial de la humedad del aire

**Fuente:** El investigador

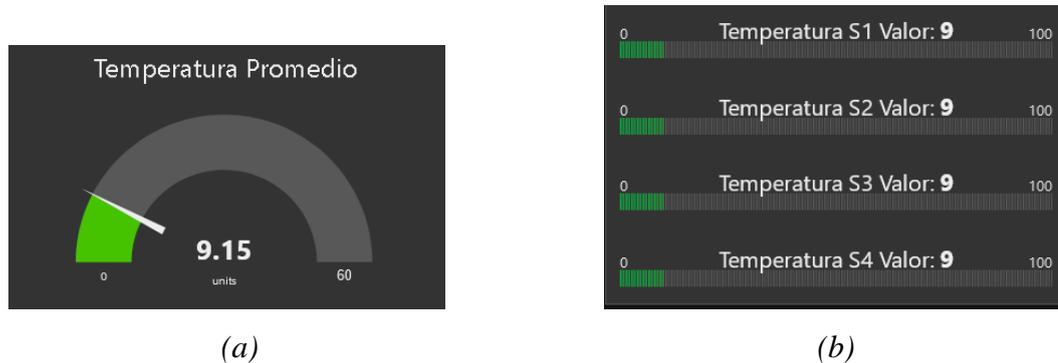
En la figura 143 se puede observar el valor promedio de la humedad del suelo, así como la humedad del suelo por sector.



**Figura 143.** (a) Monitoreo promedio de la humedad del suelo. (b) Monitoreo sectorial de la humedad del suelo

**Fuente:** El investigador

En la figura 144 se puede observar el valor promedio de la temperatura ambiente, así como la temperatura ambiente por sector.



**Figura 144.** (a) Monitoreo promedio de la temperatura ambiente. (b) Monitoreo sectorial de la temperatura ambiente

**Fuente:** El investigador

Para el monitoreo del pH se utiliza un meter (figura 145), ya que, se está midiendo en el agua que ingresa al invernadero a través del riego, por tanto, no es necesario medir por sectores como las anteriores variables mencionadas.



**Figura 145.** Monitoreo del nivel de pH

**Fuente:** El investigador

Para el monitoreo de la iluminación se utiliza un meter (figura 146), debido a que dentro del invernadero no hay techo que impida el paso de la luz en cualquier parte la radiación recibida por las plantas es la misma y no es necesario medir por sectores.



**Figura 146.** Monitoreo de la iluminación

**Fuente:** El investigador

El consumo del sistema es algo que se debe tener en cuenta ya que hoy en día lo que se pretende mejorar es la eficiencia en el consumo energético de cualquier sistema, Debido a esto varios actuadores se conectan a un voltaje de 220 voltios con el fin de reducir el consumo de corriente, provocando un ahorro para él propietario. Debido a esto el operario podrá observar en todo tiempo cuál es el consumo energético de su sistema, como se aprecia en la figura 147.



**Figura 147.** Monitoreo del consumo eléctrico

**Fuente:** El investigador

## Controles

La sección de controles que permite al usuario controlar de forma manual todo el sistema permitiendo activar y desactivar los actuadores en el momento que el usuario desee. para ello se debe activar el modo manual, además en esta sección se encuentra el *set point* el cual permite modificar los niveles mínimos y máximos para el control automático del invernadero, a continuación, se presenta gráficos donde se puede apreciar toda la sección de controles.

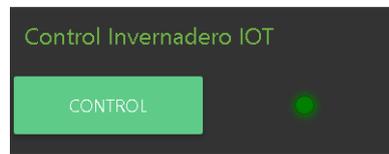
Primero se coloca un stop de emergencia para la bomba de agua, así como un control de riego donde permite encender y apagar dicho riesgo, este control será únicamente de emergencia en el caso de que en algún momento presentará algún error el sistema, como método de seguridad. En la figura 148 es posible visualizarlo.



**Figura 148.** Control General

**Fuente:** El investigador

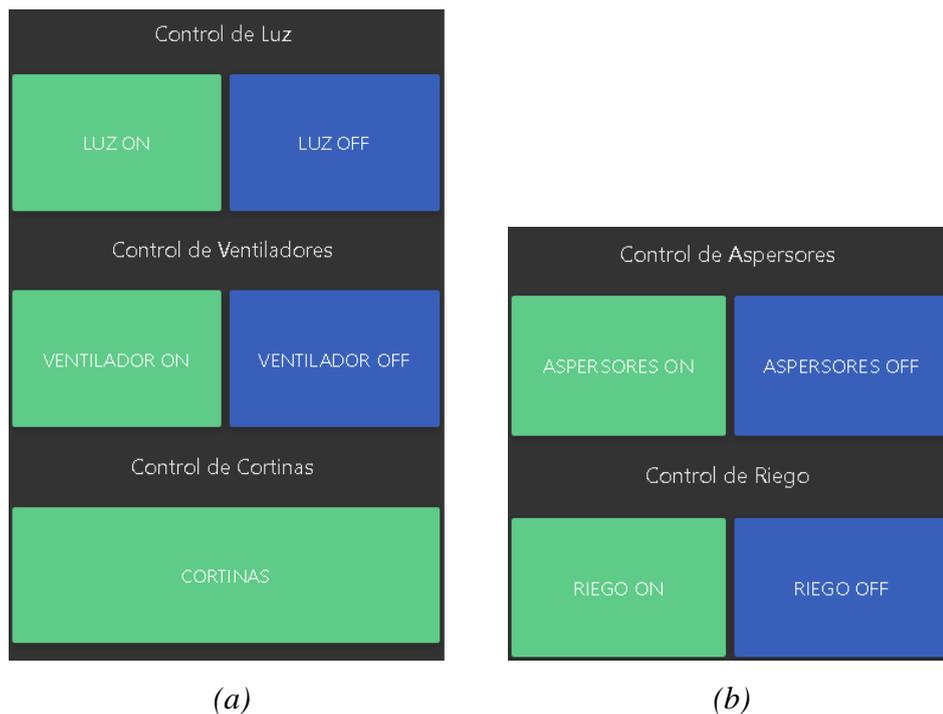
Para poder activar o desactivar el control manual se coloca un botón llamado control el cual está enlazado a un LED, cuando el LED está de color verde significa que el sistema está funcionando de forma manual, cuando el LED está en rojo significa que el sistema está funcionando de forma automática, tal como se observa en la figura 149.



**Figura 149.** Control Manual

*Fuente: El investigador*

Para poder controlar el sistema de forma manual se colocan botones de encendido y apagado de cada 1 de los actuadores por separado para que el operario tenga la opción y la libertad de encender y apagar cada vez que crea necesario, en la figura 150 se puede observar cada uno de los de controles.



**Figura 150.** (a) y (b) Control de actuadores

*Fuente: El investigador*

Para la parte del set point para el control automático del sistema se agrega seguridad ya que solamente el operario debe tener acceso para poder modificar ciertos valores dependiendo la producción del invernadero, para ello deberá ingresar una contraseña

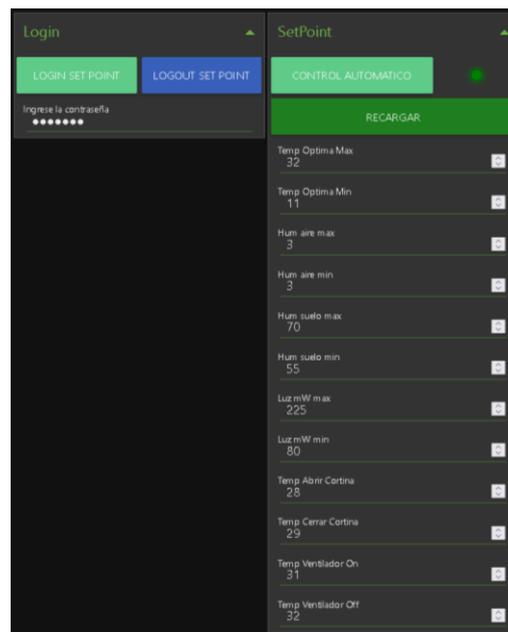
para poder loguearse y acceder al set point, en la figura 151 se puede observar los botones de ingreso y salida del login.



**Figura 151. Login**

**Fuente: El investigador**

Después de haber ingresado la contraseña se despliega el set point, como se puede ver en la figura 152 el set point se diseña para poder modificar los valores mínimos y máximos de cada una de las variables con las cuales está trabajando el sistema, con ello el sistema en modo automático encenderá o apagará los actuadores cuando llegué a los mencionados límites.

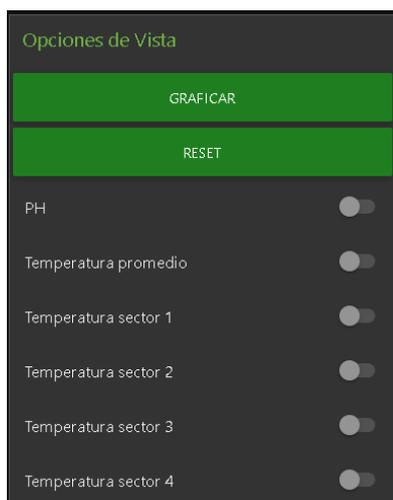


**Figura 152. SetPoint**

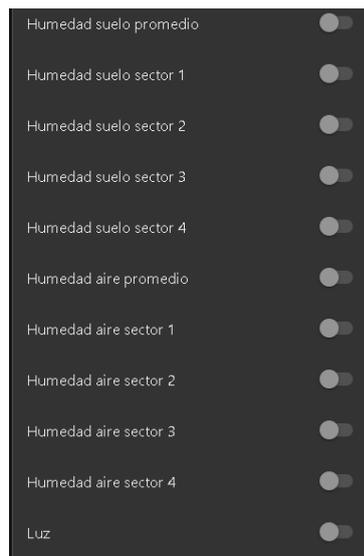
**Fuente: El investigador**

## **Estadísticas**

En la sección de estadísticas se podrán graficar los datos con los cuales trabaja el sistema, con el fin de que el operario pueda hacer un análisis de lo que está sucediendo en el invernadero a lo largo del día, como se puede observar en la figura 153 es posible graficar todas y cada una de las variables, así como los promedios.



(a)



(b)

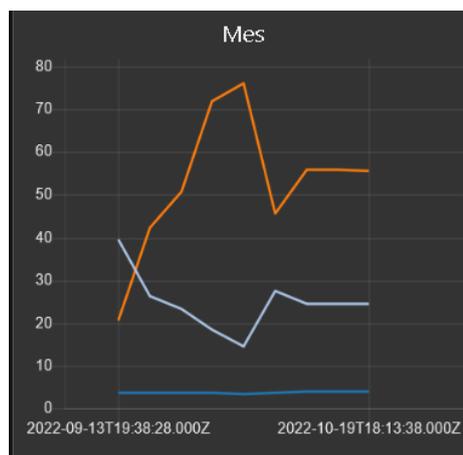
**Figura 153.** (a) y (b) Botones para graficar

**Fuente:** El investigador

Las gráficas que se utilizan son dos, la primera gráfica muestra los valores tomados a lo largo del día, así como los promedios, la segunda gráfica muestra los valores tomados durante todo el mes, con ello de manera fácil y rápida el operario podrá observar cómo evoluciona su plantación dependiendo de las condiciones climáticas que puede observar en el interior del invernadero, a continuación en la figura 154, se muestran las dos graficas.



(a)



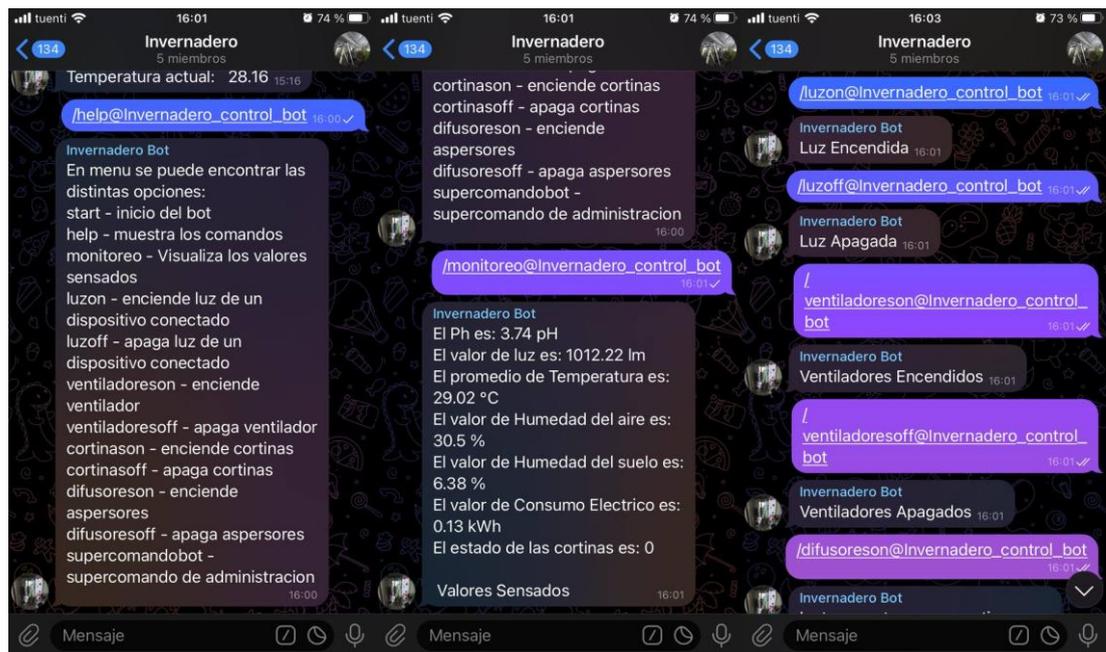
(b)

**Figura 154.** (a) y (b) Gráficas estadísticas

**Fuente:** El investigador

### 3.1.3.2.2 App de Telegram

El sistema de control y monitoreo trabaja juntamente con Telegram a través de la creación de un Bot, el sistema puede ser controlado y monitoreado a través de esta app, debido a que lo que se pretende es tener un control remoto total del invernadero, para brindar al usuario un sistema autónomo capaz de tomar sus propias decisiones, pero si es necesario controlar de forma manual también se integra esta función. A través de la creación de comandos en Telegram y con la ayuda de Node-RED se consigue comunicar la aplicación con el sistema, en a figura 155 se presentan los resultados:



(a)

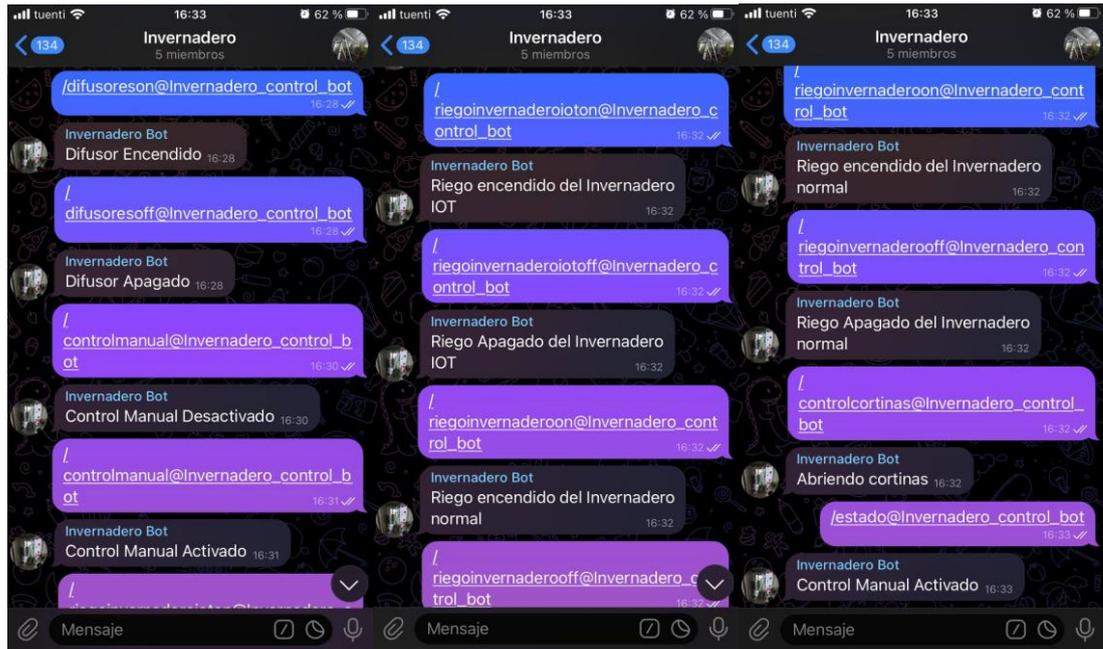
(b)

(c)

**Figura 155.** (a), (b) y (c) control y monitoreo del sistema a través de Telegram

**Fuente:** El investigador

El primer comando que se agrega es help, este comando muestra todas las opciones de control que brinda el sistema con su respectiva descripción. Como se puede observar en las figuras anteriores el control manual a través de la aplicación Telegram funciona correctamente, al enviar cada uno de los comandos el sistema responde y envía órdenes para que los diferentes actuadores puedan encender o apagar, de la misma manera como se puede observar se incorpora un comando de monitoreo el cual muestra los promedios de las diferentes variables medidas en el interior del invernadero, así como el estado de las cortinas donde cero significa cerrado y 1 significa abierto. Cabe destacar que estos valores son tomados de la última fila de cada una de las tablas de la base de datos de MySQL, es decir el monitoreo es en tiempo real.



(a) (b) (c)

**Figura 156.** (a), (b) y (c) control del sistema a través de Telegram

**Fuente:** El investigador

Se observa en la gráfica (c) de la figura 156 que en el sistema se agrega a un comando el cual activa y desactiva el control manual, cuando está activado el sistema dejará de funcionar automáticamente y esperará órdenes del usuario, en este caso el usuario contará con la ayuda de un sistema de alertas el cual le permitirá estar en continuo conocimiento de lo que sucede en el invernadero además del comando monitoreo, cuando el control manual está desactivado el sistema de control y monitoreo funciona de manera automática.

Además de controlar y monitorear el sistema, se agregó un sistema de alertas (figura 157) para que el usuario que tenga un conocimiento total de lo que pasa en el invernadero, estas alertas se envían cuando los promedios de las variables sensadas superan el máximo o mínimo impuesto por el usuario, o a su vez cuando se está produciendo variaciones donde el operario debe tener pleno conocimiento, en esos casos recibirá mensajes de alerta.



**Figura 157.** Mensajes de alerta en la app Telegram

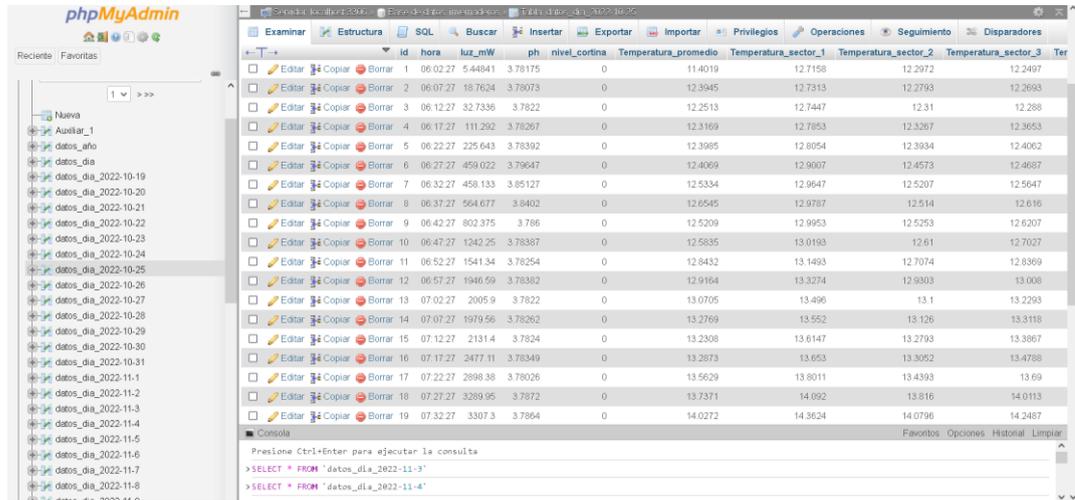
**Fuente:** El investigador

Como se puede observar en la figura 157 la humedad del aire se encuentra por encima del valor promedio ideal, por ende, se envía un mensaje alerta, además la temperatura se encuentra en el valor promedio adecuado por lo que se envió un mensaje de alerta para que el usuario al detener el modo manual activado proceda a abrir las cortinas, ya que son las 9:30 h y la temperatura tiende a incrementar con el paso de las horas.

### 3.1.3.2.3 Base de datos

La base de datos utilizada para el presente proyecto es phpMyAdmin, como se observa en la figura 158 los datos de están siendo guardados cada 5 minutos, con ello se puede comprobar que el funcionamiento del sistema es correcto y el usuario ya cuenta con formación para poder analizar el progreso de sus cultivos. la cantidad de datos recogidos a lo largo del día son 146 de cada una de las variables medidas, con ellos se puedes sacar promedios y poder analizar el funcionamiento del sistema de mejor manera. Una de las cosas a tomar en cuenta es que para realizar los promedios c programa dos tablas una tabla almacena los datos del día desde las 6 AM hasta las 6:00 PM, y la otra tabla almacena los datos desde las 6:00 PM hasta las 6 AM, con ello se logra evidenciar datos más cercanos a la realidad puesto que las condiciones climáticas durante el día y la noche presentan variaciones muy notorias, por lo que al necesitar

analizar datos reales se divide entre el día y la noche. En la figura 158 se puede evidenciar los datos almacenados del día 25 de octubre del año 2022.



**Figura 158.** Base de datos MySQL

*Fuente: El investigador*

### 3.1.4 Adquisición de datos con el sistema funcionando solo para monitoreo

En la tabla 25 se puede observar los datos sensados durante las dos primeras semanas de instalación del sistema de control y monitoreo. Las dos primeras semanas se configura el sistema para que realice el sensado de las diferentes variables y poder realizar una comparación entre el antes y después de la instalación del sistema.

