



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**DESARROLLO DE UN SISTEMA IOT PARA EL MONITOREO REMOTO DE
VARIABLES EN UN HUERTO AGRÍCOLA**

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

AUTOR: BRYAN GEOVANNY PEÑARANDA ARCENTALES

TUTOR: ING. JUAN DIEGO JARA SALTOS, MgT.

Cuenca – Ecuador

2024

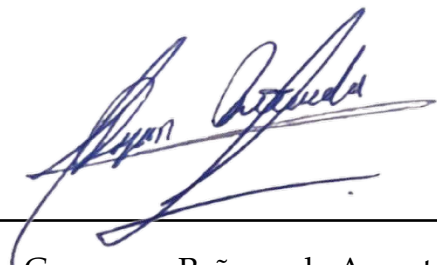
**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Bryan Geovanny Peñaranda Arcentales con documento de identificación N° 0106881980; manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 18 de marzo del 2024

Atentamente,



Bryan Geovanny Peñaranda Arcentales

0106881980

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Bryan Geovanny Peñaranda Arcentales con documento de identificación N° 0106881980, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy el autor del Proyecto Técnico: “Desarrollo de un sistema IoT para el monitoreo remoto de variables en un huerto agrícola” el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 18 de marzo del 2024

Atentamente,



Bryan Geovanny Peñaranda Arcentales

0106881980

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Diego Jara Saltos con documento de identificación N° 0103543658, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN SISTEMA IOT PARA EL MONITOREO REMOTO DE VARIABLES EN UN HUERTO AGRÍCOLA, realizado por Bryan Geovanny Peñaranda Arcentales con documento de identificación N° 0106881980, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 18 de marzo del 2024

Atentamente,



Ing. Juan Diego Jara Saltos, MgT

0103543658

AGRADECIMIENTOS

Agradecimientos de Bryan Peñaranda

Quisiera expresar mi profundo agradecimiento a mi familia, cuyo apoyo incondicional ha sido mi mayor fortaleza. Gracias por creer en mí y por ser mi constante fuente de inspiración.

También quiero agradecer al Ing. Juan Jara, tutor de este trabajo de titulación, quien ha sido más que un guía académico. Su orientación y sabios consejos han sido fundamentales en la culminación de este proyecto. Gracias por compartir su experiencia y conocimientos de manera generosa.

Agradezco también a mis compañeros de clase, quienes han compartido este viaje conmigo, brindándome su amistad y colaboración en cada desafío académico. Este logro no habría sido posible sin el apoyo de todas las personas que forman parte de mi vida, y por eso les dedico mi más sincero agradecimiento.

DEDICATORIAS

Dedicatoria de Bryan Peñaranda

A mis padres, fuente inagotable de apoyo, amor y sabiduría. A mi familia, por su constante aliento y comprensión en cada etapa de este desafiante camino académico. A mis amigos, cuyas risas y palabras de aliento han iluminado mis días. A mis profesores, por su inspiración, conocimiento y guía a lo largo de esta travesía educativa. A todos aquellos que, de una u otra manera, han contribuido a mi formación académica y personal, dedicándoles con gratitud el fruto de este esfuerzo. Este logro no solo es mío, sino de todos aquellos que han creído en mí y han sido parte fundamental de mi camino hacia el conocimiento.

Índice general

Agradecimientos	I
Dedicatorias	II
Índice General	III
Índice de figuras	X
Índice de tablas	XI
Resumen	XII
Abstract	XIII
Antecedentes	1
Justificación	4
Objetivos	6
Objetivo General	6
Objetivos Específicos	6
Introducción	7
1. Fundamentación Teórica	9
1.1. Agricultura 4.0	9
1.1.1. Tecnologías utilizadas en la Agricultura 4.0	10
1.1.2. Beneficios de la Agricultura 4.0	10

1.2.	Huerto agrícola casero	11
1.2.1.	Requerimientos ambientales del cultivo	12
1.2.2.	Parámetros climáticos	12
1.2.3.	Parámetros edafoclimáticos	14
1.3.	Internet de las Cosas (IoT)	15
1.3.1.	Propiedades del IoT	15
1.3.2.	Arquitectura del IoT	16
1.3.3.	Tecnologías para IoT	16
1.4.	IoT en la Agricultura	17
1.4.1.	Aplicaciones en Agricultura	17
1.4.2.	Beneficios en la Agricultura	18
1.5.	Tecnología LoRa	19
1.5.1.	LoRa	19
1.5.2.	LoRaWAN	21
1.5.3.	Topologías	22
1.5.4.	Modelo OSI	24
1.5.5.	LoRa en la agricultura	24
1.6.	MQTT	25
1.7.	Node-RED	26
1.8.	InfluxDB	26
1.9.	Grafana	27
2.	Marco Metodológico	28
2.1.	Esquema de bloques del sistema	28
2.2.	Componentes electrónicos	29
2.2.1.	Heltec Wi-Fi LoRa 32 (V2)	29
2.2.2.	Sensor de Temperatura Ambiente DHT22 AM2302	31
2.2.3.	Sensor de Humedad del Suelo HD-38	31
2.2.4.	Sensor de Radiación UV ML8511	32
2.2.5.	Elevador de voltaje MT3608	33
2.2.6.	Cargador Baterías Litio TP4056	34

2.2.7.	Batería Recargable 18650	35
2.2.8.	Panel Fotovoltaico	36
2.3.	Huerto agrícola	37
2.4.	Diagrama de comunicaciones	39
2.4.1.	Diagrama de comunicación LoRa	39
2.4.2.	Comunicación del ESP32 LoRa Heltec V2 hacia los sensores	39
2.4.3.	Suministro energético mediante energía solar	40
2.5.	Placa PCB	40
2.6.	Diseño y Fabricación de protección para dispositivos	41
3.	Implementación y Análisis de Resultados	43
3.1.	Diseño del dispositivo electrónico	43
3.1.1.	Interconexión de los componentes mediante una placa de pruebas	43
3.1.2.	Diagrama de flujo del dispositivo transmisor	46
3.1.3.	Programación del sensor DHT22	47
3.1.4.	Programación del sensor HD-38	47
3.1.5.	Programación del sensor ML85-11	48
3.2.	Comunicación inalámbrica con tecnología LoRa	48
3.2.1.	Selección de la topología de comunicación	48
3.2.2.	Programación del dispositivo transmisor	49
3.2.3.	Diagrama de flujo del ESP32 Heltec LoRa (Receptor)	51
3.2.4.	Programación del dispositivo receptor como Gateway LoRa	52
3.2.5.	Envío de datos	57
3.2.6.	Recepción de datos	57
3.3.	Plataforma de visualización	58
3.3.1.	Habilitación de los puertos de comunicación	58
3.3.2.	Configuración de Node-RED y Broker MQTT	60
3.3.3.	Conexión al Broker MQTT	60
3.3.4.	Configuración de una Dashboard en Node-RED	62
3.3.5.	Comunicación con InfluxDB	64
3.3.6.	Implementación de una interfaz gráfica en Grafana	70

3.4. Implementación y validación del sistema	73
3.4.1. Elaboración y montaje del circuito impreso	73
3.4.2. Diseño 3D	75
3.4.3. Impresión 3D	76
3.4.4. Montaje de las estructura del dispositivo transmisor y receptor	78
3.4.5. Aplicación en el huerto agrícola	80
3.4.6. Recepción y almacenamiento de información de cada variable climática	81
4. Conclusiones y Recomendaciones	82
4.1. Conclusiones	82
4.2. Recomendaciones	83
Glosario	85
Referencias	89
Apéndices	90