**Tabla 25.** Datos sensados. Semana 1 y 2

DÍA	Radiación (Wh/m <sup>2</sup> )	pH	Temperatura promedio (°C)	Humedad aire promedio (%)	Humedad suelo promedio (%)	Consumo (kW/h)	Producción (Sacos de 12kg)	
<b>SEMANA 1</b>								
16/11/2022	3649,92	4,09	23,44	68,45	50,68	0,00	13	
17/11/2022	3730,29	4,07	27,61	52,76	46,29	0,00		
18/11/2022	3680,38	4,04	28,74	54,55	40,98	0,00		
19/11/2022	3789,33	4,30	33,01	39,92	48,63	1,35		
20/11/2022	3817,44	3,92	32,40	43,29	60,66	0,00		
21/11/2022	3854,23	3,65	31,63	40,97	58,39	0,00		
22/11/2022	3583,38	3,66	32,05	43,84	54,40	0,00		
<b>PROMEDIO</b>	<b>3729,28</b>	<b>3,96</b>	<b>29,84</b>	<b>49,11</b>	<b>51,43</b>	<b>∑ = 1,35</b>		
<b>SEMANA 2</b>								
23/11/2022	3921,29	3,73	31,71	45,27	52,76	0,00		
24/11/2022	3821,79	4,17	27,25	49,69	56,56	0,00		
25/11/2022	3841,40	4,16	26,72	52,95	54,11	1,22		

26/11/2022	3754,24	4,12	26,64	48,93	52,69	0,00
27/11/2022	3992,47	4,14	30,53	44,23	53,46	0,00
28/11/2022	3920,08	4,07	29,50	48,81	53,17	0,00
29/11/2022	3868,03	3,36	30,81	58,82	57,93	0,35
<b>PROMEDIO</b>	<b>3874,19</b>	<b>3,96</b>	<b>29,02</b>	<b>49,81</b>	<b>54,38</b>	<b>Σ = 1,57</b>

*Elaborado por: El investigador*

Durante las dos primeras semanas se puede observar que la cantidad de radiación que reciben las plantas es la adecuada, puesto que tiene un valor promedio de 3801.735 Wh/m<sup>2</sup>. Al medir el pH se observa que está muy por debajo del valor recomendado ya que el promedio de las dos semanas de 3,96 cuando el valor mínimo debe ser 5,5. En cuanto la temperatura el promedio indica que está alrededor de los 29 °C, pero se puede observar en la tabla que hubo días en los cuales se alcanzaron los 33,01 grados centígrados, un promedio por encima del valor recomendado para el cultivo de pimiento. la humedad del aire o de ambiente tiene un valor promedio de 49% valor que está dentro del rango normal dentro de un invernadero ya que en las noches tiende a subir, pero a lo largo del día al abrir las cortinas y activar ventiladores dicha humedad tiende a bajar considerablemente. El promedio de la humedad del suelo entre las dos semanas es de 52,9% valor que está dentro del rango recomendado, pero se puede observar que hubo 3 días consecutivos donde la humedad fue inferior al 50% qué es el valor mínimo recomendado. El consumo eléctrico de las dos primeras semanas fue de 2.92 kW/h, valor que se debe aquel usuario activo el riego por goteo 3 veces durante las dos semanas por períodos cortos de tiempo.

### 3.1.5 Eficiencia del sistema

A partir de la tercera semana (tabla 26) se activa el modo automático en el sistema, por ende, el sistema empieza a trabajar con ventiladores, apertura y cierre de cortinas a través de un motor, iluminación artificial, sistema de riego y sistema de aspersión en base a los valores sensados.

**Tabla 26.** Datos sensados con el sistema en modo automático. Semana 3.

DÍA	Radiación (Wh/m <sup>2</sup> )	pH	Temperatura promedio (°C)	Humedad aire promedio (%)	Humedad suelo promedio (%)	Consumo (kW/h)	Producción (Sacos de 12kg)
30/11/2022	3815,73	3,49	26,11	45,27	57,78	1,34	12
1/12/2022	3914,93	6,17	27,25	64,69	66,56	2,74	
2/12/2022	3935,02	6,16	26,72	47,95	62,11	1,34	

3/12/2022	3944,94	5,92	25,64	46,93	60,69	1,24	
4/12/2022	4089,77	5,84	29,53	41,23	57,46	1,71	
5/12/2022	4015,62	6,07	29,50	38,81	63,17	3,12	
6/12/2022	3962,30	5,86	28,81	45,82	61,93	1,57	
<b>PROMEDIO</b>	3954,04	5,64	27,65	47,24	60,81	$\Sigma = 13,06$	

*Elaborado por: El investigador*

Como se puede observar en la tabla 26 al analizar los datos ya presenta mejorías notorias las cuales favorecen al cultivo. El promedio de la radiación por día es de 3954,04  $Wh/m^2$  valor adecuado para que las plantas crezcan y produzcan adecuadamente. El valor promedio del pH es de 5,64, ese valor se debe a aquel primer día de la semana marca 3,49 ya que el sistema de riegos se activa al segundo día y ahí es cuando el usuario a través de productos químicos controla el pH por medio del agua que ingresa al invernadero a través del riego por goteo. El valor de la temperatura promedio marca 27,65 °C, valor que está dentro del rango recomendado. La humedad de ambiente tiene un valor promedio de 47,24%, promedio de la humedad del suelo es de 60,81% valor dentro del rango ideal para el cultivo de pimiento. en cuanto al consumo eléctrico se puede notar un incremento considerable durante la primera semana del funcionamiento del sistema ya que la primera semana donde solo se realizó el sensado de variables el consumo fue de 1,35 kW/h, y en la primera semana donde el sistema ya está funcionando completamente y de manera automática el consumo fue de 13,06 kW/h, esto se debe a que el sistema de iluminación ya se enciende en las mañanas de forma automática para despertar a la planta antes de que salga el sol y que la fotosíntesis de esta se active antes de lo que lo hacía habitualmente con ello el tiempo de trabajo de la planta aumenta con el propósito de incrementar la producción, además en caso de ser necesario el sistema de iluminación se activará cuando al finalizar el día las plantas no hayan recibido al menos . Durante la primera semana de funcionamiento del sistema informa automática no se puede ver aún cambios en la producción puesto que el cultivo se realiza cada dos semanas.

**Tabla 27.** Datos sensados con el sistema funcionando en modo automático. Desde la semana 4 a la semana 8.

SEMANA	Radiación (Wh/m <sup>2</sup> )	pH	Temperatura promedio (°C)	Humedad aire promedio (%)	Humedad suelo promedio (%)	Consumo (kW/h)	Producción (Sacos de 12kg)
Del 7 al 13 de diciembre	4063,73	5,59	27,26	48,20	60,05	12,32	12
Del 14 al 20 de diciembre	4014,13	5,91	27,75	47,78	58,96	12,97	14
Del 21 al 27 de diciembre	3885,42	5,72	25,41	49,23	57,49	11,64	
Del 28 de diciembre al 3 de enero	3771,34	5,53	23,58	50,77	60,82	9,85	17
Del 4 al 10 de enero	3940,97	6,02	26,29	47,34	58,61	13,37	
<b>PROMEDIO</b>	3935,12	5,83	26,06	48,89	59,19	12,03	$\Sigma = 42$

*Elaborado por: El investigador*

Durante las 5 semanas siguientes (tabla 27) se puede observar que los valores medidos son cercanos a los medidos durante la primera semana de funcionamiento del sistema en modo automático sin prestar grandes cambios, pero si mostrando valores dentro del rango adecuado para el cultivo de pimiento. en cuanto al cultivo se puede observar que durante las dos primeras semanas la cosecha fue incluso menor que las primeras dos semanas cuando solamente se estuvo monitoreando el comportamiento del invernadero. A partir de la semana 5 y 6 se observa un leve incremento en la producción con un total de 14 sacos de pimiento cosechado, pero a partir de la semana 7 y 8 se observa ya un cambio notorio en la cosecha con 17 sacos cosechados lo que indica que el sistema está generando un ambiente amigable para el cultivo de pimiento bajo invernadero.

### 3.1.5.1 Prueba del funcionamiento del HMI

Con el objetivo de comprobar que el HMI del sistema funcione de manera adecuada se realiza el monitoreo del invernadero el día 11 de enero del 2023 alrededor de las 11:30 h, como se puede observar en los gráficos de las figuras 159, 160, 161, 162 y 163, los valores promedios medidos están acordes a los valores sensados en cada sector del invernadero.

El día se presenta lluvioso y con temperaturas bajas debido a eso se pueden observar los siguientes valores medidos en el interior del invernadero.



(a)



(b)

**Figura 159.** Monitoreo de la humedad del aire

*Fuente: El investigador*



(a)



(b)

**Figura 160.** Monitoreo de la humedad del suelo

*Fuente: El investigador*



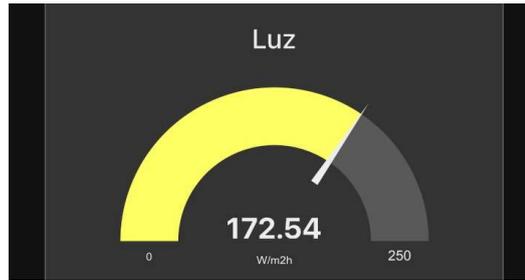
(a)



(b)

**Figura 161.** Monitoreo de la temperatura

*Fuente: El investigador*



**Figura 162.** Monitoreo de la radiación

*Fuente: El investigador*



**Figura 163.** Monitoreo del pH

*Fuente: El investigador*



**Figura 164.** Monitoreo de estado

*Fuente: El investigador*

### 3.1.5.2 Monitoreo del invernadero de un día completo

Con la intención de analizar el funcionamiento del sistema durante un día completo, se muestra a continuación la tabla obtenida de la base de datos donde se puede observar que se almacenan datos cada 5 minutos. Como se mencionó anteriormente el día se presenta lluvioso y con temperaturas bajas debido a eso se puede observar la tabla 28.

*Tabla 28. Valores tomados en la mañana del 11 de enero del 2023*

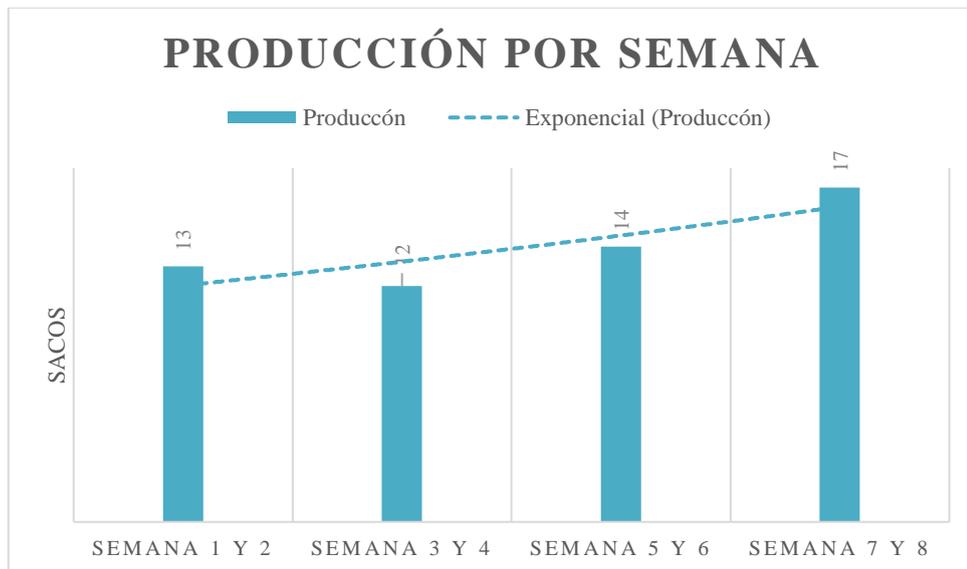
id	hora	radiación _Wh/m <sup>2</sup>	pH	C	T_pro	H_aire_pro	H_suelo_pro	Cons_kWh/h
1	06:00:36	56,60	5,15	0	10,84	88,25	56,20	1,0233
2	06:05:36	53,60	5,16	0	10,69	87,09	55,31	1,0233
3	06:10:36	52,19	5,15	0	9,63	78,31	49,67	1,0233
4	06:15:36	51,50	5,15	0	10,90	88,32	56,13	1,0232
5	06:20:36	51,26	5,16	0	10,91	88,37	56,16	1,0234
6	06:25:36	51,15	5,16	0	10,96	88,49	56,09	1,0232
7	06:30:36	51,07	5,16	0	11,07	88,86	56,05	0,0034
8	06:35:36	52,78	5,16	0	11,17	88,98	56,14	0,0033
9	06:40:36	70,31	5,17	0	11,25	88,99	56,08	0,0034
10	06:45:36	87,42	5,18	0	11,34	89,00	56,11	0,0036
11	06:50:36	102,25	5,18	0	11,42	89,00	56,11	0,0034
12	06:55:36	108,36	5,19	0	11,54	89,00	56,06	0,0034
13	07:00:36	111,81	5,19	0	11,67	89,01	56,09	0,0033
14	07:05:36	120,80	5,19	0	11,82	89,12	56,01	0,0032
15	07:10:36	126,81	5,20	0	12,06	89,92	56,03	0,0033
16	07:15:36	117,51	5,20	0	12,10	89,98	55,80	0,0032
17	07:20:36	121,38	5,20	0	12,11	89,98	56,04	0,0033
18	07:25:36	133,11	5,20	0	12,28	90,00	56,06	0,0033
19	07:30:36	138,62	5,20	0	12,40	90,00	56,09	0,0033
20	07:35:36	141,13	5,20	0	12,60	90,00	56,00	0,0035
21	07:40:36	145,13	5,21	0	12,93	90,45	56,00	0,0032
22	07:45:36	147,37	5,20	0	13,20	91,00	56,10	0,0034
23	07:50:37	143,31	5,20	0	13,31	91,00	56,00	0,0036
24	07:55:37	139,26	5,20	0	13,26	91,00	56,00	0,0034
25	08:00:37	144,64	5,20	0	13,35	91,00	56,05	0,0033
26	08:05:37	147,26	5,21	0	13,53	91,00	56,03	0,0032
27	08:10:37	142,00	5,20	0	13,58	91,00	56,02	0,0034
28	08:15:37	149,82	5,20	0	13,61	91,00	55,94	0,0032
29	08:20:37	153,88	5,20	0	13,56	91,00	55,99	0,0033
30	08:25:37	156,08	5,20	0	14,03	91,00	55,94	0,0034
31	08:30:37	156,36	5,20	0	14,39	91,00	56,06	0,0034
32	08:35:37	155,10	5,20	0	14,60	91,00	56,02	0,0035
33	08:40:37	252,87	5,20	0	14,41	91,00	56,08	0,0033
34	08:45:37	253,07	5,20	0	14,16	91,00	56,09	0,0035
35	08:50:37	257,07	5,20	0	14,38	91,00	55,94	0,0033
36	08:55:37	20,44	5,20	0	14,86	91,00	56,01	0,0036
37	09:00:37	260,22	5,20	0	15,22	91,00	56,09	0,0034
38	09:05:37	359,50	5,21	0	15,32	91,00	56,07	0,0033
39	09:10:37	362,06	5,20	0	15,75	91,19	56,04	0,0035
40	09:15:37	261,31	5,21	0	16,15	91,55	55,98	0,0034
41	09:20:37	266,16	5,21	0	16,71	90,99	56,17	0,0035
42	09:25:37	369,25	5,22	0	17,33	89,98	56,14	0,0032
43	09:30:37	370,62	5,21	0	18,14	89,36	56,18	0,0035
44	09:35:37	371,68	5,21	0	18,90	88,05	56,28	0,0035
45	09:40:37	370,29	5,22	0	19,35	86,86	56,27	0,0034
46	09:45:37	368,66	5,21	0	19,44	85,43	56,37	0,0035
47	09:50:37	363,41	5,21	0	18,82	87,22	56,34	0,0034
48	09:55:37	363,68	5,21	0	18,44	88,12	56,21	0,0034
49	10:00:37	364,48	5,21	0	17,96	88,17	56,35	0,0037
50	10:05:37	466,05	5,21	0	18,05	88,51	56,37	0,0034
51	10:10:37	468,29	5,21	0	18,51	87,84	56,41	0,0036

52	10:15:37	466,06	5,21	0	18,55	87,13	56,32	0,0035
53	10:20:37	462,88	5,20	0	17,73	87,41	56,42	0,0036
54	10:25:37	464,98	5,20	0	17,57	88,42	56,51	0,0035
55	10:30:37	463,72	5,20	0	18,00	88,63	56,28	0,0034
56	10:35:37	562,25	5,20	0	17,71	88,50	56,50	0,0036
57	10:40:38	563,39	5,20	0	17,66	88,74	56,50	0,0033
58	10:45:38	564,10	5,20	0	17,85	88,58	56,55	0,0034
59	10:50:38	567,14	5,20	0	18,04	87,96	56,53	0,0035
60	10:55:38	569,60	5,20	0	18,77	85,91	56,45	0,0035
61	11:00:38	569,26	5,20	0	19,16	84,65	56,40	0,0035
62	11:05:38	573,13	5,20	0	20,05	81,55	56,49	0,0033
63	11:10:38	573,73	5,20	0	21,15	78,22	56,49	0,0035
64	11:15:38	477,25	5,20	0	22,59	73,99	56,46	0,0034
65	11:20:38	475,57	5,21	0	23,59	69,04	56,54	0,0034
66	11:25:38	474,39	5,29	0	24,31	66,46	56,76	0,0034
67	11:30:38	472,48	5,54	0	24,28	65,98	64,74	0,0034
68	11:35:38	472,61	5,65	0	24,38	66,43	75,57	0,0034
69	11:40:38	575,60	5,63	0	24,98	62,43	75,21	0,0036
70	11:45:38	576,87	5,65	0	26,72	57,44	75,38	0,0034
71	11:50:38	577,93	5,66	1	28,34	53,89	75,38	0,0033
72	11:55:38	577,59	5,66	1	30,10	49,15	75,37	0,0034
73	12:00:38	576,85	5,66	1	29,73	46,79	75,32	0,0036
74	12:05:38	578,97	5,64	1	29,24	47,05	75,16	0,0035
75	12:10:38	577,76	5,63	1	30,07	44,32	74,96	0,0034

*Elaborado por: El investigador*

### 3.1.5.3 Producción y consumo energético del sistema

La producción del invernadero si presenta un incremento a lo largo de las 8 semanas de prueba del sistema como se puede observar en la figura 165.

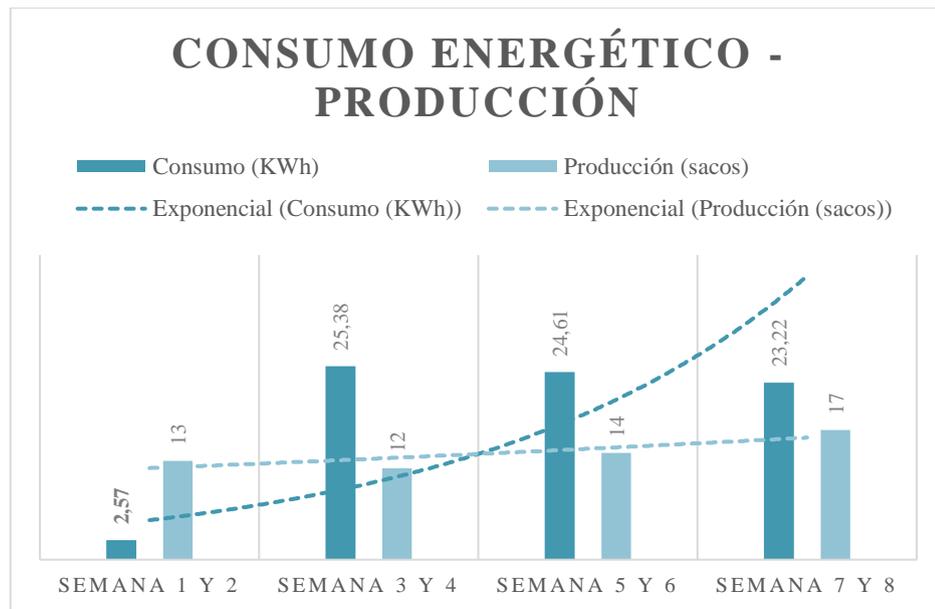


*Figura 165. Producción por semana*

*Fuente: El investigador*

En la gráfica 165 se puede apreciar que a partir de la quinta y sexta semana la

producción aumento de forma moderada, con una producción de 14 sacó, pero a partir de la séptima y octava semana ya el incremento en la producción es más notorio con una producción de 17 sacos. Al hacer la relación entre la cosecha de las primeras dos semanas y la cosecha de las dos últimas semanas existe una diferencia de cuatro sacos, lo que representa un incremento en la producción del 30, 77%.



**Figura 166.** Comparación entre consumo energético y producción

**Fuente:** El investigador

En la figura 166 se puede apreciar que el incremento del consumo energético es bastante notorio sin embargo el incremento en la producción se hace notorio a partir de la semana 7 y 8, algo que realmente está dentro de lo normal puesto que la planta empieza a dar resultados a partir del mes o mes y medio después de estar en un ambiente amigable para su producción, por ende, los resultados obtenidos están dentro de lo normal.

En la tabla 29 se realiza una comparación entre la producción del invernadero y el consumo, pero desde un punto de vista económico, con el fin de analizar si el sistema de control y monitoreo brinda ganancias o pérdidas al usuario después de su implementación.