Índice de figuras

1.1. Huerto agrícola casero [Fuente: Autor]	11
1.2. Arquitectura de una red IoT [15]	16
1.3. Monitoreo de variables [3]	18
1.4. Técnica de modulación [16]	19
1.5. Frecuencias LoRa [20]	20
1.6. Arquitectura LoRaWAN [21]	21
1.7. Conexión punto a punto [25]	23
1.8. Conexión multipunto [25]	23
1.9. Modelo OSI [27]	24
1.10. Arquitectura de MQTT [29]	25
1.11. Node-RED [31]	26
1.12. InfluxDB [32]	27
1.13. Grafana [33]	27
2.1. Esquema visual[Fuente: Autor]	29
2.2. Módulo WiFi LoRa 32 V3 SX1262 de 915 MHz ESP32 Bluetooth OLED [35] .	30
2.3. Sensor de Temperatura Ambiente (DHT21 AM2302) [37]	31
2.4. Sensor HD-38 [39]	32
2.5. Sensor ML8511 [41]	33
2.6. Módulo DC-DC Elevador Step Up regulable MT3608 2A [43]	34
2.7. Módulo TP4056 [45]	35
2.8. Batería 18650 [47]	36
2.9. Panel solar [48]	37

2.10. Localización del huerto agrícola [Fuente: Google Maps]	38
2.11. Huerto agrícola casero [Fuente: Autor]	38
2.12. Sistema de comunicación inalámbrica de la red [Fuente: Autor]	39
2.13. Comunicación del ESP32 LoRa Heltec V2 hacia los sensores [Fuente: Autor]	40
2.14. Esquema de suministro eléctrico mediante energía solar [Fuente: Autor] . .	40
2.15. Vista 2D de la placa [Fuente: Autor]	41
2.16. Vista 3D de la placa [Fuente: Autor]	41
2.17. Estructura para el dispositivo transmisor [Fuente: Autor]	42
2.18. Estructura para el receptor [Fuente: Autor]	42
3.1. Identificación de puertos del microcontrolador [35]	43
3.2. Esquema de conexiones de los sensores con el módulo Heltec WiFi LoRa 32(V2) [Fuente: Autor]	44
3.3. Conexión de dispositivos [Fuente: Autor]	44
3.4. Lógica de programación del dispositivo transmisor [Fuente: Autor]	46
3.5. Librería del sensor DHT [Fuente: IDE Arduino]	47
3.6. Medición de la temperatura y humedad [Fuente: IDE Arduino]	47
3.7. Medición del porcentaje de humedad del suelo [Fuente: IDE Arduino]	47
3.8. Medición de la radiación ultravioleta (UV) [Fuente: IDE Arduino]	48
3.9. Topología P2P [24]	48
3.10. Configuraciones básicas de comunicación del dispositivo transmisor [Fuente: IDE Arduino]	49
3.11. Inicialización de la comunicación LoRa y Pantalla OLED [Fuente: IDE Arduino]	50
3.12. Palabra de sincronización [Fuente: IDE Arduino]	50
3.13. Construcción y envío de datos de los sensores [Fuente: IDE Arduino]	50
3.14. Lógica de programación del dispositivo receptor [Fuente: Autor]	51
3.15. Configuraciones básicas de comunicación del dispositivo receptor [Fuente: IDE Arduino]	52
3.16. Conexión WiFi y cliente MQTT [Fuente: IDE Arduino]	52
3.17. Palabra de sincronización (Receptor) [Fuente: IDE Arduino]	53