**Tabla 29.** *Análisis de eficiencia del sistema*

	<b>Producción (sacos)</b>	<b>Costo en mercado (\$)</b>	<b>Consumo (KW/h)</b>	<b>Consumo (\$)</b>
<b>Sin el sistema</b>	26	312,00	5,84	0,54
<b>Con el sistema</b>	34	408,00	46,44	4,28
<b>Diferencia</b>	8	96,00	40,6	3,74

*Elaborado por: El investigador*

Para la realización de la tabla anterior se tomaron los datos de las primeras dos semanas y se multiplicaron por dos para realizar un estudio mensual, y se tomaron las dos últimas semanas donde también se multiplicó por dos para dicho análisis. Con esos datos se procede a hacer una comparación entre un mes sin el sistema de control y monitoreo y otro mes con el sistema de control y monitoreo, como se puede observar hay una diferencia de 8 sacos al mes en la producción lo que representa \$96,00. Con relación al consumo se observa una diferencia de 40,6 KW/h lo que representa económicamente una cifra de \$3,74. Por tanto se observa que la ganancia es bastante superior al gasto que representa mensualmente el tener activó el sistema de control y monitoreo. Si el sistema tiene un costo de \$1.974,90 (teniendo en cuenta solo los materiales), y la diferencia entre lo que gana y lo que paga mensualmente del consumo energético es de \$92,26, el usuario tardará 1 año y 10 meses en recuperar la inversión, pero es necesario tener en cuenta que al contar con un invernadero de estructura metálica su tiempo de vida útil es superior a los 25 años [55], el plástico del invernadero tiene un tiempo de vida útil de alrededor de los 5 años, además el tiempo de vida útil de una bomba de agua es aproximadamente 20 años según la norma API 610. Por tanto, el usuario experimentará ganancias durante un largo periodo de tiempo después del año y 10 meses mencionado anteriormente.

Cabe recalcar un punto importante del sistema de control y monitoreo, la autonomía, que le permite al usuario importante deslindarse de gran manera del invernadero día a día, puesto que antes de la instalación del sistema el usuario debía estar constantemente pendiente del invernadero debido a que la temperatura durante el presenta variaciones, así como el estar constantemente pendiente del riego ya que lo realizaba de manera intuitiva contando los días que pasaban entre riego y riego u observado la humedad del suelo.

### 3.1.6 Presupuesto

En la tabla 30 se detalla el costo de cada uno de los materiales, dispositivos y equipos que se utilizaron en la implementación del sistema de control y monitoreo para el cultivo de hortalizas bajo invernadero, donde se especifica que el costo total es de \$2.019,90 incluido IVA.

**Tabla 30. Presupuesto del sistema de control y monitoreo**

<b>PLACA NODO LORA (2)</b>				
<b>Ítem</b>	<b>Componente</b>	<b>Cantidad</b>	<b>Precio Unitario (\$)</b>	<b>Precio total (\$)</b>
1	Lora32 V2 Lilygo 868/915MHz	2	45,00	90,00
2	MOC3020M	8	0,45	3,60
3	TRIAC BT138	8	0,80	6,40
4	Diodo 1N4728	8	0,15	1,20
5	Potenciómetro de precisión de 100K	8	0,35	2,80
6	Amplificador operacional LM358N	4	0,50	2,00
7	Borneras	12	0,30	3,60
8	Espadines	6	0,60	3,60
9	Baquelita electrónica	2	1,50	3,00
10	Estaño (1 metro)	1	0,75	0,75
11	Acido para baquelita	1	0,70	0,70
12	Carcasa	2	9,00	18,00
<b>PLACA NODO SENSOR (4)</b>				
13	Módulo ESP32	4	12,50	50,00
14	MOC3020M	8	0,45	3,60
15	TRIAC BT138	8	0,80	6,40
16	Diodo 1N4728	8	0,15	1,20
17	Dip Switch 4 vías	4		
18	Potenciómetro de precisión de 100K	8	0,35	2,80
19	Amplificador operacional LM358N	4	0,50	2,00
20	Borneras	20	0,30	6,00
21	Espadines	16	0,60	9,60
22	Baquelita electrónica	4	1,50	6,00
23	Estaño (1 metro)	2	0,75	1,50
24	Ácido para baquelita	2	0,70	1,40
25	Carcasa	4	9,00	36,00
<b>GATEWAY</b>				
26	Lora32 V2 Lilygo 868/915MHz	1	45,00	45,00
27	Fuente 5V	1	6,00	6,00
28	Cable USB	1	3,00	3,00
<b>SERVIDOR</b>				
29	Raspberry Pi 4B con 4GB de memoria RAM + kit completo	1	195,00	195,00
30	Adaptar HDMI – HDMI mini	1	5,00	5,00
<b>SENSORES</b>				
31	DHT11	4	4,00	16,00
32	Sensor de humedad de suelo FC-28	4	4,00	16,00
33	Sensor de pH	1	50,00	50,00
34	Panel solar	2	2,00	4,00
35	Sensor de corriente	2	10,50	21,00
<b>TABLERO ELÉCTRICO NODO LORA INVERNADERO</b>				
36	Tablero eléctrico	1	45,00	45,00
37	Riel para relés	1	2,00	2,00

38	Cable de cobre, calibre 14 (por metros)	3	0,60	1,80
39	Transformador de 220Vac a 12Vac	1	4,00	4,00
40	Relés de 10A	2	7,50	15,00
41	Relés de 5A	3	6,50	19,50
42	Fuente 12Vdc, 3A	1	7,00	7,00
43	Correas (bridas)	1	3,50	3,50
<b>TABLERO ELÉCTRICO NODO LORA BOMBA DE AGUA</b>				
44	Tablero eléctrico	1	45,00	45,00
45	Riel para relés	1	2,00	2,00
46	Cable de cobre, calibre 14 (por metros)	3	0,60	1,80
47	Transformador de 220Vac a 12Vac	1	4,00	4,00
48	Relés de 10A	2	7,50	15,00
49	Relés de 5A	3	6,50	19,50
50	Fuente 12Vdc, 3A	1	7,00	7,00
51	Correas (bridas)	1	3,50	3,50
52	Breaker 2 fases (220V)	1	12,50	12,50
53	Breaker 1 fase (110V)	1	8,40	8,40
<b>SISTEMA DE ELEVACIÓN DE CORTINAS</b>				
54	Motor eléctrico de bajas revoluciones	1	187,00	187,00
55	Poles de 2 in	1	2,20	2,20
56	Polea de 2,5 in	1	2,50	2,50
57	Banda	1	6,50	6,50
58	Tubo G de 2 in	6	8,00	48,00
59	Chumaceras 3/4 in	10	3,50	35,00
60	Rulimanes 1,5 in	14	4,00	56,00
61	Varilla de 8mm	5	6,00	30,00
62	Tubo 1 in	5	7,50	37,50
63	Cemento de contacto (1L)	1	4,50	4,50
<b>ILUMINACIÓN</b>				
64	Focos de alta potencia 60W	17	6,50	110,50
65	Boquillas	17	0,60	10,20
66	Cable gemelo calibre 14 (rollo de 100m)	1,5	65,00	97,50
<b>VENTILACIÓN</b>				
67	Ventiladores 150w	2	155,00	310,00
68	Cable gemelo calibre 14 (rollo 100m)	0,5	65,00	32,50
<b>RIEGO</b>				
69	Electroválvulas	3	32,50	97,50
70	Nebulizadores	5	2,50	12,50
<b>OTROS</b>				
71	Cable sólido calibre 12 (por metro)	50	0,80	40
72	Cable calibre 14	25	0,45	11,25
73	Cable de teléfono (rollo de 100m)	1	35,00	35,00
74	Electrodos (por libra)	2	3,75	7,50
75	Discos de corte para amoladora	3	1,00	3,00
76	Cinta adhesiva (tape)	3	1,20	3,60
<b>TOTAL</b>				\$2.019,90

*Elaborado por: El investigador*

Al costo total del prototipo calculado en la tabla anterior es necesario agregarle \$25,00 de los costos de transporte, \$90,00 por 3 días de trabajo de un carpintero metálico,

\$30,00 de materiales de oficina y \$800,00 del sueldo base de un ingeniero en la actualidad, ya que el tiempo invertido para desarrollar el proyecto ha sido alrededor de un mes. Por tanto, el costo total de la implementación del sistema es de \$2964,90.

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- En el presente proyecto de investigación, se implementa un sistema de control y monitoreo para el cultivo de hortalizas bajo invernadero basado en tecnología Lora y MQTT, utilizando como caso de cultivo una cosecha de pimiento, donde se obtiene como resultado un incremento de 13 a 17 sacos de producción cada dos semanas, es decir experimenta un incremento del 30,77% equivalente a \$96,00 en la producción del invernadero, así como un control manual y automatizado del invernadero en base a la temperatura, humedad, pH e iluminación, dotando al invernadero de una autonomía inteligente, así como de un monitoreo en tiempo real de las condiciones climáticas de su interior. El sistema presenta un consumo de 46,44kW/h equivalente a \$4,28.
- Analizar la situación actual del proceso de cultivo de hortalizas bajo invernadero, permite determinar las necesidades que existe en este tipo de cultivos, ya que la tecnificación en invernaderos a baja escala es muy carente, debido a que cuentan en su mayoría solamente con riego por goteo, sin un sistema de control de temperaturas, de humedad, pH ni iluminación, provocando condiciones poco favorables para el cultivo. Al tomar datos antes de implementar el sistema se registraron temperaturas promedio de 33,01 °C, un valor de pH de 3,65 y humedad del suelo de 40,98%, valores notablemente fuera de los rangos adecuados para el cultivo. Después de implementar el sistema se logra establecer promedios de 25,75 °C, niveles de pH de 6,02 y una humedad del 60,82%. Por ende, la implementación del sistema se convierte en una buena alternativa para mejorar las condiciones climáticas para un cultivo.
- El desarrollo del sistema de control y monitoreo se realizó en base a la tecnología LoRaWAN y MQTT. LoRaWAN trabaja con una arquitectura de red basada en la topología estrella dónde existen dos o más Nodos LoRa, un Gateway y un servidor, además de los nodos sensores. la placa seleccionada para desarrollar la función de gateway y nodos, es Lora32 V2 Lilygo 868/915MHz, la cual admite trabajar dentro de la frecuencia ISM del Ecuador, permitiendo comunicaciones de hasta 8 km de distancia con línea de vista directa. En el enlace Nodo LoRa del invernadero y el Gateway se registró unas pérdidas del espacio libre de 74.102 dB, mientras que entre

el Nodo LoRa de la Bomba de agua y el Gateway se registró unas pérdidas del espacio libre de 72.786 dB, donde se consideró una frecuencia de operación de 915 MHz.

- A través de la aplicación Telegram el sistema envía notificaciones de alerta, además el sistema puede ser controlado de manera rápida y sencilla, monitoreando cada uno de los parámetros propuestos con el uso de comandos previamente insertados en la aplicación y registrados en la programación de los nodos. Por otra parte, a través del HMI creado a través de Node-RED también es posible visualizar de forma gráfica los datos almacenados en la base de datos de un día o un mes. De forma experimental se realizan numerosas pruebas para medir el tiempo de respuesta del sistema con una velocidad de internet de 34Mbps, con un promedio de respuesta de 0,78s desde la app de Telegram, tanto desde un dispositivo móvil como desde una computadora. Así mismo se realizaron pruebas desde el HMI donde se obtuvo un tiempo de respuesta promedio de 0,81s.

#### **4.2 Recomendaciones**

- Se recomienda realizar modificaciones en la estructura del invernadero incrementando las aperturas del techo para la ventilación, se recomienda pasar de 20 cm a 40 cm, con ello el control de temperatura sería más rápido y los ventiladores no pasarían largos periodos de tiempo activados, generando una reducción en el consumo eléctrico.
- En el caso de incrementar el número de nodos Lora, se recomienda ubicarlos con línea de vista directa entre el nodo lora y el Gateway, para evitar posibles errores de comunicación y pérdida de datos o al menos procurar que el 60% de la primera zona de Fresnel esté libre de obstáculos. Además, respetar las distancias máximas de comunicación entre dispositivos con tecnología Lora, hasta 10 km en zonas urbanas y hasta 20 km en campo abierto y zonas rurales.
- Para reducir el consumo de corriente se recomienda que la mayoría de los dispositivos, equipos o maquinaria que se deseen implementar en el invernadero funcionen a 220 voltios AC, así el usuario estará en la capacidad de incrementar el número de equipos funcionando al mismo tiempo. También, se recomienda implementar baterías con paneles solares para el funcionamiento de toda la parte electrónica del proyecto para garantizar la disponibilidad del sistema, sin importar los cortes de energía eléctrica.
- Con el fin de tener una mayor precisión en los promedios realizados de temperatura y

humedad se recomienda incrementar el número de sensores. En este proyecto se trabajó con 4 sectores, pero se podrían aumentar a 8 sectores, con ello el número de sensores también se dobla, consiguiendo reducir el área monitoreada y así los valores y promedios calculados serán lo más cercanos a la realidad.

- En el presente proyecto se realiza el monitoreo del pH, por lo que se recomienda su automatización, con el propósito de evitar que las plantas experimenten rangos de pH por debajo de 5,2 o por encima de 6,5 dependiendo del cultivo. Además, el mismo sistema se podría utilizar para automatizar la aplicación de nutrientes, fertilizantes o pesticidas que permitan el buen desarrollo de los cultivos. Para la automatización se recomienda la utilización de una Bomba Peristáltica de 60ml/min a 12 Vdc para dosificar la solución reguladora de pH. En el ANEXO 16 se detalla el diagrama de conexión para la automatización.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] E. Hernández, «Desarrollo de un Sistema de monitorización y control de un invernadero aplicando Tecnología IoT,» Universidad de Cantabria, Cantabria, España, 2019.
- [2] H. Mora y J. Rosas, «Diseño, desarrollo e implementación de una red de sensores inalámbricos (WSN) para el control, monitoreo y toma de decisiones aplicado en la agricultura de precisión basado en Internet de las Cosas (IoT),» Universidad Ricardo Palma, Lima, Perú, 2019.
- [3] C. Villarroel, V. Goykovic, P. Collao, M. Barraza, J. Fernández, Á. Villarroel, R. Valdivia y C. Castro, «Evaluación de desempeño de un invernadero ubicado en el desierto de Atacama, Chile, a través de IoT,» *Interciencia*, vol. 44, n° 7, pp. 386-393, 2019.
- [4] Y. Aguilar, S. Sosa, J. Galindo, A. Morales, R. Gallardo y I. Villaseñor, «Diseño y desarrollo de un sistema de monitoreo remoto implementando Internet de las cosas,» *Research in Computing Science*, vol. 149, n° 11, p. 235–247, 2020.
- [5] V. Soria, «Desarrollo de una red IoT con tecnología Lora para gestión de invernaderos,» Universidad Politécnica Salesiana, Quito, Ecuador, 2021.
- [6] Berger, «Berger,» 2020. [En línea]. Available: <https://www.berger.ca/es/recursos-para-los-productores/tips-y-consejos-practicos/cultivar-invernadero-ventajas-desventajas/>. [Último acceso: 25 julio 2021].
- [7] R. Barraqueta, «Implementación de invernadero con sistema de control y monitoreo inalámbricos,» ESCUELA UNIVERSITARIA POLITÉCNICA DE LA ALMUNIA DE DOÑA GODINA, ZARAGOZA, 2016.
- [8] Spagnol, 23 marzo 2021. [En línea]. Available: <https://www.spagnol.com/es-es/soluciones/invernadero-automatizado>. [Último acceso: 31 mayo 2022].
- [9] Redagícola, «Redagícola,» 30 noviembre 2019. [En línea]. Available: <https://www.redagricola.com/cl/el-avance-de-la-automatizacion-en-la-agricultura/>.
- [10] Sofos organización de conocimiento , «Sofos,» 15 agosto 2021. [En línea]. Available: <http://www.sofoscorp.com/tecnologia-en-la-agricultura-y-automatizacion-en-procesos-de-cosechas/>.

- [11] E. García y F. Flego, «Agricultura de Precisión,» *Tecnología Agropecuaria*, pp. 99-116, 2015.
- [12] Universitat de lleida, «Grupo de Recerca en AgròTICA,» 07 04 2020. [En línea]. Available: <http://www.grap.udl.cat/es/presentacion/ap.html>.
- [13] Agricultura de Precisión (AP), «Proxima Systems,» 5 marzo 2019. [En línea]. Available: <https://www.proximasystems.net/agricultura/agricultura-de-precision-ap/>.
- [14] S. Santos y J. Kienzle, *Agricultura 4.0: Robótica agrícola y equipos automatizados para la producción agrícola sostenible.*, Roma: FAO, 2021.
- [15] Pat Mooney, Grupo ETC, *La insostenible Agricultura 4.0 - Digitalización y poder corporativo en la cadena alimentaria*, Ciudad de México: Rosa-Luxemburg-Stiftung Oficina para México, 2020.
- [16] S. A. -. M. P. Digitales, «CumpuSoluciones,» 8 marzo 2022. [En línea]. Available: <https://www.compusuluciones.com/blog/greentegrate-software-agricultura-4-0/>.
- [17] Proxima systems, «Proxima systems,» 4 febrero 2019. [En línea]. Available: <https://www.proximasystems.net/agricultura/que-es-la-agricultura-4-0/>.
- [18] N. Iglesias, *Producción de hortalizas*, Río Negro, Argentina: INTA, 2006.
- [19] J. P. Palacios, «DISEÑO DE UN SISTEMA DE MONITOREO, REGISTRO Y CONTROL DE TEMPERATURA Y HUMEDAD PARA UN CULTIVO DE INVERNADERO.,» UNIVERSIDAD TECNOLÓGICA DE PEREIRA, Pereira (Colombia), 2016.
- [20] C. G. R. C. C. L. Basil Al-Hadithi, «Desarrollo de un Sistema de Iluminación Artificial Inteligente para Cultivos Protegido,» cada. Universidad Politécnica de Madrid, Madrid (España), 2016.
- [21] F. D. d. P. y. P. V. 90, *El Cultivo Protegido en Clima Mediterráneo*, Roma: ORGANIZACIÓN DE LAS NACIONES UNIDAS PARA LA AGRICULTURA Y LA ALIMENTACIÓN, 2002.
- [22] P. D. D. H. P. d. D. d. l. C. d. V. y. C. Agrícola, *Cultivo de tomate*, MCA/Nicaragua: Chemonics International, Inc, 2008.
- [23] B. Alberto y S. Gardenia, «PRODUCCIÓN DE TRES HÍBRIDOS DE PIMIENTO (*Capsicum annuum*) A PARTIR DE SEMILLAS SOMETIDAS A IMBIBICIÓN E IMBIBICIÓN MÁS CAMPO MAGNÉTICO EN EL CAMPO EXPERIMENTAL RÍO

VERDE, CANTÓN SANTA ELENA,» Universidad Estatal Península de Santa Elena, La libertad - Ecuador , 2007.

- [24] Flores y plantas, «Floresyplantas.net,» 10 marzo 2016. [En línea]. Available: <https://www.floresyplantas.net/el-cultivo-del-pimiento/>.
- [25] L. Angel, «Huertos y más,» 1 junio 2020. [En línea]. Available: <https://huertosymas.com/etapas-de-crecimiento-del-pimiento/>.
- [26] J. Chiriboga, «ADAPTACIÓN Y RENDIMIENTO DE OCHO VARIEDADES DE PIMIENTO (*Capsicum annuum* L.) EN INVERNADERO, CANTÓN RIOBAMBA, PROVINCIA CHIMBORAZO,» ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Riobamba- Ecuador, 2019.
- [27] M. Huilcarema, «EFECTO FOLIAR DE DOS DOSIS DE BORO EN EL RENDIMIENTO DE OCHO VARIEDADES DE PIMIENTO (*Capsicum annuum* L.) EN INVERNADERO,» ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, RIOBAMBA- ECUADOR, 2020.
- [28] G. Chuncho, C. Chuncho y Z. Aguirre, «Anatomía y morfología vegetal,» Universidad Nacional de Loja, Loja, Ecuador, 2019.
- [29] D. News, «DPL News,» 30 10 2019. [En línea]. Available: <https://dplnews.com/el-internet-de-las-cosas-iot-tendencias-claves-en-el-sector-energia/>.
- [30] S. S. Jordi Salazar, «Internet de las cosas,» České vysoké učení technické v Praze Fakulta elektrotechnická, Praga, República Checa, 2017.
- [31] O. B. S. B. Pradyumna Gokhale, «Introduction to IOT,» IARJSET, India, 2018.
- [32] I. I. Connexions, «Generalitat Valenciana,» 05 01 2022. [En línea]. Available: <https://www.istecdigital.es/internet-of-things-iot/>.
- [33] M. W. S. M. A. K. T. K. M.U. Farooq, «A Review on Internet of Things (IoT),» International Journal of Computer Applications, 2015.
- [34] R. Pérez, «Evaluación de LoRa/LoRaWAN para escenarios de Smart City,» Universitat Politècnica de Catalunya, Barcelona, España, 2017.
- [35] H. Andrade, «Sistema de monitoreo inalámbrico de una piscícola con tecnología LoRa para la producción de truchas en Valle Hermoso - Pelileo,» Universidad Técnica de Ambato, Ambto, 2022.
- [36] D. Caisa, «Prototipo IOT de telecontrol del riego a goteo para la agricultura inteligente

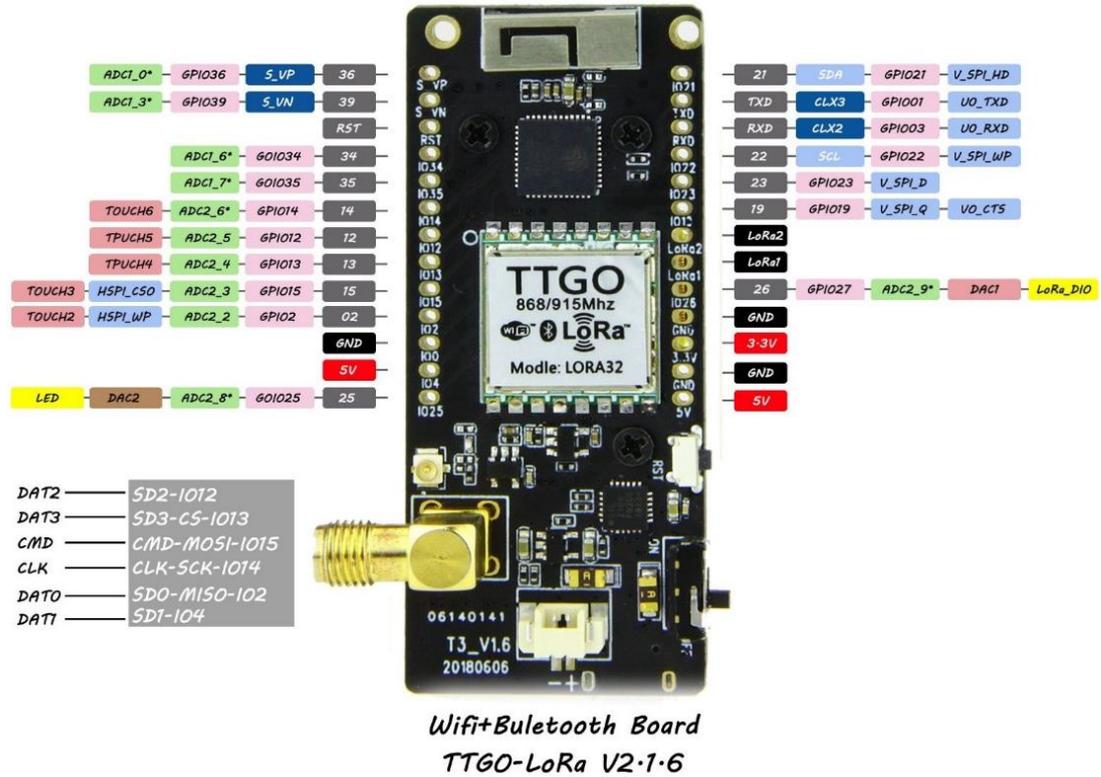
- con tecnología LORA.,» Universidad Técnica de Ambato, Ambato, 2022.
- [37] A. d. R. y. C. d. I. Telecomunicaciones, «PLAN NACIONAL DE FRECUENCIAS ECUADOR 2021,» Dirección Técnica de Regulación del Espectro Radioeléctrico, Ecuador, 2021.
- [38] M. Leslie, «Sistema inteligente de monitoreo y vigilancia para plantaciones agrícolas de pitahaya en el cantón Palora basado en tecnologías IoT y LoRa,» UTA, Ambato, 2022.
- [39] J. C. Buestán, «Desarrollo de una red IoT con tecnología LoRa para detección de automóviles,» Universidad Politécnica Salesiana, Quito, 2019.
- [40] E. Pérez, «Mediem,» 27 agosto 2018. [En línea]. Available: <https://medium.com/pruebas-de-laboratorio-de-la-modulaci%C3%B3n-lora/lorawand00f48384160>.
- [41] L. D. González, «Test y despliegue de tecnología de comunicaciones LoRa para aplicaciones de Internet of Things,» Universidad Politécnica de Madrid, Madrid, 2019.
- [42] M. Calleja y J. Guasch, «Monitorización de sensores con arduino utilizando el protocolo MQTT,» Universitat Politècnica de Catalunya, Barcelona, España, 2019.
- [43] A. M. Dipa Son, «A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT),» Universidad CHARUSAT, India, 2017.
- [44] S. D. Cuji Alex, «NTEGRACIÓN DE CHIRPSTACK CON NODE RED PARA ALOJAR DATOS EN GOOGLE CLOUD PLATFORM,» UNIVERSIDAD POLITÉCNICA SALESIANA, Cuenca, 2022.
- [45] Node-RED, [En línea]. Available: <https://nodered.org/>.
- [46] Q. Carlos, «PANEL DE MONITOREO Y CONTROL REMOTO UTILIZANDO LA PLATAFORMA IBM CLOUD. APLICACIÓN A UNA LINEA DE PRODUCCIÓN VIRTUALIZADA EN 3D,» Universidad Politécnica Salesiana, Guayaquil, 2022.
- [47] C. K. León Carlos, «DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA BASADO EN EL SOFTWARE LIBRE NODE-RED PARA EL MONITOREO Y OPERACIÓN DE LA PLANTA DIDÁCTICA MPS PA. COMPACT WORKSTATION DE FESTO,» Universidad Politécnica Salesiana, Guayaquil, 2021.
- [48] F. R. J. R. Antonio Ruiz, Guía Práctica de Sensores, España: Creaciones, 2010.
- [49] «USINAINFO,» [En línea]. Available: <https://www.usinainfo.com.br/sensores-474>.

- [50] L. Corona, G. Abarca y J. Mares, *Sensores y Actuadores*, México D.F.: Grupo Editorial Patria, 2014.
- [51] A. (. finales), «Slideshare,» 2018. [En línea]. Available: <https://es.slideshare.net/LinderTorricoFlores/actuadores-elementos-finales>.
- [52] Regaber, 2022. [En línea]. Available: <https://regaber.com/pimiento/>.
- [53] Promesa, 2020. [En línea]. Available: <https://www.promesa.com.ec/producto/bomba-paolo-cp2004-1-hp>.
- [54] «SINELEC,» [En línea]. Available: <https://blog.gruposinelec.com/actualidad/que-es-node-red-y-para-que-sirve/>.
- [55] Horticulture, «Diferentes tipos de invernaderos,» Mas Horticulture, Candiana, Italia.
- [56] J. C. B. Vera, «Desarrollo de una red IoT con tecnología LoRa para detección de automóviles,» UNIVERSIDAD POLITÉCNICA SALESIANA, Quito (Ecuador), 2019.
- [57] Rs-Online, «RS,» 16 febrero 2020. [En línea]. Available: <https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things>.
- [58] F. Moreno, «Demostrador arquitectura publish/subscribe con MQTT,» Universitat Politècnica de Catalunya, Barcelona, 2018.
- [59] Promesa, Bombas y Plmería, Ecuador, 2020.

## ANEXO 1

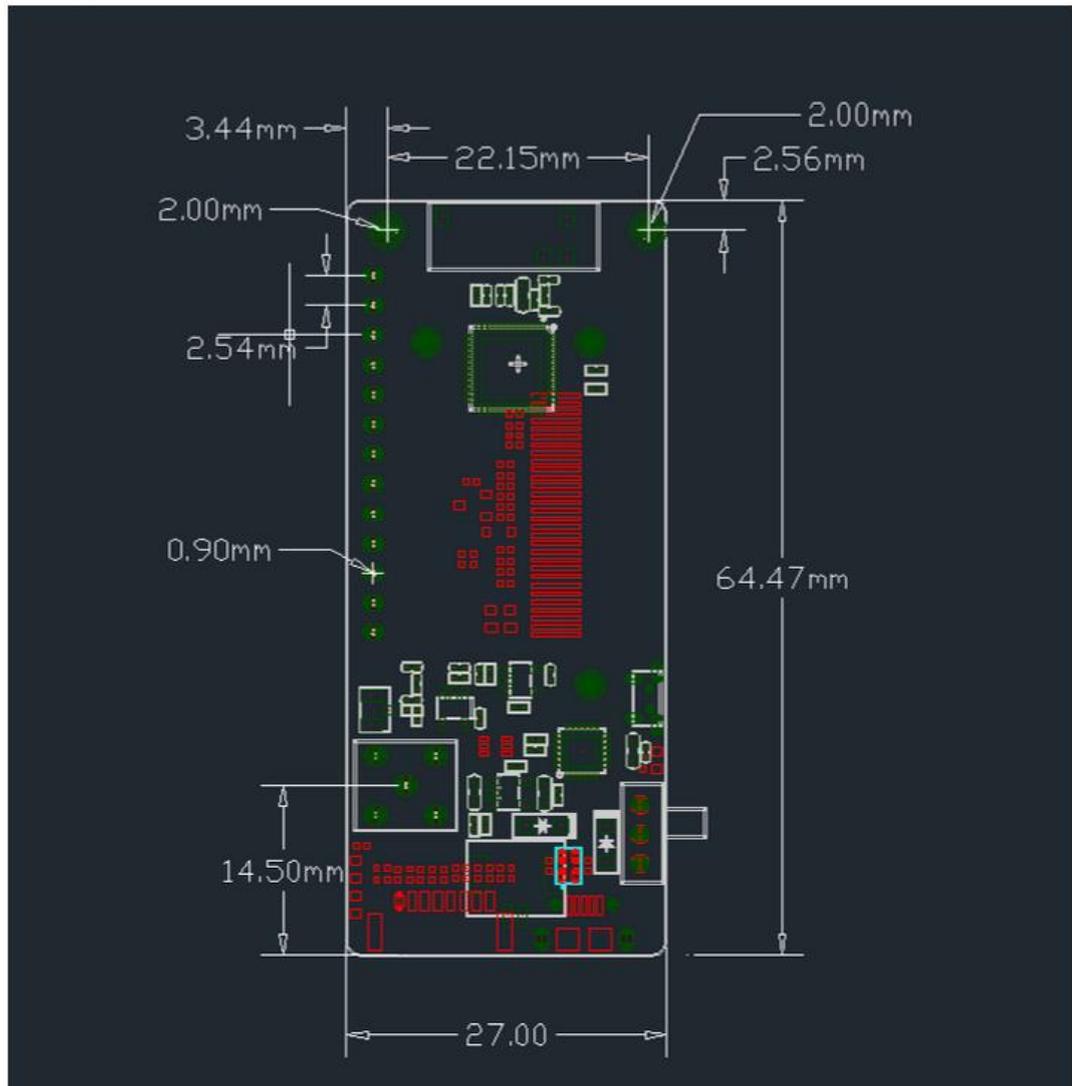
### MÓDULO LORA V2 LILYGO 868/915MHZ

Datasheet del módulo LoRa, diagrama de estructura del módulo y tabla de especificaciones técnicas.



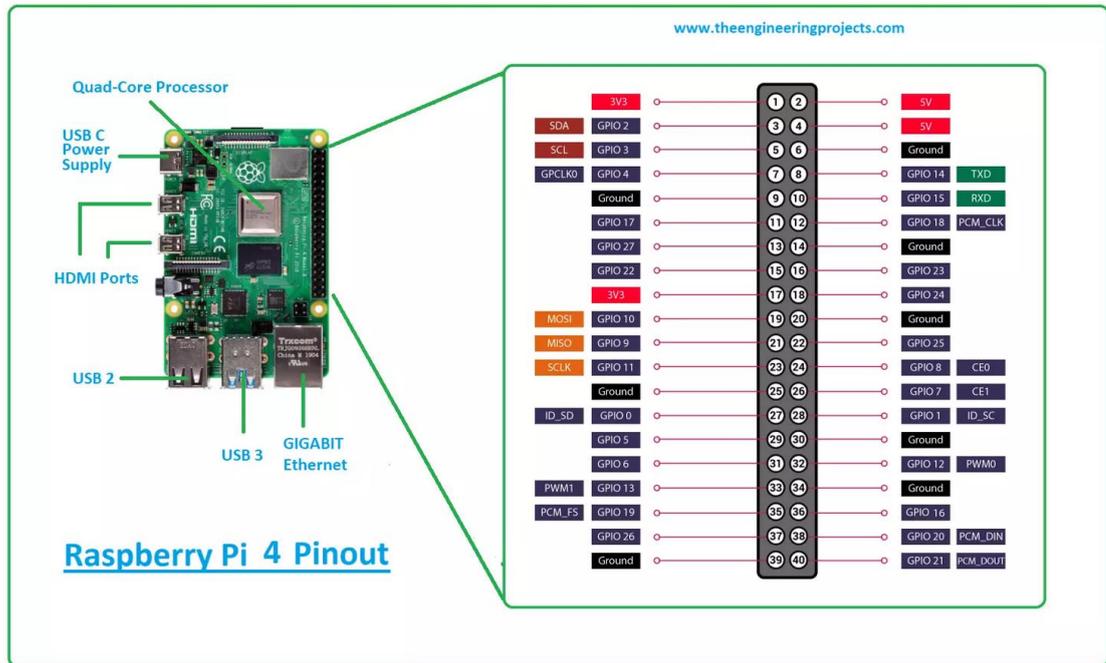
TTGO LORA32 V1.0/ V2.0/ V2.1_1.6 Version			
	TTGO LORA32 V1.0	TTGO LORA32 V2.0	TTGO LORA32 V2.1_1.6
ESPchip	ESP32-D0WDQ6	ESP32 PICO-D4	ESP32 PICO-D4
flash	4M bytes	4M bytes	4M bytes
USB-TO-UART IC	CP2102	CP2104	CP2104
Antenna	3D Antenna	3D Antenna	3D Antenna
MicroSD Card Slot	NO	YES	YES
Battery powered switch	NO	YES	YES
Battery charging	YES	YES	YES
LORA Shielded cover	NO	YES	YES
flash light	IO15	IO22	IO23
LORA antenna	IPX	IPX	SMA
LORA PIN	26, 14, 18, 5, 27, 19	26, 9, 18, 5, 27, 19	26, 23, 18, 5, 27, 19

# TTGO LoRa32 V2.1\_1.6 version Structure diagram



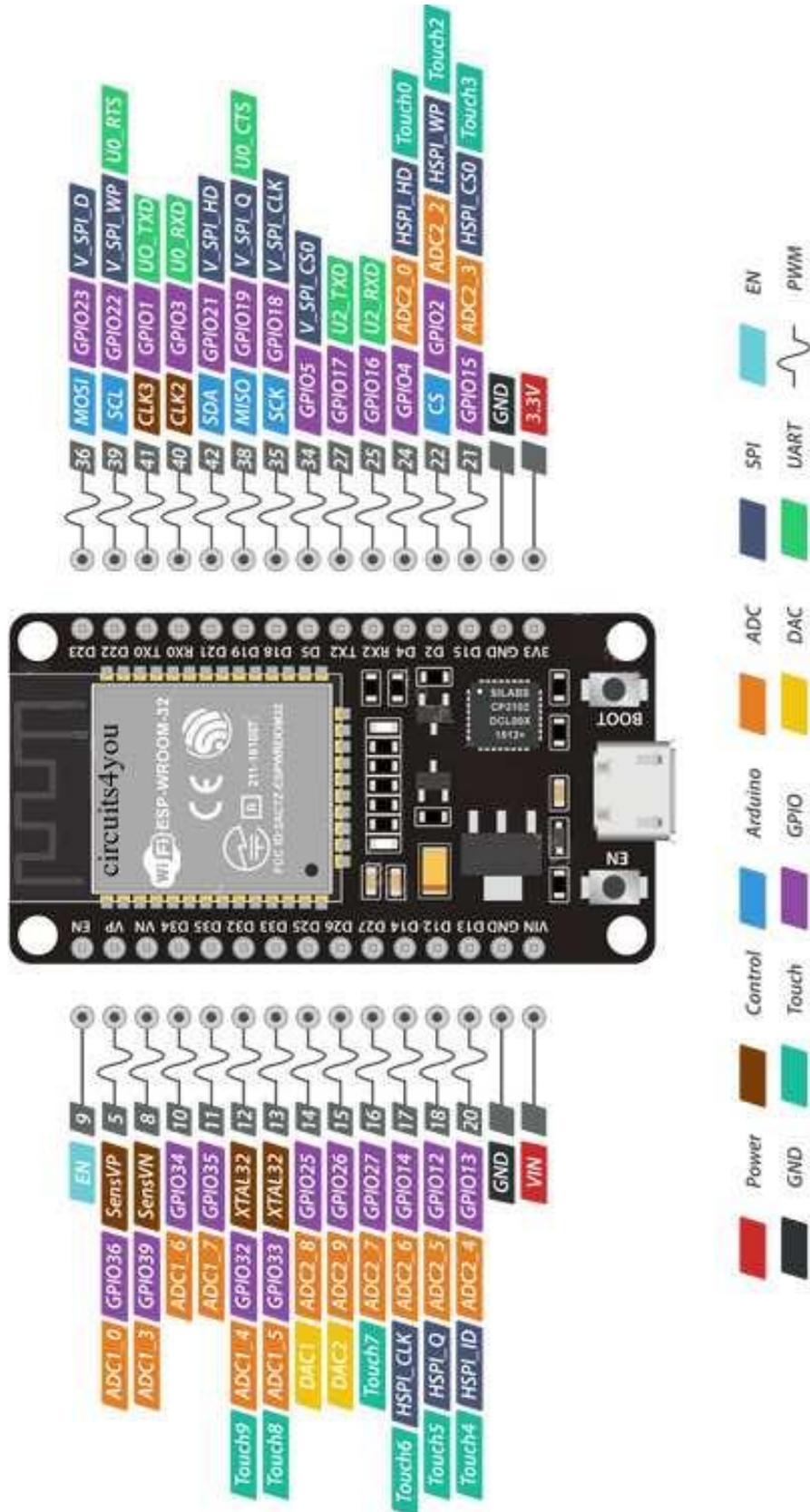
## ANEXO 2

### DATASHEET DE LA RASPBERRY PI 4



# ANEXO 3

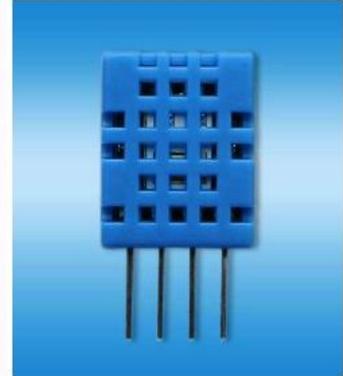
## MÓDULO ESP32



## ANEXO 4

### SENSOR DHT11

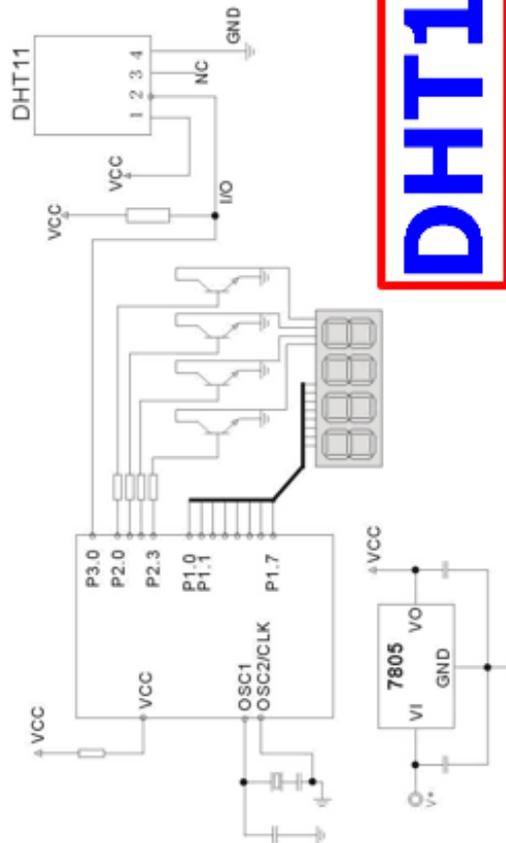
Model	DHT11	
Power supply	3-5.5V DC	
Output signal	digital signal via single-bus	
Sensing element	Polymer resistor	
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius	
Accuracy	humidity +4%RH (Max +-5%RH); temperature +-2.0Celsius	
Resolution sensitivity	or humidity 1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-1Celsius
Humidity hysteresis	+-1%RH	
Long-term Stability	+-0.5%RH/year	
Sensing period	Average: 2s	
Interchangeability	fully interchangeable	
Dimensions	size 12*15.5*5.5mm	



**ADSONG**

Temp, Humidity & Dew point measurement experts

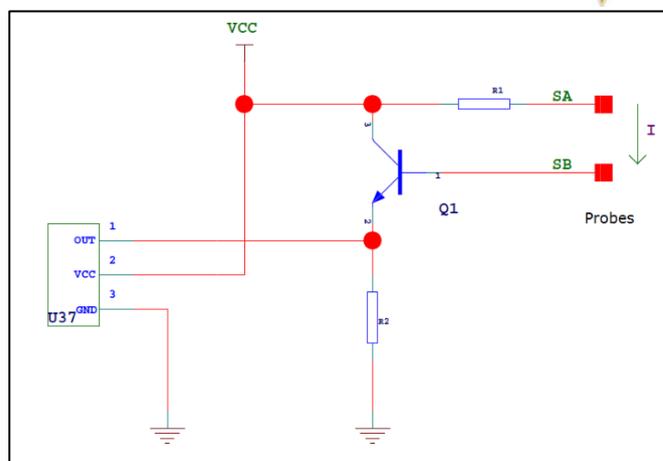
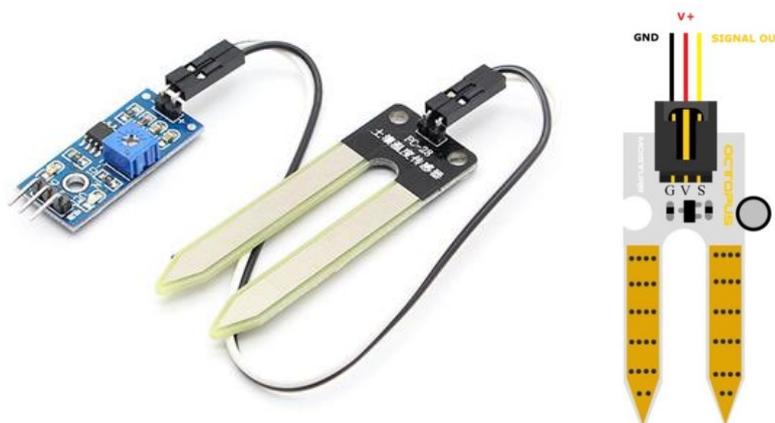
#### 6. Typical circuit



**DHT11**

## ANEXO 5

### SENSOR DE HUMEDAD FC-28



### ESPECIFICACIONES TÉCNICAS

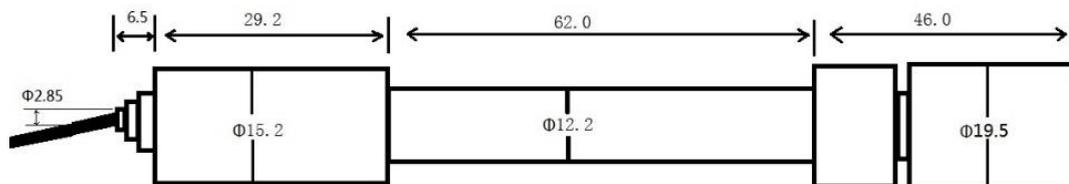
- Voltaje de alimentación: 3.3V – 5V DC (VCC)
- Corriente de operación: 35mA
- Voltaje de señal de salida analógico (AO) : 0 a VCC
- Voltaje de señal de salida digital (DO) : 3.3V/5V TTL
- Opamp LM393 en modo comparador, umbral (threshold) regulable por potenciómetro
- Superficie de electrodo: Estaño
- Incluye: Electrodo, Placa y cable de conexión
- Vida útil electrodo sumergido: 3 a 6 meses
- Dimensiones YL-38: 30\*16 mm
- Dimensiones YL-69: 60\*20\*5 mm
- Peso:

### CONEXIONES

- **VCC:** Voltaje de alimentación (3.3V – 5V DC)
- **GND:** Tierra (GND 0V)
- **DO:** Salida digital
- **AO:** Salida analógica

## ANEXO 6

### SENSOR DE PH SKU: SEN0161

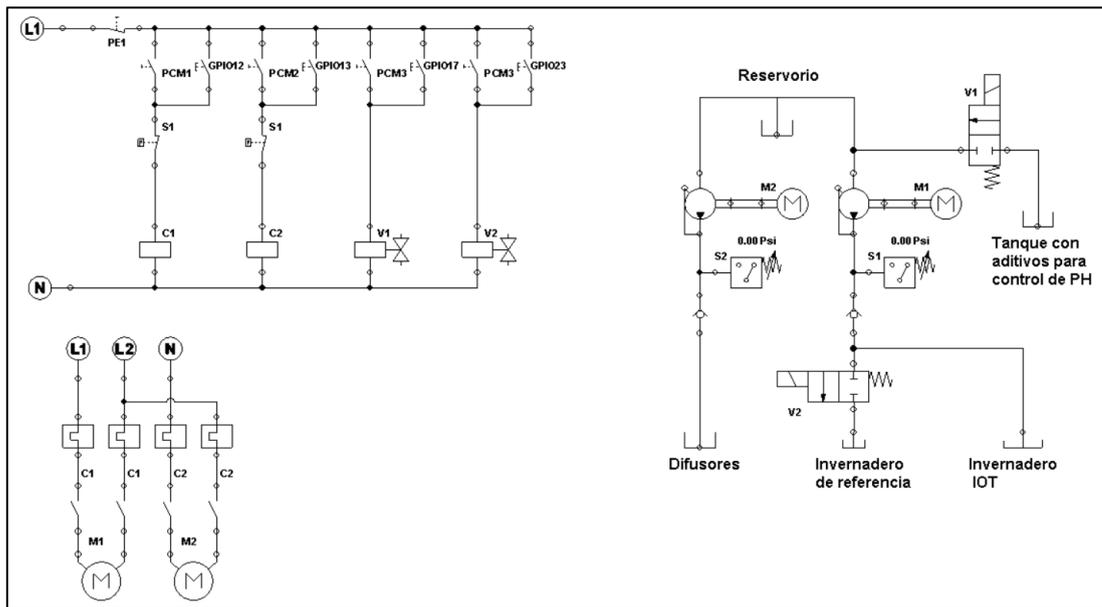
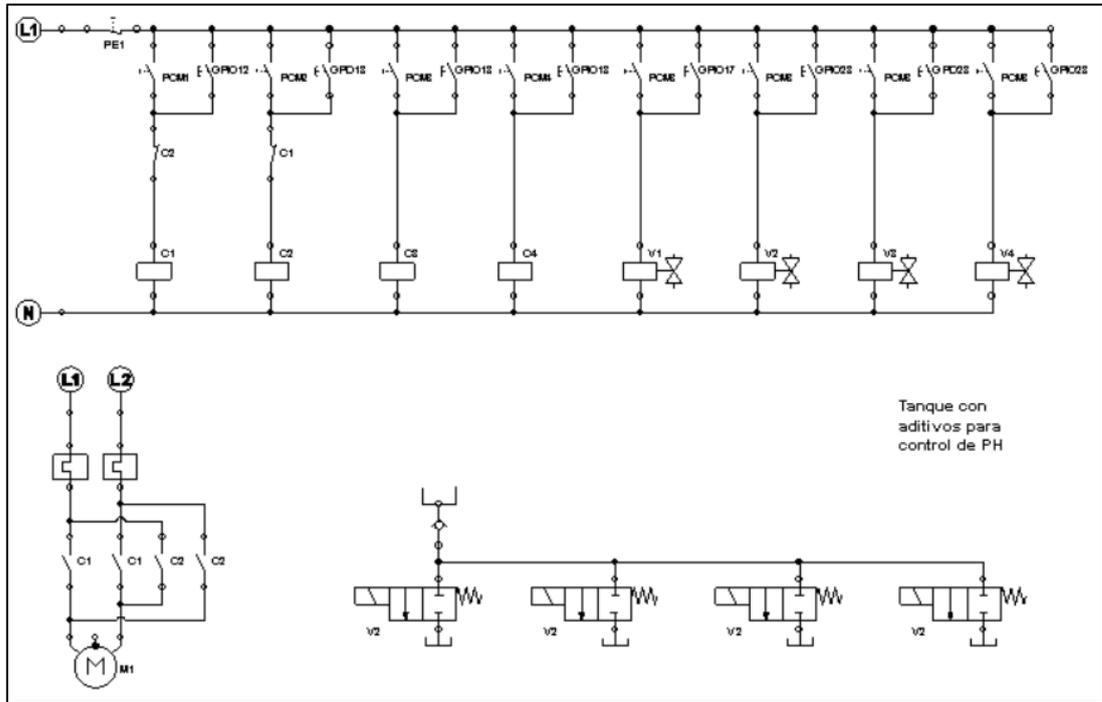


## Especificación

- Potencia del módulo: 5.00V
- Tamaño del módulo: 43 mm × 32 mm
- Rango de medición: 0-14PH
- Temperatura de medición: 0-60 °C
- Precisión:  $\pm 0.1$ pH (25 °C)
- Tiempo de respuesta:  $\leq 1$  min.
- Sensor de pH con conector BNC
- Interfaz PH2.0 (parche de 3 pines)
- Potenciómetro de ajuste de ganancia
- LED del indicador de encendido
- Longitud del cable del sensor al conector BNC: 660 mm

# ANEXO 7

## CIRCUITOS DE POTENCIA



## ANEXO 8

### VENTILADOR INDUSTRIAL PE45-T1 130W



- Garantía española
- Potencia: 120 W.
- Tamaño de aspas: 45cm.
- Rejilla de protección.
- Cabezal orientable.
- Base estable.
- 3 velocidades de ventilación.
- Asa de transporte.

## ANEXO 9

### MOTOR FURIUS FL500

**FURIUS**

MOTOR PARA PUERTAS CORREDIZAS

**FLASH FL500**

SLIDING GATE OPERATOR FOR 500 kg DOORS  
MOTOR PARA PUERTAS CORREDIZAS DE 500 Kg

**1 YEAR WARRANTY**

Función de cierre automático

Facil instalación

Detección de obstáculos

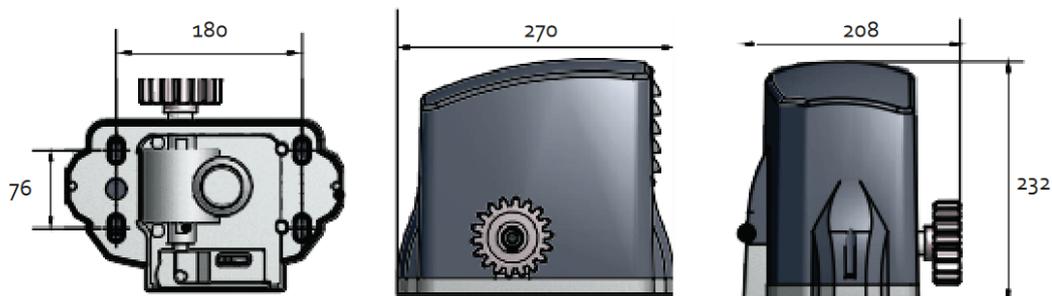
**ACCESORIOS INCLUIDOS:**

- Control remoto x 4m
- Control remoto x 3 (FL54-1)

VOLTAJE DE ENTRADA	110V
FRECUENCIA	60/50Hz
PESO MAX. DE LA PUERTA	500kg
FRECUENCIA DEL CONTROL	433,53MHz
VELOCIDAD	17 m/min
FINAL DE CARRERA	Magnético
LONGITUD DE CREMALLERAS	4m
CICLOS/HORA	20
TIEMPO APERTURA	10,5s

**1 YEAR WARRANTY**

**FL500**

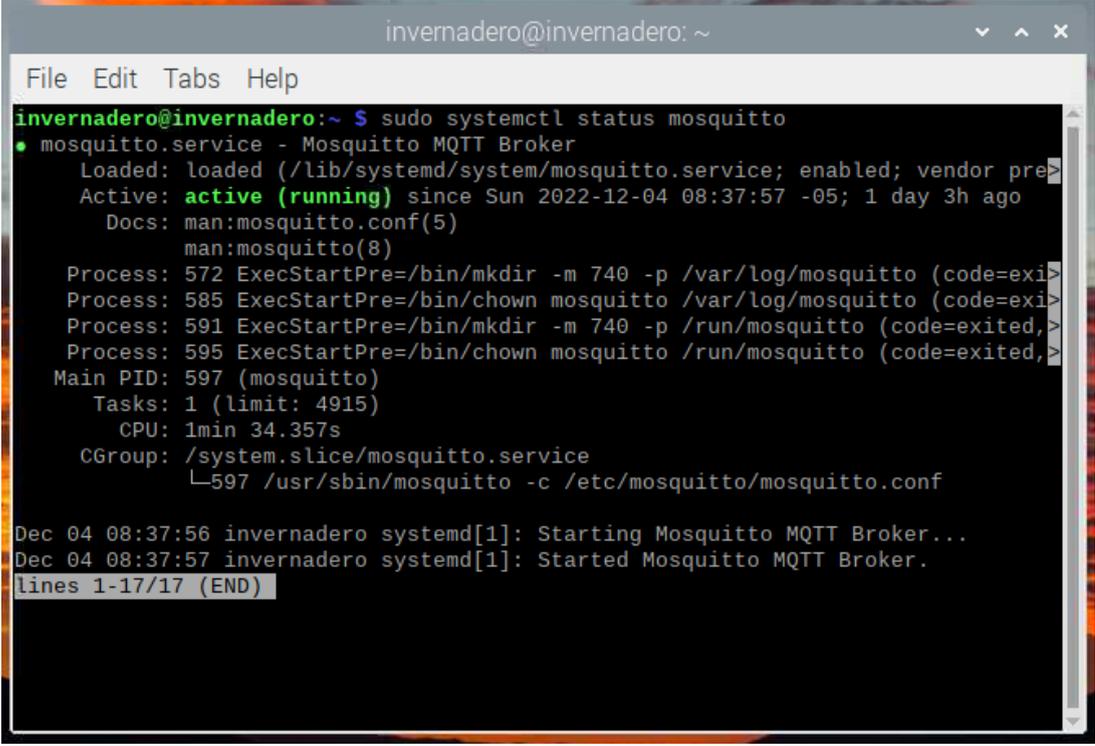


Descripción técnica	FL500	FL500S	FL600	FL800	FL800S
Alimentación	110V	110V	110V	110V	110V/220V
Central	STORM	ROCKET	ROCKET	ROCKET	BOOSTER
Maniobras por hora (3m)	20	30	25	35	60
Peso máximo del portón	500Kg	500Kg	600Kg	800Kg	800Kg
Peso del equipo (kg)	4,52 Kg	5,21 Kg	5,29 Kg	5,83 Kg	4,73 Kg
Rotación	1740 RPM	1740 RPM	1740 RPM	1740 RPM	4200 RPM
Temperatura de trabajo	-5°C a 55°C				
Tiempo de apertura (3 m)	10,5 SEG.	9 SEG.	10,5 SEG.	10,5 SEG.	4 SEG.
Velocidad nominal	17 m/min	20,9 m/min	17 m/min	17 m/min	41,4 m/min
Clase	I	I	I	I	I
ip	24	24	24	24	24
Cremallera	4m	4m	4m	4m	4m

## ANEXO 10

### INSTALACIÓN DEL PROTOCOLO MQTT

Para la instalación del protocolo MQTT el primer paso es ingresar al modo root y después se procede a ingresar el comando “sudo apt install mosquitto”, después de ingresar dicho comando el protocolo se instala automáticamente y cuando nos solicita autorización para continuar se ingresa el comando “s”, una vez se autorice el protocolo MQTT procede a finalizar su instalación. para verificar que el protocolo MQTT fue instalado correctamente se debe ingresar el comando “sudo systemctl status mosquitto”, y deberá salir que se encuentra activo y ejecutándose como tal y como se puede ver en la figura siguiente. Si es que el protocolo MQTT se encuentra desactivado se procederá a ingresar el comando “sudo systemctl start mosquitto”.

A terminal window titled 'invernadero@invernadero: ~' showing the command 'sudo systemctl status mosquitto' and its output. The output indicates that the 'mosquitto.service' is loaded and active (running) since Sun 2022-12-04 08:37:57 -05; 1 day 3h ago. It lists several processes involved in the service's execution, including 'mkdir' and 'chown' commands, and shows the main PID as 597. Log messages at the bottom confirm the service's start time.

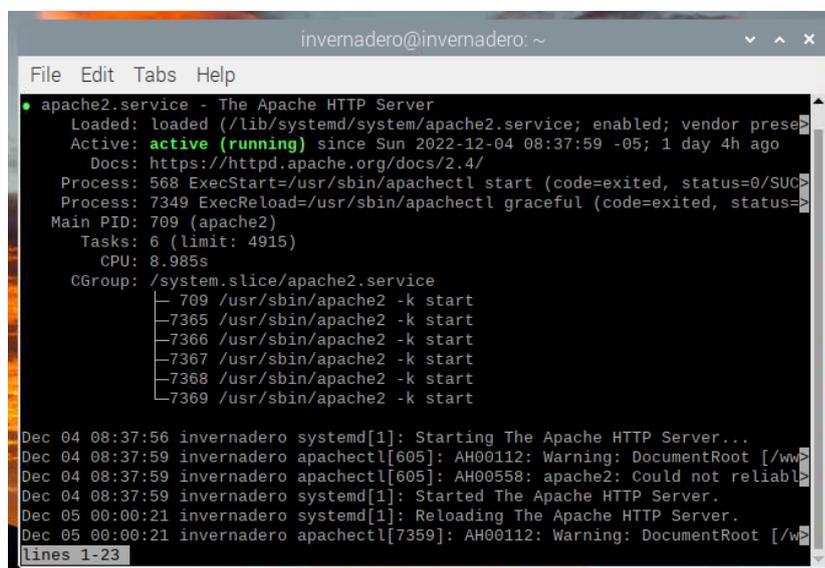
```
invernadero@invernadero:~ $ sudo systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre>
   Active: active (running) since Sun 2022-12-04 08:37:57 -05; 1 day 3h ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 572 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exi>
   Process: 585 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exi>
   Process: 591 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited,>
   Process: 595 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited,>
   Main PID: 597 (mosquitto)
     Tasks: 1 (limit: 4915)
        CPU: 1min 34.357s
   CGroup: /system.slice/mosquitto.service
           └─597 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Dec 04 08:37:56 invernadero systemd[1]: Starting Mosquitto MQTT Broker...
Dec 04 08:37:57 invernadero systemd[1]: Started Mosquitto MQTT Broker.
lines 1-17/17 (END)
```

## ANEXO 11

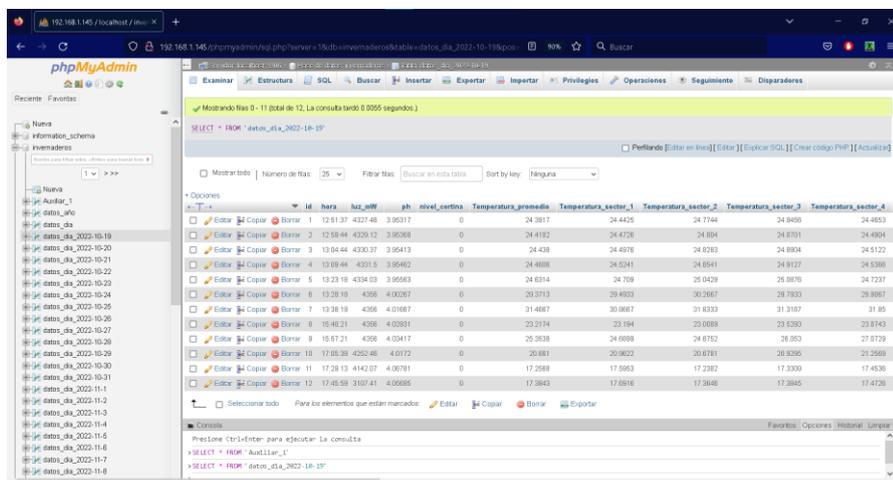
### INSTALACIÓN DEL SERVIDOR LAMP

El servidor LAMP es del conjunto de cuatro tecnologías que conforman una plataforma que trabaja del lado del servidor, estas cuatro tecnologías son: Linux, Apache, MySQL y PHP. A poder instalar el servidor lan es necesario primero actualizar los repositorios y una vez actualizados se procede a su instalación a través del comando “sudo apt install apache2”. Una vez instalado el servidor LAMP se puede verificar que su instalación fue correcta y se encuentra activo a través del comando “sudo systemctl status apache2”. En la siguiente figura se puede verificar que el servidor la amp está corriendo correctamente y se encuentra activo.



```
invernadero@invernadero: ~  
File Edit Tabs Help  
● apache2.service - The Apache HTTP Server  
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese  
  Active: active (running) since Sun 2022-12-04 08:37:59 -05; 1 day 4h ago  
  Docs: https://httpd.apache.org/docs/2.4/  
  Process: 568 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC  
  Process: 7349 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=>  
  Main PID: 709 (apache2)  
  Tasks: 6 (limit: 4915)  
  CPU: 8.985s  
  CGroup: /system.slice/apache2.service  
        └─ 709 /usr/sbin/apache2 -k start  
          └─ 7365 /usr/sbin/apache2 -k start  
            └─ 7366 /usr/sbin/apache2 -k start  
              └─ 7367 /usr/sbin/apache2 -k start  
                └─ 7368 /usr/sbin/apache2 -k start  
                  └─ 7369 /usr/sbin/apache2 -k start  
  
Dec 04 08:37:56 invernadero systemd[1]: Starting The Apache HTTP Server...  
Dec 04 08:37:59 invernadero apachectl[605]: AH00112: Warning: DocumentRoot [/ww  
Dec 04 08:37:59 invernadero apachectl[605]: AH00558: apache2: Could not reliabl  
Dec 04 08:37:59 invernadero systemd[1]: Started The Apache HTTP Server.  
Dec 05 00:00:21 invernadero systemd[1]: Reloading The Apache HTTP Server.  
Dec 05 00:00:21 invernadero apachectl[7359]: AH00112: Warning: DocumentRoot [/w  
lines 1-23
```

Una vez instalado el servidor también se instala phpMyAdmin, una base de datos la cual se utilizará para almacenar los datos adquiridos por los sensores para proceder a visualizarnos y el HMI.



## ANEXO 12

### INSTALACIÓN DE NODE-RED

Node-RED es una herramienta de programación visual que permite apreciar de forma gráfica los enlaces y relaciones que va viendo entre cada bloque según se avanza en la programación.

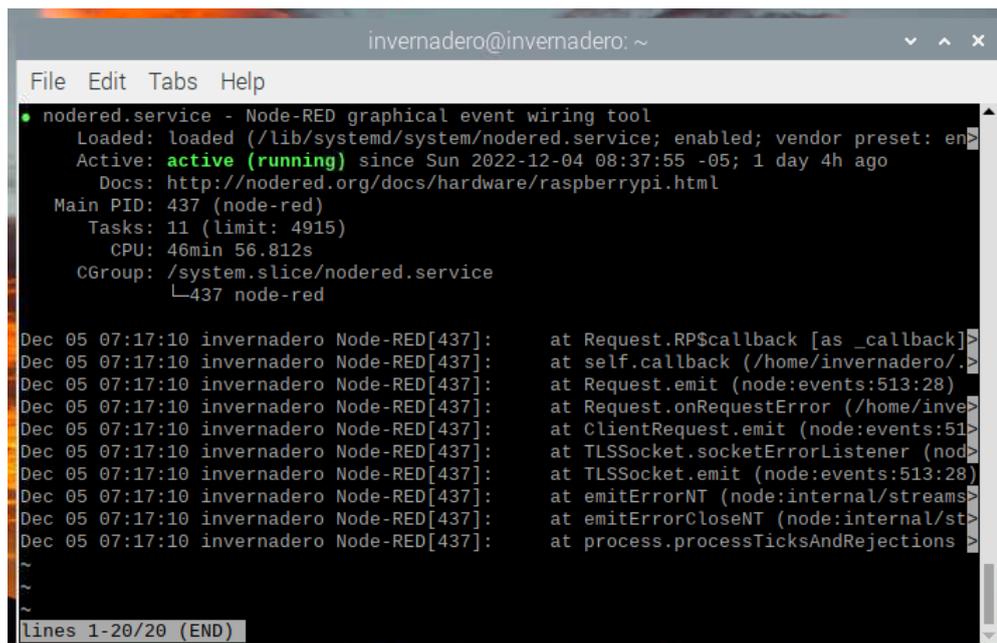
Para proceder a instalar el servicio no red primero es necesario instalar los requisitos que necesita este servicio, estos son Node.js y npm, estos requisitos van a permitir el diseño del HMI, donde se podrá visualizar los datos, el gráficos y promedios del invernadero en tiempo real, así como dar órdenes hacia los actuadores. Para la instalación de estos prerrequisitos se utilizan los siguientes comandos,

```
sudo apt install nodejs
```

para instalar los archivos Node.js, para instalar los archivos npm,

```
sudo apt install npm
```

Una vez instalados estos prerrequisitos se procede a instalar el servicio Node-RED, por ende, se procede a su instalación a través del comando “sudo npm install -g -unsafe-perm node-red”, es necesario abrir el puerto para qué la comunicación entre el servidor y Node-RED, en este caso se hace uso del puerto 1880. Una vez instalado se verifica que el servicio de no de red se encuentra activo y ejecutándose para ello se utiliza el comando “sudo systemctl status nodered”, como se puede ver en la siguiente gráfica el servicio Node-RED sí está trabajando.