3.18. Inicialización y configuración del módulo LoRa [Fuente: IDE Arduino] . . .	53
3.19. Parametrización de conexión WiFi y MQTT [Fuente: IDE Arduino]	54
3.20. Procesamiento de datos desde paquetes LoRa y publicación en MQTT [Fuente: IDE Arduino]	54
3.21. Gestión continua de conexión y mensajes MQTT [Fuente: IDE Arduino] . .	55
3.22. Publicación eficiente de datos en tópicos específicos [Fuente: IDE Arduino]	55
3.23. Configuración y conexión a la red WiFi [Fuente: IDE Arduino]	56
3.24. Reconexión al servidor [Fuente: IDE Arduino]	56
3.25. Envío de información de la variables [Fuente: Autor]	57
3.26. Modulo receptor [Fuente: Autor]	57
3.27. Recepción y publicación de datos [Fuente: IDE Arduino]	58
3.28. Configuración de los puertos de entrada [Fuente: Autor]	59
3.29. Configuración de los puertos de salida [Fuente: Autor]	59
3.30. Inicialización de Node-RED [Fuente: Autor]	60
3.31. Conexión de Node-RED al broker MQTT [Fuente: Autor]	60
3.32. Nodos de publicación MQTT [Fuente: Node-RED]	61
3.33. Establecimiento de nodos MQTT [Fuente: Node-RED]	61
3.34. Configuración del servidor y puerto [Fuente: Node-RED]	62
3.35. Configuración los nodos de la interfaz gráfica [Fuente: Node-RED]	62
3.36. Configuración de la interfaz gráfica [Fuente: Node-RED]	63
3.37. Diseño de la interfaz para el sistema de supervisión [Fuente: Node-RED] . .	63
3.38. Inicializacion de InfluxDB [Fuente: Autor]	64
3.39. Puerto de enlace de InfluxDB [Fuente: Autor]	64
3.40. Menu de InfluxDB [Fuente: InfluxDB]	65
3.41. Organización en InfluxDB [Fuente: InfluxDB]	65
3.42. Creación del Bucket [Fuente: InfluxDB]	66
3.43. Ajustes del nodo InfluxDB [Fuente: InfluxDB]	66
3.44. Propiedades del nodo InfluxDB [Fuente: Node-RED]	67
3.45. Propiedades del servidor InfluxDB [Fuente: InfluxDB]	67
3.46. Registro de datos de la temperatura ambiente [Fuente: InfluxDB]	68
3.47. Registro de datos de la humedad ambiente [Fuente: InfluxDB]	68

3.48. Registro de datos de la radiación solar (UV) [Fuente: InfluxDB]	69
3.49. Registros de humedad del suelo [Fuente: InfluxDB]	69
3.50. Menu principal de Grafana [Fuente: Grafana]	70
3.51. Conexión de InfluxDB [Fuente: Grafana]	70
3.52. Integración HTTP entre InfluxDB y Grafana [Fuente: Grafana]	71
3.53. Configuración de la organización y bucket de InfluxDB en Grafana [Fuente: Grafana]	71
3.54. Desarrollo del gráfico indicador en Grafana [Fuente: Grafana]	72
3.55. Gráfico lineal en Grafana [Fuente: Grafana]	72
3.56. Plataforma de visualización en Grafana [Fuente: Grafana]	73
3.57. Elaboración de la placa PCB [Fuente: Autor]	74
3.58. Montaje de la placa PCB [Fuente: Autor]	74
3.59. Dimensiones (mm) de la base [Fuente: Fusion360]	75
3.60. Dimensión (mm) de la altura [Fuente: Fusion360]	75
3.61. Dimensiones (mm) de la base [Fuente: Fusion360]	76
3.62. Dimensión (mm) de la altura [Fuente: Fusion360]	76
3.63. Modelos 3D de las estructuras [Fuente: Autor]	77
3.64. Impresión en 3D de las estructuras [Fuente: Autor]	77
3.65. Visualización interna del dispositivo transmisor [Fuente: Autor]	78
3.66. Montaje del dispositivo transmisor [Fuente: Autor]	78
3.67. Visualización interna del dispositivo receptor [Fuente: Autor]	79
3.68. Montaje del dispositivo receptor [Fuente: Autor]	79
3.69. Montaje del dispositivo transmisor en el huerto agrícola [Fuente: Autor] . . .	80
3.70. Colocación del sensor en la tierra [Fuente: Autor]	80
3.71. Historial ambiental de los últimos 30 días [Fuente: Autor]	81

Índice de tablas

- 1.1. Rangos de temperatura e impacto en cultivos [11] 12
- 1.2. Rango óptimo de temperatura [11] 13
- 1.3. Índice de radiación ultravioleta (UV) y su impacto en cultivos [12] 13
- 1.4. Rango óptimo de radiación UV [12] 14
- 1.5. Rangos de humedad del suelo y su impacto en cultivos [13] 14
- 1.6. Rango óptimo de humedad del suelo [13] 15
- 1.7. Atributos de los principales [17] 17