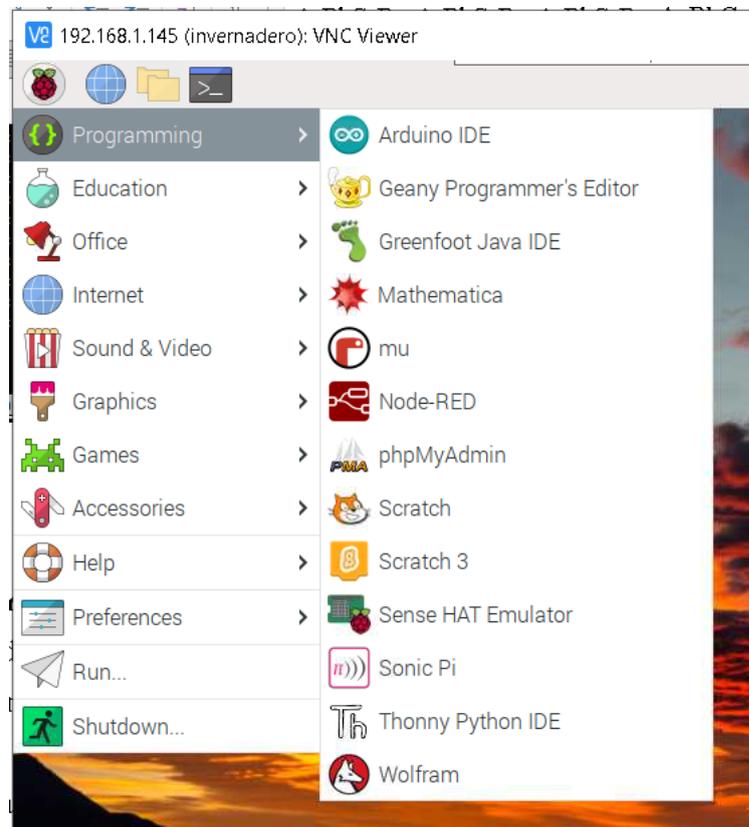
```
invernadero@invernadero: ~  
File Edit Tabs Help  
● nodered.service - Node-RED graphical event wiring tool  
Loaded: loaded (/lib/systemd/system/nodered.service; enabled; vendor preset: en  
Active: active (running) since Sun 2022-12-04 08:37:55 -05; 1 day 4h ago  
Docs: http://nodered.org/docs/hardware/raspberrypi.html  
Main PID: 437 (node-red)  
Tasks: 11 (limit: 4915)  
CPU: 46min 56.812s  
CGroup: /system.slice/nodered.service  
└─437 node-red  
  
Dec 05 07:17:10 invernadero Node-RED[437]: at Request.RP$callback [as _callback]  
Dec 05 07:17:10 invernadero Node-RED[437]: at self.callback (/home/invernadero/.  
Dec 05 07:17:10 invernadero Node-RED[437]: at Request.emit (node:events:513:28)  
Dec 05 07:17:10 invernadero Node-RED[437]: at Request.onRequestError (/home/inve  
Dec 05 07:17:10 invernadero Node-RED[437]: at ClientRequest.emit (node:events:51  
Dec 05 07:17:10 invernadero Node-RED[437]: at TLSSocket.socketErrorListener (nod  
Dec 05 07:17:10 invernadero Node-RED[437]: at TLSSocket.emit (node:events:513:28)  
Dec 05 07:17:10 invernadero Node-RED[437]: at emitErrorNT (node:internal/streams  
Dec 05 07:17:10 invernadero Node-RED[437]: at emitErrorCloseNT (node:internal/st  
Dec 05 07:17:10 invernadero Node-RED[437]: at process.processTicksAndRejections  
~  
~  
lines 1-20/20 (END)
```

## ANEXO 13

### INSTALACIÓN DE ARDUINO IDE

Se procede a instalar Arduino IDE en la raspberry para poder tener un control remoto de la programación que se encuentra instalada en el gateway, así como realizar diferentes cambios si es que fuera necesario de manera fácil y rápida en cualquiera de las programaciones utilizadas tanto en los nodos Lora, Gateway o nodos sensores.

Para la instalación de Arduino IDE se procede a insertar el comando “sudo apt-get install arduino arduino-core”, la herramienta de programación Arduino procede a instalarse y se puede verificar a través del icono de raspbian, que se encuentra en la parte superior izquierda del escritorio.



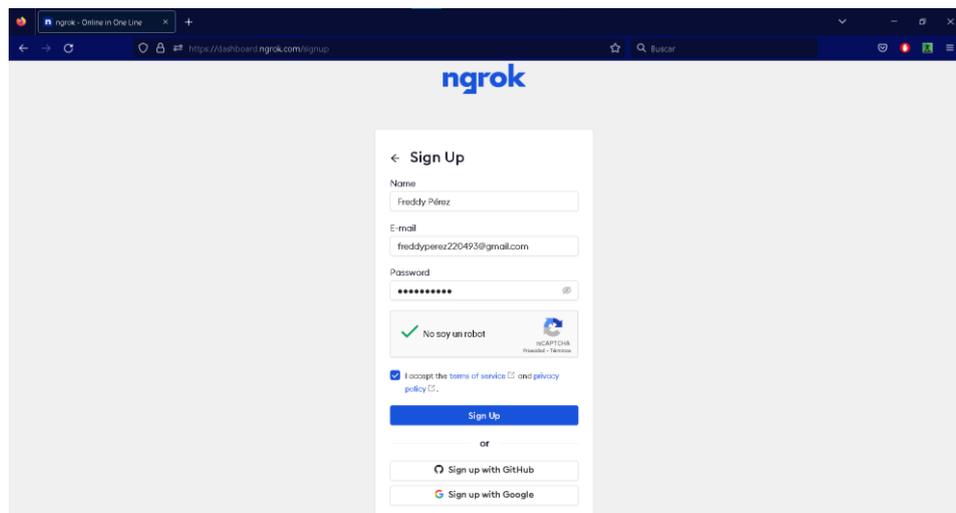
En la figura anterior también se puede apreciar que se encuentra instalado correctamente el servicio de Node-RED y phmMyAdmin.

## ANEXO 14

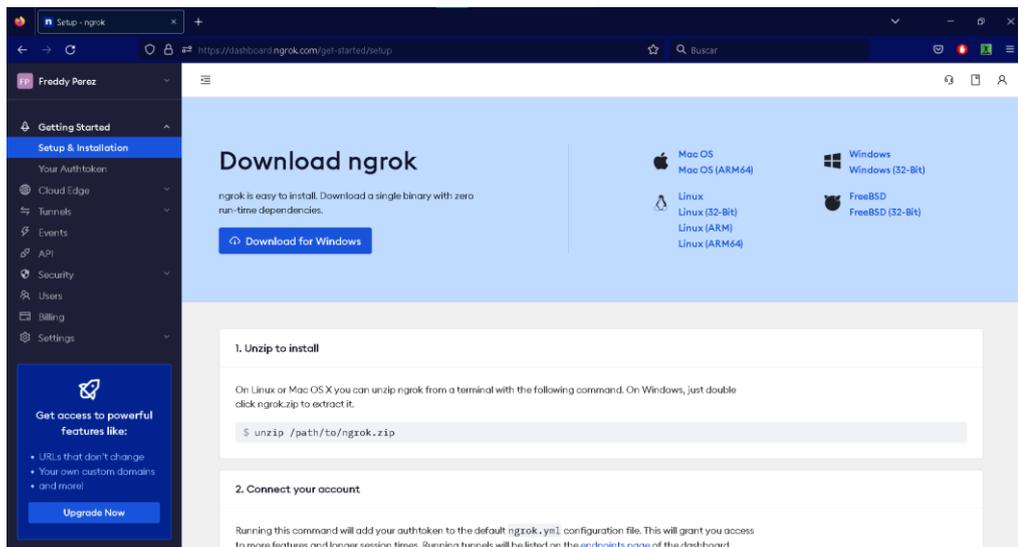
### INSTALACIÓN DE NGROK

Ngrok un servicio que permite exponer una URL fuera de una red LAN, para poder visualizarla a través de internet, ya que se pretende controlar el sistema desde cualquier parte del mundo a través de internet es necesario la instalación de este servicio.

Primero se accede a la página oficial de negro y se procede al registro, introduciendo nombre, email y contraseña. una vez aceptado los términos el registro se realiza con éxito, en la siguiente imagen se puede comprobar el registro.



Una vez se realizó el registro con éxito se abre una ventana en la cual en Ngrok detalla la manera de instalar su servicio, de conectarlo y un comando de ayuda, tal como se puede visualizar en la siguiente figura, en la esquina superior izquierda se puede comprobar que la cuenta es la del investigador del proyecto.



En la siguiente figura se puede apreciar el comando que en negro recomienda para la

instalación de su servicio, por ende, solamente se copia y pega el comando y se produce la instalación.

### 1. Unzip to install

On Linux or Mac OS X you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip to extract it.

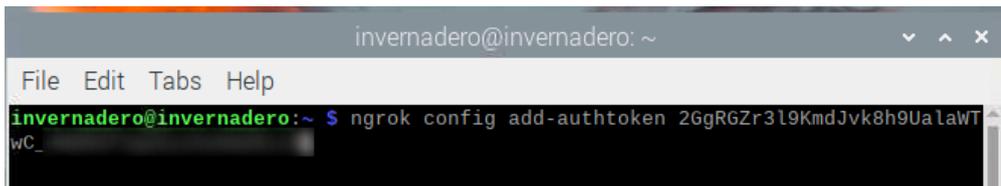
```
$ unzip /path/to/ngrok.zip
```

Una vez instalado el servicio en Ngrok el siguiente paso es conectar la cuenta, en la siguiente figura se puede apreciar el token de autenticación que brinda Ngrok, el comando que se puede apreciar solo se copia y se pega tal como está para que se pueda conectar la cuenta.

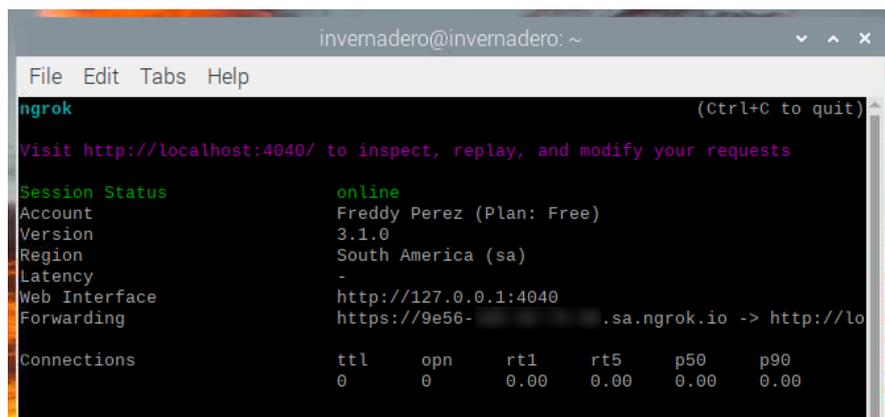
### 2. Connect your account

Running this command will add your authtoken to the default `ngrok.yml` configuration file. This will grant you access to more features and longer session times. Running tunnels will be listed on the [endpoints page](#) of the dashboard.

```
$ ngrok config add-authtoken 2GgRGZr3l9KmdJvk8h9UalaWTwC_
```



Para poder verificar que el servicio es negro está trabajando correctamente se ingresa el comando “ngrok http 80”, en la siguiente figura se puede observar que el servicio está corriendo correctamente.



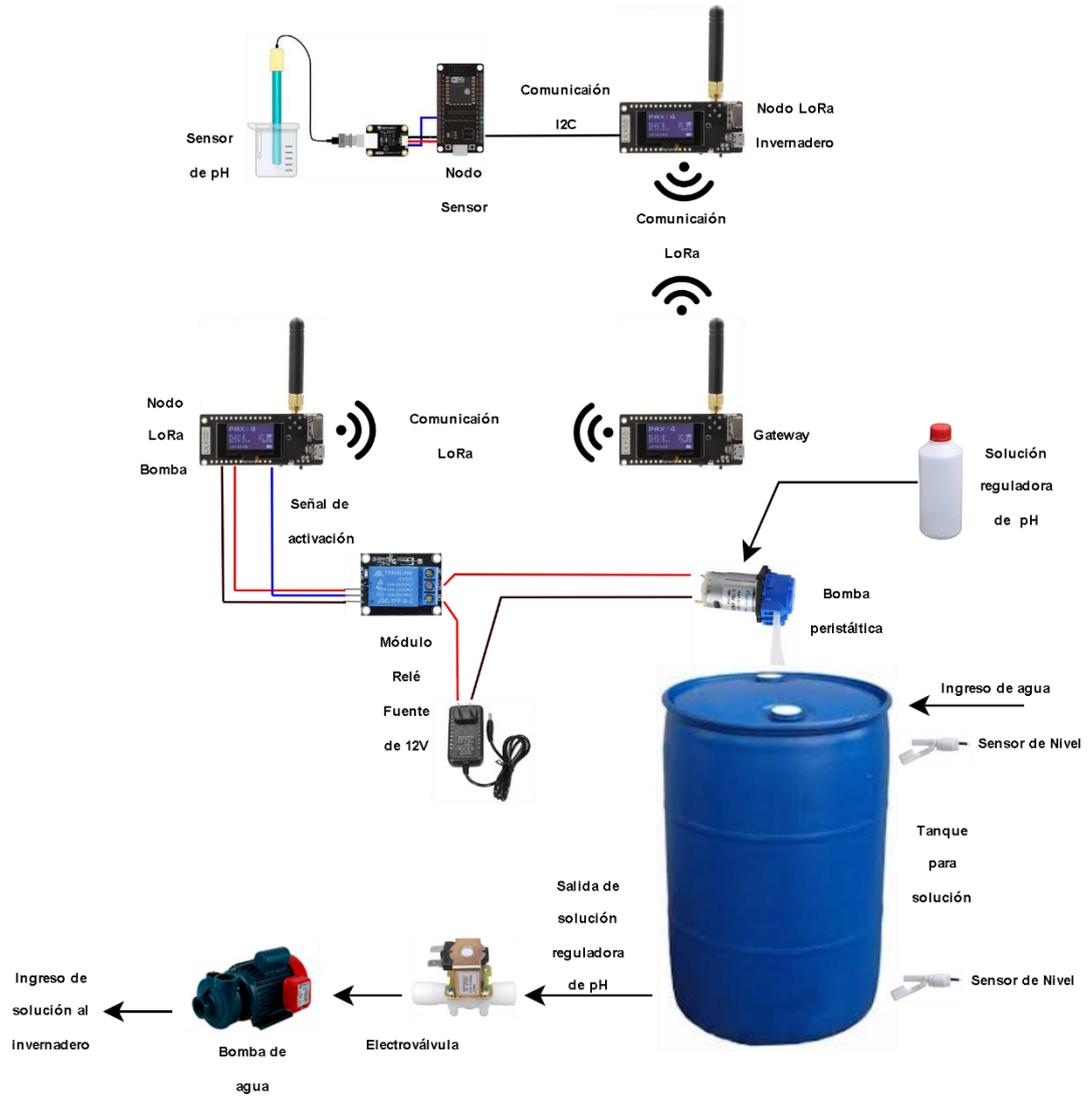
## ANEXO 15

### ILUMINACIÓN NOCTURNA DEL INVERNADERO



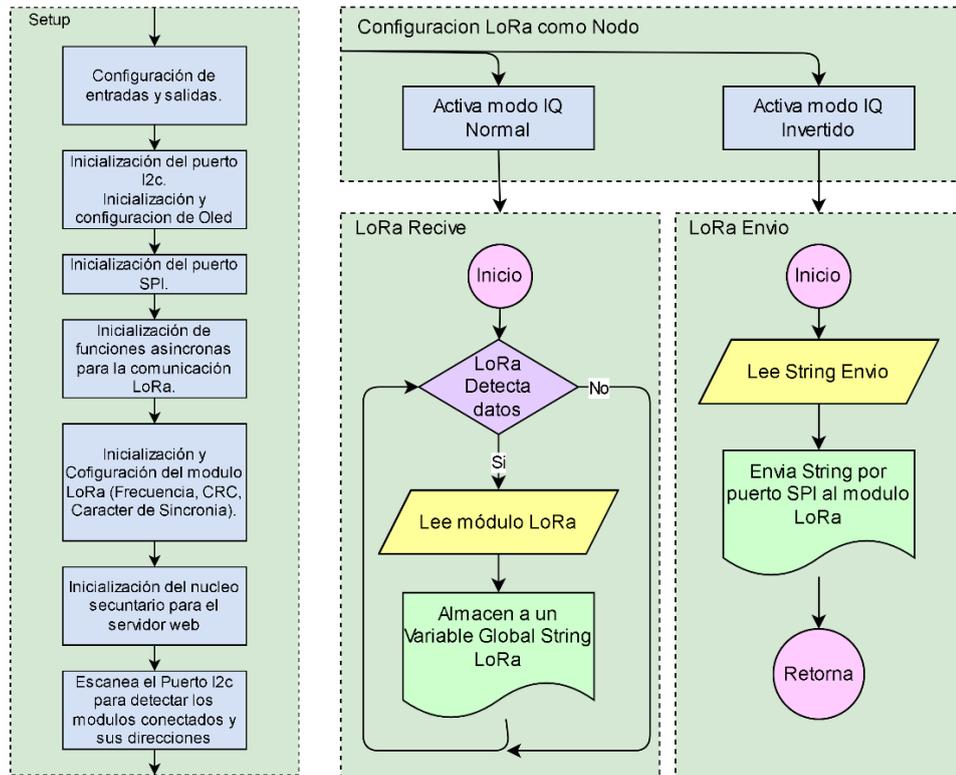
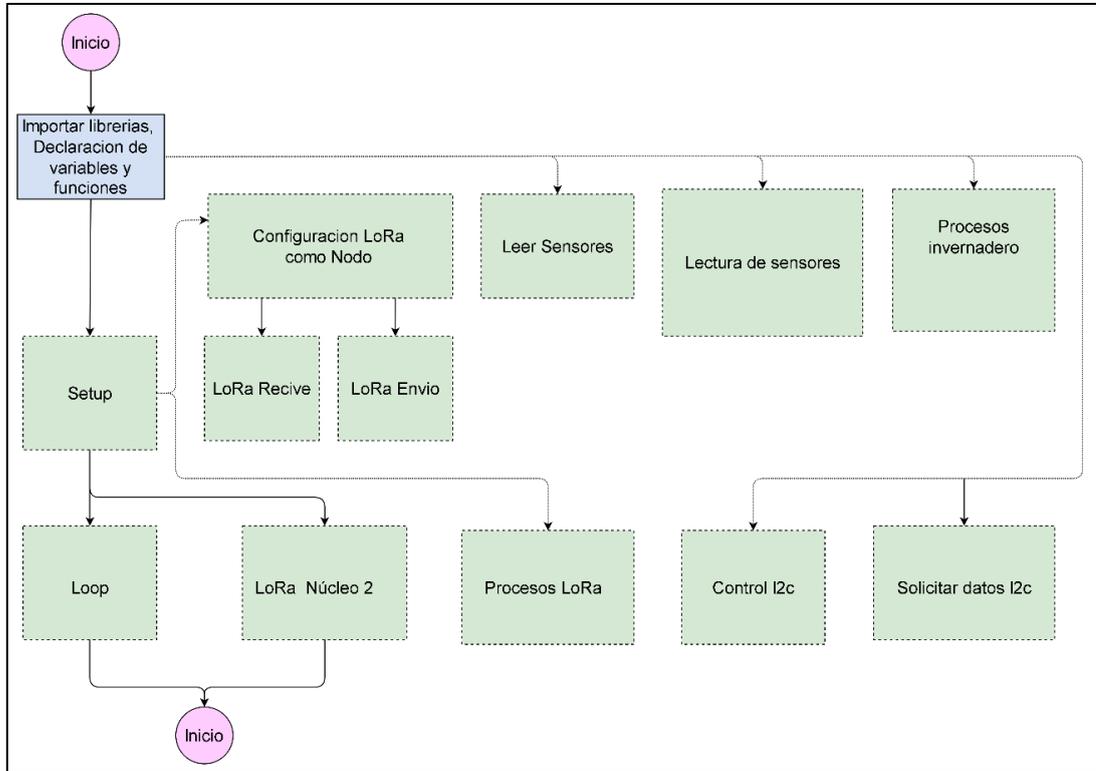
## ANEXO 16