- 3.1. Corriente de consumo de los componentes electrónicos [Fuente: Autor] 45

Resumen

El Internet de las cosas (IoT) se combina con la comunicación inalámbrica LoRa para crear un mecanismo de monitoreo de parámetros ambientales. En el cual, LoRa permite la comunicación eficiente de datos a grandes distancias, en cambio los sensores IoT recopilan información con precisión varios parámetros ambientales como la temperatura ambiente, radiación UV y humedad del suelo. Estos datos son transmitidos a través de la red LoRa y enviados a una plataforma central para su análisis y toma de decisiones informada en tiempo real, facilitando la gestión ambiental efectiva. Para el monitoreo y la recopilación de parámetros ambientales en un huerto agrícola utilizando esta tecnología representa un avance significativo en la detección ambiental especialmente en el área.

Utilizando la conectividad IoT, el sistema recopila datos de diferentes sensores colocados estratégicamente para capturar diversos parámetros ambientales del área mediante el microcontrolador, donde se transmitirá a una estación mediante una topología punto a punto con tecnología LoRa para posterior ser visualizadas en una interfaz gráfica permitiendo un monitoreo y análisis eficiente en tiempo real, se procesa la información recibida hacia una plataforma en la nube.

Además, el sistema basado en IoT permite la personalización y escalabilidad, lo que permite la integración con otros sistemas de monitoreo ambiental, facilitando una comprensión de los patrones climáticos y su impacto en un huerto agrícola, para tomar decisiones informadas para mitigar el cambio climático y desarrollar estrategias sostenibles basadas en datos climáticos almacenados.

Palabras clave: sensores, IoT, monitoreo, análisis, LoRa

Abstract

The Internet of Things (IoT) is combined with LoRa technology to create a system for monitoring environmental parameters. LoRa enables efficient data communication over long distances and low power consumption, while IoT sensors accurately collect information on various environmental parameters such as ambient temperature, UV radiation and soil moisture. This data is transmitted through the LoRa network and sent to a central platform for analysis and informed decision-making in real time, facilitating effective environmental management. For the monitoring and collection of environmental parameters in an agricultural garden using this technology represents a significant advance in environmental sensing especially in the area.

Using IoT connectivity, the system collects data from different sensors strategically placed to capture various environmental parameters of the area via the microcontroller, where it will be transmitted to a station through a point-to-point topology with LoRa technology and then displayed on a graphical interface allowing efficient real-time monitoring and analysis, processing the received data and storing it in a cloud-based platform.

In addition, the IoT-based system allows for customisation and scalability, enabling integration with other environmental monitoring systems, facilitating an understanding of weather patterns and their impact on the agricultural garden, to make informed decisions to mitigate climate change and develop sustainable strategies based on reliable and comprehensive climate data.

Keywords: sensors, IoT, monitoring, analysis, LoRa

Antecedentes

Según el informe *“Agricultura 4.0: El futuro de la tecnología agrícola”* prevé un aumento del 33% en la población mundial para 2050, alcanzando los 10 millones de personas. Debido a esto, el consumo de alimentos será un 50% mayor que lo que fue la producción agrícola en 2013 [1]. Al mismo tiempo, según un informe de la (FAO) publicado en 2019, se proyecta que 931 millones de toneladas de recursos alimenticios en todo el mundo se desperdician, de las cuales 118 millones de toneladas se pierden en la cadena de distribución y afectan a todos, desde los agricultores hasta los pequeños agricultores [2]. Esto equivale a una escasez mundial de alimentos del 14%, lo cual es una estadística alarmante dado que el 8,76% de la población mundial sufre hambre.

Por lo cual, es esencial avanzar en la tecnología agrícola para abordar este desafío. Esto implica brindar a los agricultores herramientas y tecnologías que les permitan producir más alimentos y reducir las pérdidas. Existen problemas con la agricultura tanto a nivel mundial como en el Ecuador, incluida la falta de asistencia técnica y una tecnificación inadecuada de los cultivos.

En la última década, los sistemas de monitoreo basado en IoT ha ganado un amplio reconocimiento en la agricultura de precisión y medio ambiente, en estos escenarios se utiliza una variedad de sensores para recopilar información de parámetros ambientales, que luego son almacenados, procesados y presentados mediante plataformas IoT para su posterior análisis. En el ámbito agrícola, se observa una tendencia emergente que es la agricultura urbana o agricultura en el hogar, que buscan la calidad de sus cosechas y alimentación, especialmente en zonas urbanas [3].