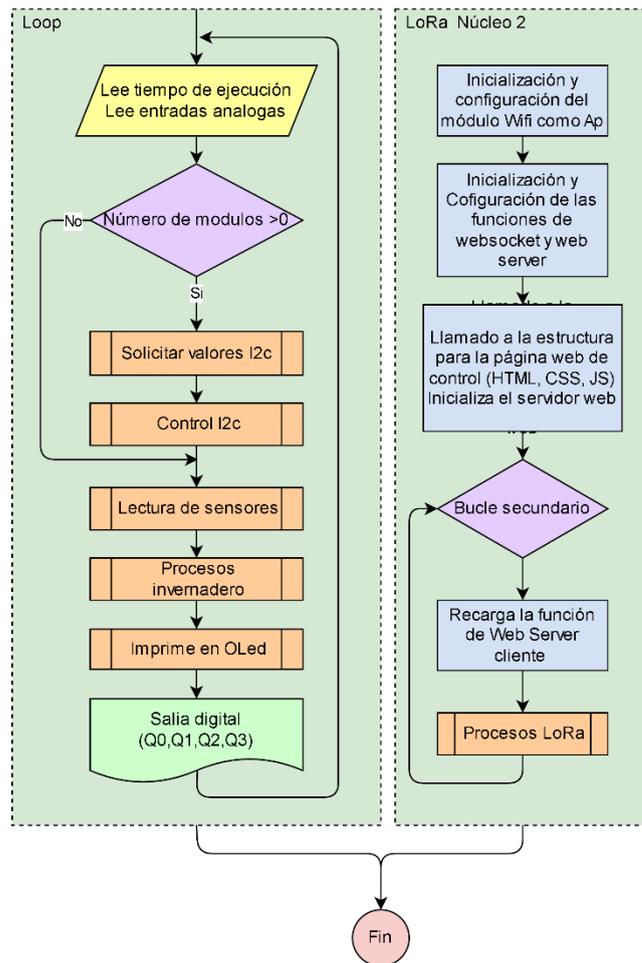
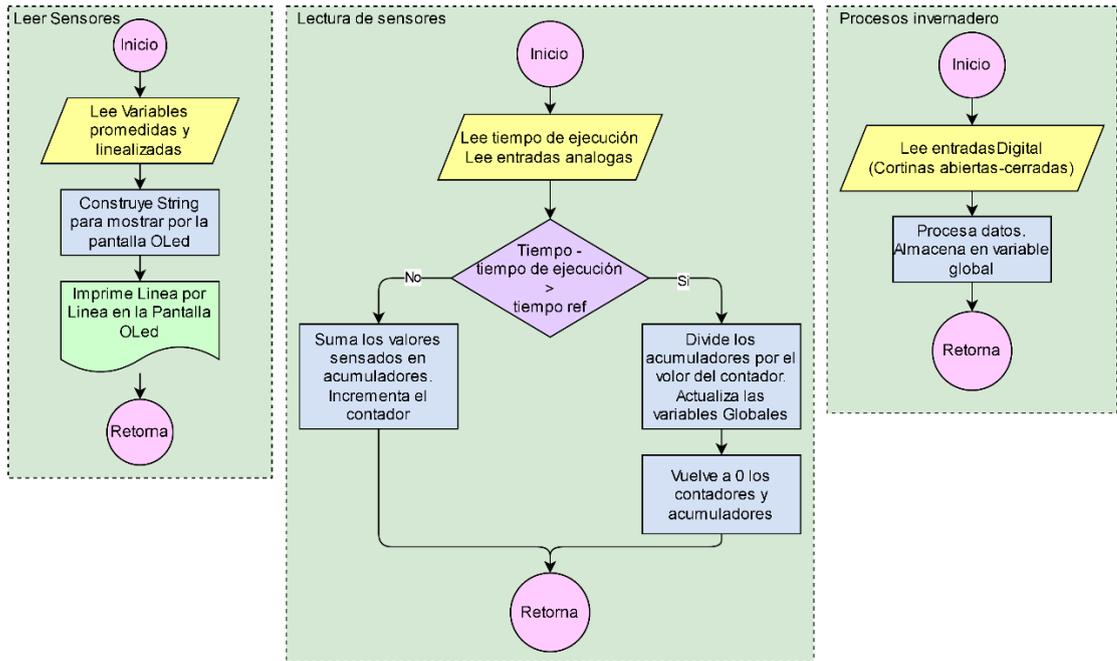
# DIAGRAMA DE CONEXIÓN PARA LA AUTOMATIZACIÓN DEL PH

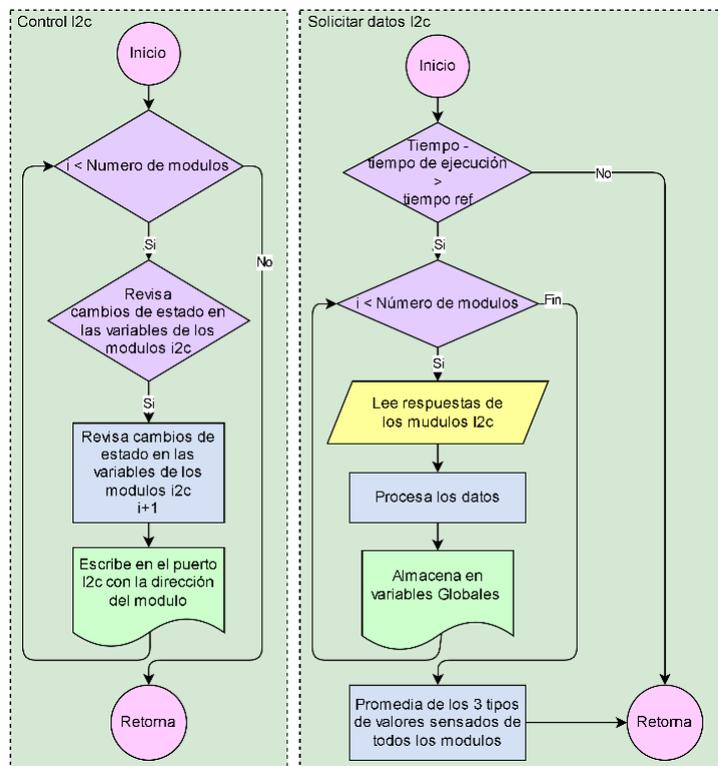
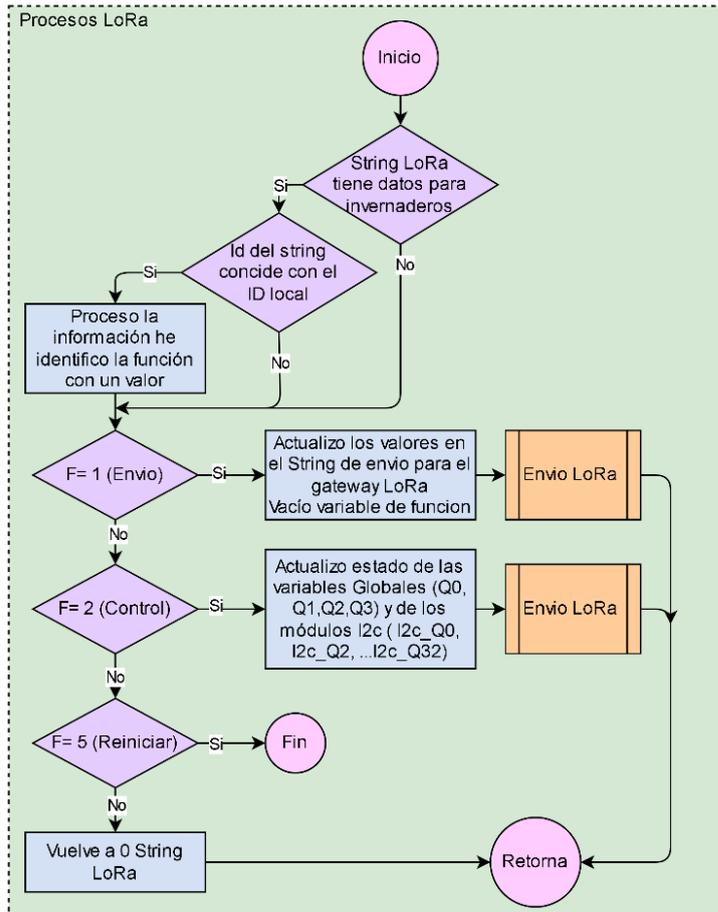


## ANEXO 17

# LÓGICA DE PROGRAMACIÓN DEL NODO LORA INVERNADERO







## ANEXO 18

### PROGRAMACIÓN DEL NODO LORA INVERNADERO

```
#include <Arduino.h>

#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#include <SPIFFS.h>
#include <ESPAsyncWebServer.h>
#include <AsyncTCP.h>

#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 23
#define DIO0 26
#define SDA 21
#define SCL 22
#define pin_sensor_luz 36
#define pin_sensor_ph 39
#define pin_sensor_cortinas1 34
#define pin_sensor_cortinas2 35
#define q0 1 //web page q 1
#define q1 12 //web page q 12
#define q2 4 //web page q 4
#define q3 25 //web page q 25

#define BAND 915E6
#define OLED_RST -1
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define resistor 10000
#define volt 3.3

const byte syn_byte = 255;
const byte local_id = 16;
const byte ga_id = 255;
const byte no_id = 16;
boolean contactores_locales_stado[] = { 0, 0, 0, 0 };
boolean contactor_i2c_stado[] = {
  0, 0,
  0, 0,
  0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, // Sectores
```

```

    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 // Sectores
};
boolean st_dif = 0, wf = 1, aux_c_s[16], pc;
byte fun, dir_sect_i2c[16], n_dir_sect = 0;
int cont_f = 0;
float tmp_s[15], hum_s[15], hums_s[15]; // i2c
float tmp_p, hum_p, hums_p; // i2c
float pW, vr_luz, vr_ph;
float s_luz = 0, s_ph = 0;
unsigned long now;
unsigned long timeaux_i2c_oled = 0;
unsigned long timeaux_i2c = 0; // i2c
unsigned long timeaux_filtro = 0;
String hostname = " ", message1 = "", lora_r = "", oled_line[10], datoI2Creceive = "",
LoRa_receive = "", datoI2CtoLora;

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RST);
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");
TaskHandle_t Task1;

void LoRa_rxMode() {
    LoRa.enableInvertIQ(); // invertir activamente las señales I y Q
    LoRa.receive(); // establecer el modo de recepción
}
void LoRa_txMode() {
    LoRa.idle(); // establece el modo de espera
    LoRa.disableInvertIQ(); // modo normal
}
void LoRa_sendMessage(String message) {
    LoRa_txMode(); // establece el modo tx
    LoRa.beginPacket(); // inicia el paquete
    LoRa.print(message); // añade la carga útil
    LoRa.endPacket(true); // termina el paquete y lo envía
    //Serial.println(message);
}
void onReceive(int packetSize) {
    //String message = "";
    while (LoRa.available()) {
        lora_r = LoRa.readString();
    }
    //Serial.print("Node Receive: ");
    //Serial.println(lora_r);
}
void onTxDone() {
    // Serial.println("TxDone");
    LoRa_rxMode();
}

```

```

}

void notifyClients(String state) {
    ws.textAll(state);
}

void handleMessage(void* arg, uint8_t* data, size_t len) {
    AwsFrameInfo* info = (AwsFrameInfo*)arg;
    if (info->final && info->index == 0 && info->len == len && info->opcode ==
WS_TEXT) {
        data[len] = 0;
        message1 = (char*)data;
        //Serial.println(message1);
        if (message1.indexOf("ql") >= 0) {
            contactores_locales_stado[1] = !contactores_locales_stado[1];
            notifyClients("ql" + String(contactores_locales_stado[1]));
        }
        if (message1.indexOf("qv") >= 0) {
            contactores_locales_stado[2] = !contactores_locales_stado[2];
            notifyClients("qv" + String(contactores_locales_stado[2]));
        }
        if (message1.indexOf("qc") >= 0) {
            contactores_locales_stado[3] = 1;
            digitalWrite(q3, (contactores_locales_stado[3]));
            delay(100);
            contactores_locales_stado[3] = 0;
            digitalWrite(q3, (contactores_locales_stado[3]));

            if (digitalRead(pin_sensor_cortinas1) == 1) {
                notifyClients("qc0");
            } else if (digitalRead(pin_sensor_cortinas2) == 1) {
                notifyClients("qc1");
            } else {
                notifyClients("qc0");
            }
        }
    }
    if (message1.indexOf("qd") >= 0) {
        st_dif = !st_dif;
        notifyClients("qd" + String(st_dif));
        LoRa_receive = " ";
        lora_r = " ";
        fun = 3;
    }
}

for (int j = 0; j < n_dir_sect * 2; j++) {
    // Serial.print(j);
    if (message1.indexOf("q" + String(j + 10)) >= 0) {
        contactor_i2c_stado[j] = !contactor_i2c_stado[j];
    }
}

```

```

        notifyClients(("q" + String(j + 10) + "=" + String(contactor_i2c_stado[j]) +
"&"));
    }
}
    notifyClients("OK");    // oled_line[1] = message1;    // Serial.println(message1);
}
}

```

```

void onEvent(AsyncWebSocket* server, AsyncWebSocketClient* client,
AwsEventType type, void* arg, uint8_t* data, size_t len) {

```

```

    String str1 = "";

```

```

    switch (type) {

```

```

        case WS_EVT_CONNECT:

```

```

            for (int j = 0; j < n_dir_sect * 2; j++) {

```

```

                str1 = str1 + (("q" + String(j + 10) + "=" + String(contactor_i2c_stado[j]) + "&"));

```

```

            }

```

```

            notifyClients(

```

```

                "msect" + String(n_dir_sect) +

```

```

                "$ql" + String(contactores_locales_stado[1]) +

```

```

                "qv" + String(contactores_locales_stado[2]) +

```

```

                "qc" + String(pc) +

```

```

                "qd" + String(st_dif) +

```

```

                str1);

```

```

            break;

```

```

        case WS_EVT_DISCONNECT:

```

```

            // Serial.printf("WebSocket client #%u disconnected\n", client->id());

```

```

            break;

```

```

        case WS_EVT_DATA:

```

```

            handleWebSocketMessage(arg, data, len);

```

```

            break;

```

```

        case WS_EVT_PONG:

```

```

        case WS_EVT_ERROR:

```

```

            break;

```

```

    }

```

```

}

```

```

void process_data() {

```

```

    String datoLoRasend = "";

```

```

    if (((lora_r.substring(0, lora_r.indexOf("r"))).indexOf("i") >= 0)) {

```

```

        if (lora_r.substring(

```

```

            0,

```

```

            lora_r.indexOf("i")).toInt() == local_id) {

```

```

            if ((lora_r.substring(lora_r.indexOf("r") + 1, lora_r.indexOf("e"))).indexOf("g")

```

```

            >= 0) {

```

```

                if (lora_r.substring(

```

```

                    lora_r.indexOf("r") + 1,

```

```

                    lora_r.indexOf("g")).toInt() == ga_id) { // id_r = lora_r.substring(0,

```

```

(lora_r.substring(0, lora_r.indexOf("r"))).indexOf("i").toInt();
    LoRa_receive = lora_r.substring(
        lora_r.indexOf('f') - 1,
        lora_r.length());
    if (LoRa_receive.indexOf('f') >= 0) {
        fun = LoRa_receive.substring(
            0,
            LoRa_receive.indexOf('f')).toInt();
        LoRa_receive = LoRa_receive.substring(2, LoRa_receive.length());
    } else {
        fun = 0;
    }
}
}
}
}
lora_r = " ";
// Serial.println(LoRa_receive);
byte q_id, q_st;
switch (fun) {
    case 0: // no hace nada
        break;
    case 1: // envia valor del los sensores al gateway LoRa
        datoLoRasend = (
            String(ga_id) + "gr" +
            String(local_id) + "ie" +
            String(1) + "f" +
            datoI2CtoLora + //255gr16ie3f
            String(pW).c_str() + "w" +
            String(vr_ph).c_str() + "p" +
            String(pc).c_str() + "c"); // Serial.println(datoLoRasend);
        LoRa_sendMessage(datoLoRasend);
        fun = 0;
        break;
    case 2: // confirma que se ejecutaron las funciones de control //
String(q, BIN).toCharArray(bits, 4);
    //e2f5-15q0
    q_id = LoRa_receive.substring(0, LoRa_receive.indexOf("q")).toInt();
    q_st = LoRa_receive.substring(LoRa_receive.indexOf("q") + 1,
LoRa_receive.length()).toInt();
    if (q_id < 4) {
        if (q_id == 3) {
            contactores_locales_stado[3] = 1;
            digitalWrite(q3, (contactores_locales_stado[3]));
            delay(100);
            contactores_locales_stado[3] = 0;
            digitalWrite(q3, (contactores_locales_stado[3]));
        } else {
            contactores_locales_stado[q_id] = q_st;

```

```

    }
  } else {
    contactor_i2c_stado[q_id - 4] = q_st;
  }

  datoLoRasend = (
    String(ga_id) + "gr" +
    String(local_id) + "ie" +
    String(4) + "f");
  LoRa_sendMessage(datoLoRasend);
  fun = 0;
  break;
case 3:
  datoLoRasend = (
    String(ga_id) + "gr" + //255gr16ie3f16nrge2f
    String(local_id) + "ie" +
    String(3) + "f" +
    String(no_id) + "nrge2f2q" +
    String(st_dif));
  LoRa_sendMessage(datoLoRasend);
  delay(200);
  datoLoRasend = (
    String(ga_id) + "gr" + //255gr16ie3f16nrge2f
    String(local_id) + "ie" +
    String(3) + "f" +
    String(no_id) + "nrge2f1q" +
    String(st_dif));
  LoRa_sendMessage(datoLoRasend);
  delay(200);
  datoLoRasend = (
    String(ga_id) + "gr" + //255gr16ie3f16nrge2f
    String(local_id) + "ie" +
    String(3) + "f" +
    String(no_id) + "nrge2f2q" +
    String(st_dif));
  LoRa_sendMessage(datoLoRasend);
  delay(200);
  datoLoRasend = (
    String(ga_id) + "gr" + //255gr16ie3f16nrge2f
    String(local_id) + "ie" +
    String(3) + "f" +
    String(no_id) + "nrge2f1q" +
    String(st_dif));
  LoRa_sendMessage(datoLoRasend);
  delay(200);
  fun = 0;
  break;
default:
  break;

```

```

    }
    LoRa_receive = " ";
}

void p_wifi(void* pvParameters) {
    /* WiFi init */
    // Serial.println("WiFi init");
    WiFi.disconnect(true);
    WiFi.mode(WIFI_AP);
    WiFi.softAP(hostname.c_str(), "espinvlora");
    WiFi.softAPConfig(
        IPAddress(192, 168, 1, 1),
        IPAddress(192, 168, 1, 1),
        IPAddress(255, 255, 255, 240));
    /* WebSocket init */
    ws.onEvent(onEvent);
    server.addHandler(&ws);
    /* Server init */
    server.on("/", HTTP_GET, [](AsyncWebServerRequest * request) {
        request->send(SPIFFS, "/index.html", "text/html");
    });
    server.serveStatic("/", SPIFFS, "/");
    server.begin();
    // Serial.println("server started");
    vTaskDelay(1000);
    for (;;) {
        delay(200);
        ws.cleanupClients();
        process_data();
    }
}
// void p_data(void* pvParameters) {
// vTaskDelay(500);
// for (;;) {
// process_data();
// }
// }

void oled_print(int timei2c) {
    if ((now - timeaux_i2c_oled) > timei2c) {
        timeaux_i2c_oled = now;
        display.clearDisplay();
        display.setTextColor(WHITE);
        display.setTextSize(1);
        display.setRotation(1);
        for (int i = 0; i < 10; i++) {

```

```

        display.setCursor(0, (i * 10));
        display.print(oled_line[i]);
    }
    display.display();
}
}
void monitor_sectores_i2c(int timei2c) {
    if ((now - timeaux_i2c) > timei2c) {
        timeaux_i2c = now;
        datoI2Creceive = "";
        tmp_p = 0;
        hum_p = 0;
        hums_p = 0;
        for (int cont_1 = 0; cont_1 < n_dir_sect; cont_1++) {
            String datoI2Caux1 = "";
            Wire.requestFrom(dir_sect_i2c[cont_1], uint8_t(18));
            while (Wire.available()) { // Verifica que el puerto serial virtual recibe datos
                char c = Wire.read(); // Serial.println("int -> " + String(int(c)) + " char
-> " + c);
                if ((int(c) >= 32) && (int(c) <= 126)) {
                    datoI2Caux1 = datoI2Caux1 + c;
                }
            }
            //Serial.println(datoI2Caux1);
            datoI2Caux1 = datoI2Caux1 + char(33 + cont_1);
            byte pos1 = datoI2Caux1.indexOf('t');
            byte pos2 = datoI2Caux1.indexOf('h');
            byte pos3 = datoI2Caux1.indexOf('s');
            tmp_s[cont_1] = (datoI2Caux1.substring(0, pos1)).toFloat();
            hum_s[cont_1] = (datoI2Caux1.substring(pos1 + 1, pos2)).toFloat();
            hums_s[cont_1] = (datoI2Caux1.substring(pos2 + 1, pos3)).toFloat();
            tmp_p = tmp_p + tmp_s[cont_1];
            hum_p = hum_p + hum_s[cont_1];
            hums_p = hums_p + hums_s[cont_1];
            datoI2Creceive = datoI2Creceive + datoI2Caux1;
            datoI2CtoLora = datoI2Creceive;
        }
        tmp_p = tmp_p / n_dir_sect;
        hum_p = hum_p / n_dir_sect;
        hums_p = hums_p / n_dir_sect;
        oled_line[3] = "Tmp: " + String(int(tmp_p)) + " *C";
        oled_line[4] = "H.a: " + String(int(hum_p)) + " %";
        oled_line[5] = "H.s: " + String(int(hums_p)) + " %";
    }
}
void control_sectores_i2c() {
    for (int i = 0; i < n_dir_sect; i++) {
        if ((contactor_i2c_stado[i] != aux_c_s[i]) || (contactor_i2c_stado[i + n_dir_sect] !=
aux_c_s[i + n_dir_sect])) {

```

```

Wire.beginTransmission(dir_sect_i2c[i]);
Wire.write((
    String(contactor_i2c_stado[i]) + "a" +
    String(contactor_i2c_stado[i + n_dir_sect]) + "b").c_str());
Wire.endTransmission();
aux_c_s[i] = contactor_i2c_stado[i];
aux_c_s[i + n_dir_sect] = contactor_i2c_stado[i + n_dir_sect];
}
delay(50);
}

}

void analog_read_process(int timefilter) {
if ((now - timeaux_filtro) > timefilter) {
    timeaux_filtro = now;
    cont_f++;
    s_luz = s_luz + ((analogRead(pin_sensor_luz)));
    s_ph = s_ph + (analogRead(pin_sensor_ph)*0.75);
    s_luz = s_luz / cont_f;
    s_ph = s_ph / cont_f;

    if (pc){
oled_line[1] = "C: Abierto";

    }else{

oled_line[1] = "C: Cerrado";
    }
oled_line[6] = "Ph: " + String((vr_ph)) + "ph";
cont_f = 0;
s_luz = 0;
s_ph = 0;
// Serial.println(n_dir_sect);
} else {
    s_luz = s_luz + ((analogRead(pin_sensor_luz)));
    s_ph = s_ph + (analogRead(pin_sensor_ph)*0.75);
    cont_f++;
}
}

void procesos_invernadero() {
//      if      ((digitalRead(pin_sensor_cortinas1)      ==      1)      &&
(digitalRead(pin_sensor_cortinas2) == 0) ) {
//  pc = 1;
//  } else  if  ((digitalRead(pin_sensor_cortinas1)  ==  0)  &&
(digitalRead(pin_sensor_cortinas2) == 1) ) {
//  pc = 0;
//  }
}

```

```

    pc = digitalRead(pin_sensor_cortinas2);
}

void setup() {
  //initialize Serial Monitor
  //Serial.begin(115200);

  /* I2C init */
  pinMode(q0, OUTPUT);
  pinMode(q1, OUTPUT);
  pinMode(q2, OUTPUT);
  pinMode(q3, OUTPUT);
  pinMode(pin_sensor_cortinas1, INPUT_PULLDOWN);
  pinMode(pin_sensor_cortinas2, INPUT_PULLDOWN);
  digitalWrite(q0, LOW);
  digitalWrite(q1, LOW);
  digitalWrite(q2, LOW);
  digitalWrite(q3, LOW);

  // pinMode(OLED_RST, OUTPUT);
  // digitalWrite(OLED_RST, LOW);
  // delay(20);
  // digitalWrite(OLED_RST, HIGH);
  Wire.begin(SDA, SCL);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) {
    for (;;)
  }
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setRotation(1);
  display.setTextSize(1);
  display.display();

  hostname = "Inv. " + String(16 + 1);
  /* Lora init */
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(915E6)) {
    while (1);
  }

  LoRa.setSyncWord(syn_byte);
  LoRa.enableCrc();
  LoRa.onReceive(onReceive);
  LoRa.onTxDone(onTxDone);
  LoRa_rxMode();
  /* SPIFFS init */
  if (!SPIFFS.begin(true)) {

```

```

    while (1); // Serial.println("An error has occurred while mounting SPIFFS");
}
/* Core init */
xTaskCreatePinnedToCore(
  p_wifi, /* Task function. */
  "Task1", /* name of task. */
  4096, /* Stack size of task */
  NULL, /* parameter of the task */
  5, /* priority of the task */
  &Task1, /* Task handle to keep track of created task */
  1); /* pin task to core 0 */
delay(500);
for (byte i = 0; i < 16; i++) {
  Wire.beginTransaction(i);
  if (Wire.endTransmission() == 0) {
    //Serial.println(i);
    dir_sect_i2c[n_dir_sect] = i;
    n_dir_sect++;
  }
}
oled_line[0] = hostname;
oled_line[2]="Datos Pro.";
}

void loop() {
  // LoRa_sendMessage();
  now = millis();
  // n_dir_sect = 4;
  // delay(100);
  if (n_dir_sect > 0) {
    monitor_sector_i2c(100);
    control_sector_i2c();
  }
  analog_read_process(3000);
  // process_data();
  procesos_invernadero();

  oled_print(100);
  digitalWrite(q0, (contactores_locales_stado[0]));
  digitalWrite(q1, (contactores_locales_stado[1]));
  digitalWrite(q2, (contactores_locales_stado[2]));
  digitalWrite(q3, (contactores_locales_stado[3]));
  delay(100);
  contactores_locales_stado[3] = 0;
}

```

## ANEXO 19

### PROGRAMACIÓN NODO LORA BOMBA

```
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_ADS1X15.h>
// Pinout
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 23
#define DIO0 26
#define SDA 21
#define SCL 22
#define OLED_RST -1
#define q0 1 //web page q 1
#define q1 12 //web page q 12
#define q2 4 //web page q 4
#define q3 25 //web page q 25

// Constantes
#define LoRa_BAND 915E6
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Variables
boolean contactores_locales_stado[] = {
  // 32 33 34 35 inv // General
  0, 0, 0, 0, 0 // General
};

byte fun;
byte cont_f = 0, local_id = 16, ga_id = 255, inv_id = 16;
// byte q_id,q_st;
float tmp_s[16], hum_s[16], hums_s[16]; // i2c
float tmp_p, hum_p, hums_p; // i2c
unsigned long now;
unsigned long timeaux_filtro = 0;

String hostname = ""; //webserver
String oled_line[10];
String LoRa_receive = ""; // lora
```

```

String message1 = "";
String lora_r;

Adafruit_SSD1306 display = Adafruit_SSD1306(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire);
Adafruit_ADS1115 ads0, ads1;
TaskHandle_t Task1, Task2, Task3;

boolean st = 0;

void setup() {
  //Serial.begin(115200); // initialize serial
  pinMode(q0, OUTPUT);
  pinMode(q1, OUTPUT);
  pinMode(q2, OUTPUT);
  pinMode(q3, OUTPUT);
  digitalWrite(q0, LOW);
  digitalWrite(q1, LOW);
  digitalWrite(q2, LOW);
  digitalWrite(q3, LOW);
  hostname = "Nodo " + String(local_id);
  oled_line[0] = hostname;
  // /* Core init */

  xTaskCreatePinnedToCore(
    p_lora, /* Task function. */
    "Task2", /* name of task. */
    10000, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    &Task1, /* Task handle to keep track of created task */
    1); /* pin task to core 0 */
  delay(250);
  xTaskCreatePinnedToCore(
    p_i2c, /* Task function. */
    "Task3", /* name of task. */
    4096, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    &Task3, /* Task handle to keep track of created task */
    1); /* pin task to core 0 */
  delay(250);
}

void loop() {
  now = millis();
  if ((now - timeaux_filtro) > 800) {
    timeaux_filtro = now;
  }
}

```

```

cont_f++;
sv1 = sv1 + c_sv1;
sv2 = sv2 + c_sv2;
sI1 = sI1 + c_sI1;
sI2 = sI2 + c_sI2;
sv1 = sv1 / cont_f;
sv2 = sv2 / cont_f;
sI1 = sI1 / cont_f;
sI2 = sI2 / cont_f;
pW = (sv1 * sI1) + (sv2 * sI2);
oled_line[1] = ("pw: " + String(pW));
oled_line[2] = "I1: " + String(sI1 * 1000) + " mA";
oled_line[3] = "I2: " + String(sI2 * 1000) + " mA";
oled_line[4] = "V1: " + String(sv1 - 1.63) + " V";
oled_line[5] = "V2: " + String(sv2 - 1.63) + " V";
sv1 = 0;
sv2 = 0;
sI1 = 0;
sI2 = 0;
// Serial.println("promedio" + String(cont_f));
cont_f = 0;
} else {
sv1 = sv1 + c_sv1;
sv2 = sv2 + c_sv2;
sI1 = sI1 + c_sI1;
sI2 = sI2 + c_sI2;
cont_f++;
}
}
delay(10);
}

void p_i2c(void* pvParameters) {
/* I2C init */
Wire.begin(SDA, SCL);
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) {
//Serial.println("oled_fail");
for (;;)
}
if (!ads0.begin(0x48, &Wire)) {
//Serial.println("Failed to initialize ADS.");
while (1);
}
ads0.setGain(GAIN_ONE);
if (!ads1.begin(0x49, &Wire)) {
//Serial.println("Failed to initialize ADS.");
while (1);
}
ads1.setGain(GAIN_ONE);
display.clearDisplay();
}

```

```

display.setTextColor(WHITE);
display.display();
//Serial.println("oled_init");
for (;;) {
  // Serial.println("oled_run");
  c_sv1 = abs(ads1.readADC_Differential_0_1() * c_ads * 51.4 / 1000);
  delay(50);
  // float c_sv2 = sv2 + abs(ads1.readADC_Differential_2_3() * c_ads);
  c_sv2 = c_sv1;
  delay(50);
  c_sI1 = abs(ads0.readADC_Differential_0_1() * c_ads * ci30 / 1000);
  delay(50);
  c_sI2 = abs(ads0.readADC_Differential_2_3() * c_ads * ci100 / 1000);
  delay(50);
  oled_print();
  delay(50);
  contactores_locales_stado[1]      =      (contactores_locales_stado[2]      ||
contactores_locales_stado[3]      ||      contactores_locales_stado[4])      &&
contactores_locales_stado[1];
  digitalWrite(q2, (contactores_locales_stado[2]));
  digitalWrite(q3, (contactores_locales_stado[3]));
}
}

void p_lora(void* pvParameters) {
  for (;;) {
    SPI.begin(SCK, MISO, MOSI, SS);
    LoRa.setPins(SS, RST, DIO0);
    if (!LoRa.begin(915E6)) {
      while (1);
    }
    LoRa.setSyncWord(255);
    // LoRa.setSyncWord(ga_id);
    LoRa.enableCrc();
    LoRa_rxMode();
    for (long ml = 0; ml < 600; ml++) {

      String datoLoRasend = "";
      if (((lora_r.substring(0, lora_r.indexOf("r"))).indexOf("n") >= 0)) {
        if (lora_r.substring(
          0,
          lora_r.indexOf("n")).toInt() == local_id) {
          if ((lora_r.substring(lora_r.indexOf("r") + 1, lora_r.indexOf("e"))).indexOf("g")
          >= 0) {
            if (lora_r.substring(
              lora_r.indexOf("r") + 1,
              if (LoRa_receive.indexOf('f') >= 0) {
                fun = LoRa_receive.substring(
                  0,

```

```

        LoRa_receive.indexOf('f')).toInt();
        LoRa_receive = LoRa_receive.substring(2, LoRa_receive.length());
    } else {
        fun = 0;
    }
}
}
}
}
lora_r = " ";
byte q_id, q_st;
switch (fun) {
case 0: default: // no hace nada
    break;
case 1: // envia valor del los sensores al gateway LoRa
    datoLoRasend = (
        String(ga_id) + "gr" +
        String(pW).c_str() + "W"); // Serial.println(datoLoRasend);
    LoRa_sendMessage(datoLoRasend);
    fun = 0;
    break;
case 2: // confirma que se ejecutaron las funciones de control // String(q,
BIN).toCharArray(bits, 4);
    if (q_id == 1) {
        contactores_locales_stado[1] = (contactores_locales_stado[2] ||
    } else {
        contactores_locales_stado[q_id] = q_st;
    }
    datoLoRasend = (
        String(ga_id) + "gr" +
        String(local_id) + "ne" +
        String(4) + "f"); // Serial.println(datoLoRasend);
    LoRa_sendMessage(datoLoRasend);
    fun = 0;
    break;
case 3:
    datoLoRasend = (
        String(ga_id) + "gr" +
        String(local_id) + "ne" +
        String(3) + "f" +
        String(inv_id) + "ir" +
        String(contactores_locales_stado[1]) + "d");
    LoRa_sendMessage(datoLoRasend);
    break;
case 5:
    ESP.restart();
    break;
}
// Serial.println(LoRa_receive);
LoRa_receive = " ";
vTaskDelay(50);

```

```

    delay(50);
}

//Serial.println("                for out");
}
}

void LoRa_rxMode() {
    LoRa.enableInvertIQ();    // active invert I and Q signals
    LoRa.receive();          // set receive mode
}
void LoRa_txMode() {
    LoRa.idle();             // set standby mode
    LoRa.disableInvertIQ();  // normal mode
}
void LoRa_sendMessage(String message) {
    LoRa_txMode();           // set tx mode
    LoRa.endPacket(true);    // finish packet and send it
    Serial.println(message); //
}
void onReceive(int packetSize) {
    String message = "";
    while (LoRa.available()) {
        message = LoRa.readString();
        st = (message).toInt();
        // st = 1;
        lora_r = message;
        // Serial.println(message);
    }

    // Serial.print("Node Receive: ");
    // Serial.println(message);
}
void onTxDone() {
    LoRa_rxMode();
}

void oled_print() {
    display.clearDisplay();
    for (int i = 0; i < 8; i++) {
        display.setCursor(0, (i * 9));
        display.setTextSize(1);
        display.print(oled_line[i]);
    }
    display.display();
}
}

```

## ANEXO 20

### PROGRAMACIÓN DEL GATEWAY

```
#include <SPI.h>           // include libraries
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <EEPROM.h>
#include <WiFi.h>
#include <SPIFFS.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <PubSubClient.h>

// Pinout
#define LoRa_SCK 5
#define LoRa_MISO 19
#define LoRa_MOSI 27
#define LoRa_DIO0 26
#define SDA 21
#define SCL 22

// Constantes
#define LoRa_BAND 915E6
#define d_time 400
#define EEPROM_SIZE 63
#define OLED_ADD 0x3C
#define time_mqtt 1000
const char* ssid = "Speedy Alejandro"; //Speedy Alejandro   NETLIFE-ULISES
const char* password = "$18031996$"; // $18031996$       11221725
String mqtt_ip = "192.168.1.8";

// Variables
bool newRequest = false, ii = 0, st_lora = 1;
byte dir_i2c[15], fun = 2;
int local_id = 0, n_nodos[] = { 16,16 }, id_r;
int counter = 0, n_dir = 0, s_luz = 0, s_ph = 0, s_cor = 0;
long timeaux1 = 0, timeaux2 = 0, time_c = 0;
float tmp_s[16], hum_s[16], hums_s[16], tmp_p, hum_p, hums_p;

TaskHandle_t Task1;

Adafruit_SSD1306 display = Adafruit_SSD1306(SCREEN_WIDTH,
```

```

SCREEN_HEIGHT, &Wire);
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");
WiFiClient ESP_LoRa_Gateway;
PubSubClient client(ESP_LoRa_Gateway);

void EEPROM_init() {
  EEPROM.begin(EEPROM_SIZE);
}

void WiFi_init() {
  // hostname = "Gateway LoRa";
  // ssid_ap=ssid_ap + hostname.c_str();
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
}

void mqtt_init() {
  mqtt_ip = (EEPROM.readString(2));
  Serial.print(" ip -> ");
  client.setServer((mqtt_ip.c_str()), EEPROM.readInt(18));
  client.setCallback(callback);
}

void SPIFFS_init() {
  if (!SPIFFS.begin(true)) {
    Serial.println("An error has occurred while mounting SPIFFS");
  } else {
    Serial.println("SPIFFS mounted successfully");
  }
}

void WebSocket_init() {
  ws.onEvent(onEvent);
  server.addHandler(&ws);
}

void server_init() {
  server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) {
  });
  server.serveStatic("/", SPIFFS, "/");
  server.begin();
}

```

```

void i2c_init() {
  Wire.begin(SDA, SCL);
  Serial.println("i2c_init");
}

void oled_init() {
  pinMode(OLED_RST, OUTPUT);
  digitalWrite(OLED_RST, 1);
  delay(1);
  Serial.println(F("SSD1306 allocation failed"));
  for (;;) // Don't proceed, loop forever
  }
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.display();
}

void lora_init() {
  SPI.begin(LoRa_SCK, LoRa_MISO, LoRa_MOSI, LoRa_SS);
  Serial.println("LoRa init failed. Check your connections.");
  while (true);
}
LoRa.onReceive(onReceive);
LoRa.onTxDone(onTxDone);
LoRa_rxMode();
}

void core_init() {
  xTaskCreatePinnedToCore(
    conexiones_web, /* Task function. */
    "Task1", /* name of task. */
    &Task1, /* Task handle to keep track of created task */
    0); /* pin task to core 0 */
  delay(500);
}

void setup() {
  Serial.begin(115200); // initialize serial
  while (!Serial);
  EEPROM_init();
  WiFi_init();
  mqtt_init();
  SPIFFS_init();
  WebSocket_init();
  server_init();
  i2c_init();
  oled_init();
  core_init();
}

```

```

// lora_init();
oled_line[0] = hostname;
oled_print();
}

void loop() {
  long now = millis();
  oled_print();
  if (st_lora == 1) {
    for (int i = 0; i < 2;i++) {
      fun = 1;
      if (i % 2 == 0) {
        datoLoRasned = String(16) + "ir" +
          String(fun) + "f";
      } else {
        datoLoRasned = String(16) + "nr" + // String(2) + "ge" +
          String(EEPROM.read(0)) + "ge" +
        }
      LoRa_sendMessage(datoLoRasned); // send a message
      delay(400);
    }
  }
}

void LoRa_rxMode() {
  LoRa.disableInvertIQ();
  LoRa.receive();
}

void LoRa_txMode() {
  LoRa.idle();
  LoRa.enableInvertIQ();
}

void LoRa_sendMessage(String message) {
  LoRa_txMode();
  // Serial.print("Send Message! ---> ");
  Serial.println(message);
  delay(100);
}

void onReceive(int packetSize) {
  // String message = "";
  while (LoRa.available()) {
    String LoRaData = LoRa.readString();
    12gr123ie1fqq
    if (((LoRaData.substring(0, LoRaData.indexOf("r"))).indexOf("g") >= 0)) {
      if (LoRaData.substring(
        0,

```

```

        LoRaData.indexOf("g")).toInt() == local_id) {
        if ((LoRaData.substring(LoRaData.indexOf("r") + 1,
LoRaData.indexOf("e"))).indexOf("i") >= 0) {
            LoRaData.indexOf('f') - 1,
            LoRaData.length());
        } else if ((LoRaData.substring(LoRaData.indexOf("r") + 1,
LoRaData.indexOf("e"))).indexOf("n") >= 0) {
            LoRaData.indexOf('f') - 1,
            LoRaData.length());
        }
    }
}
st_lora = 1;
}
}

```

```

void onTxDone() {
    LoRa_rxMode();
}

```

```

void oled_print() {
    display.clearDisplay();
    for (int i = 0; i < 8; i++) {

        display.print(oled_line[i]);
    }
    display.display();
    delay(1000);
}

```

```

void conexiones_web(void* pvParameters) {
    lora_init();
    for (;;) {
        long now_c = millis();
        if ((WiFi.status() != WL_CONNECTED)) { // reconectar a la red WiFi
            oled_line[1] = "...";
            WiFi.reconnect();
        } else {
            oled_line[1] = WiFi.localIP().toString().c_str();
            oled_line[2] = EEPROM.readString(2);
            if (!client.connected()) {
                oled_line[4] = "Desconectado";
                reconnect();
            } else {
                oled_line[4] = "Conectado";
                // Serial.println(" D conectado ");
                if ((now_c - time_c) > time_mqtt) {
                    time_c = now_c;
                }
            }
        }
    }
}

```

```

        // true) {
        (LoRa_receive_n.substring(0, LoRa_receive_i.indexOf('f')).toInt() ==
1)) {
        int div = 0;
        Serial.println(LoRa_receive_i);
        Serial.println(LoRa_receive_n);
        for (int j = 0; LoRa_receive_i.indexOf(char(j + 33)) >= 0; j++) {
            String str_aux = LoRa_receive_i.substring(0,
LoRa_receive_i.indexOf(char(j + 33)));
            byte pos1 = str_aux.indexOf('t');
            byte pos2 = str_aux.indexOf('h');
            byte pos3 = str_aux.indexOf('s');
            tmp_p = tmp_s[j] + tmp_p;
            hum_p = hum_s[j] + hum_p;
            hums_p = hums_s[j] + hum_p;
            LoRa_receive_i =
LoRa_receive_i.substring(LoRa_receive_i.indexOf(char(j + 33)) + 1,
LoRa_receive_i.length());
            // Serial.println(LoRa_receive_i);
            div = j + 1;
        }
        tmp_p = tmp_p / div;
        hum_p = hum_p / div;
        hums_p = hum_p / div;

        //lora a mqtt
        client.publish(("Invernadero_" + String(EEPROM.read(0)) +
"/datos").c_str(), (
        "" + LoRa_receive_i.substring(0, LoRa_receive_i.indexOf('w')) +
        "," +
        "" + LoRa_receive_i.substring(LoRa_receive_i.indexOf('w') + 1,
LoRa_receive_i.indexOf('p')) + "," +
LoRa_receive_i.indexOf('c') + "," +
        "" + String(tmp_p) + "," + String(tmp_s[0]) + "," +
String(tmp_s[1]) + "," + String(tmp_s[2]) + "," + String(tmp_s[3]) + "," +
        "" + String(hum_p) + "," + String(hum_s[0]) + "," +
String(hums_s[1]) + "," + String(hums_s[2]) + "," + String(hums_s[3]) + "," +
LoRa_receive_n.indexOf('W')) + ""
        ).c_str()
        );
    }
}
}
}
client.loop();
}
ws.cleanupClients();
delay(500);
}
}
}

```

```

void notifyClients(String state) {
    ws.textAll(state);
}

void handleWebSocketMessage(void* arg, uint8_t* data, size_t len) {
    AwsFrameInfo* info = (AwsFrameInfo*)arg;
    if (info->final && info->index == 0 && info->len == len && info->opcode ==
WS_TEXT) {
        data[len] = 0;
        message1 = (char*)data;
        if (message1.indexOf("id") >= 0) {
        }
        if (message1.indexOf("ipv4") >= 0) { //
            EEPROM.writeString(2, message1.substring(4, message1.indexOf("port")));
            // Serial.println(message1.substring(message1.indexOf("port") + 4,
message1.length()));
            EEPROM.writeInt(18, message1.substring(message1.indexOf("port") + 4,
message1.length()).toInt());
            EEPROM.commit();
            notifyClients("ip" + EEPROM.readString(2) + "port" +
String(EEPROM.readInt(18)) + "=");
        }
        notifyClients("OK");
    }
}

void onEvent(AsyncWebSocket* server, AsyncWebSocketClient* client,
AwsEventType type, void* arg, uint8_t* data, size_t len) {
    switch (type) {
        case WS_EVT_CONNECT:
            // Serial.printf("WebSocket client #%u connected from %s\n", client->id(),
client->remoteIP().toString().c_str());
            notifyClients("ip" + EEPROM.readString(2) + "port" +
String(EEPROM.readInt(18)) + "=");
            notifyClients("id" + String(EEPROM.read(0)) + "#");
            break;
        case WS_EVT_DISCONNECT:
            // Serial.printf("WebSocket client #%u disconnected\n", client->id());
            break;
        case WS_EVT_DATA:
            handleWebSocketMessage(arg, data, len);
            break;
        case WS_EVT_PONG:
        case WS_EVT_ERROR:
            break;
    }
}

```

```

void callback(char* topic, byte* message, unsigned int length) {
    st_lora = 0;
    Serial.print("Message arrived on topic: ");
    Serial.print(topic);
    Serial.print(". Message: ");
    // mqtt a lora
    String messageTemp;
    for (int i = 0; i < length; i++) {
        // Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
    Serial.print(messageTemp.toInt());

    dato_e = ("q!" + messageTemp + String(char(34)) + "0#0$8%0&");
    LoRa_sendMessage(
        String(16) + "ir" +
        String(EEPROM.read(0)) + "ge" +
        String(2) + "f" + dato_e);
    delay(100);
    st_lora = 1;
}

void reconnect() {
    while (!client.connected()) {
        if (client.connect("ESP_LoRa_Gateway")) {
            client.subscribe(("Invernadero_" + String(EEPROM.read(0)) +
"/control").c_str());
        } else { // Serial.print("failed, rc="); // Serial.print(client.state()); //
Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}
}

```