Caso particular en la provincia del Azuay específicamente en el cantón Gualaceo que pertenece a la región andina del Ecuador, el sector agrícola desempeña un papel

importante tanto en la gastronomía y economía local. Sin embargo, en la comunidad de Chicaguiña existe la problemática de que no cuentan con la implementación de sistemas IoT para monitorear y optimizar el cultivo en sus huertos, por lo cual puede surgir desafíos significativos que afectarían a los agricultores por la falta de tecnificación de los parámetros ambientales del huerto, también podría complicar la toma de decisiones informadas, en donde los agricultores podrían tener dificultades para planificar cuando sembrar, cosechar o aplicar fertilizantes de manera eficiente. La topografía variada de la región andina presenta un desafío en términos de variabilidad en el clima ambiental y las propiedades del suelo, donde es importante monitorear el porcentaje de humedad del suelo, la temperatura y otras medidas en tiempo real, sin esto los agricultores podrían enfrentar ineficiencias de riego, lo que resultaría en el desperdicio de agua y en la falta de nutrición adecuada en los cultivos, esto podría dar lugar a una pérdida general de eficiencia en la producción y mayores costos operativos.

En base a esta problemática se han realizado trabajos de investigación relacionados con el tema, Baig en su investigación [4] presenta una plataforma de comercio de energía P2P basada en blockchain de Ethereum y tecnología IoT, donde se podrá realizar el monitorear y controlar la energía generada en tiempo real, utiliza microcontroladores ESP32-S2 especialmente para aplicaciones industriales para procesar datos de energía (voltaje, corriente) recopilados por los sensores. Para la transmisión de estos datos utiliza módulos LoRa para el intercambio de información de manera inalámbrica entre el dispositivo de monitoreo de energía y la plataforma de comercio.

En el dispositivo que actuara como receptor a través del módulo LoRa implementa el protocolo MQTT para transmitir la información mediante el protocolo WiFi para después mediante Node-Red que es un editor de flujo donde se pueden agregar, eliminar y conectar nodos entre sí para permitir la comunicación entre ellos y funcionara como intermediario entre MQTT y React.JS para la plataforma de comercio. Detalla el diseño, implementación y pruebas exitosas del sistema, recalcando la integración efectiva de blockchain e IoT para mejorar el comercio de energía.

De igual forma, Jasmine Gnana [5] en su trabajo va enfocado en el sector agrícola ya que lo define como un desarrollo para la civilización, se menciona el surgimiento de la agricultura de precisión como el objetivo principal para optimizar procesos y producir

resultados significativos. Presentan un sistema basado en **IoT** para monitorear parámetros microclimáticos y evaluar los requisitos de riego, se utilizan sensores para evaluar el porcentaje de humedad y temperatura del suelo, de estos datos se integran en el sistema **LoRa** para la comunicación inalámbrica. En el sistema de recepción se realiza el análisis de los datos recopilados para determinar evapotranspiración que es un indicador para la gestión efectiva del recurso hídrico en la agricultura, emplearon un programa en Cropwat para la interpretación de la información almacenada y los hallazgos de este proyecto de investigación dirigieron el camino para estimar la evapotranspiración en el contexto del microclima agrícola, contribuyendo para una gestión precisa y eficiente de la suministración del agua en el sector agrícola.

Por otro lado, Guevara [6] en su estudio propone una solución de monitoreo de bajo costo mediante un sistema de **IoT** de variables eléctricas y proyectar cuanta energía eléctrica consumieron en los hogares o empresas pequeñas. Constan de un nodo electrónico con un sensor avanzado construido sobre el chip Atmel M90E32AS y gestionado por un microcontrolador ESP32, utilizan módulos de radiofrecuencia (**RF**) que operan a 915 MHz y el protocolo **LoRaWan** basado en el transceptor Semtech SX1278 para la comunicación inalámbrica, permitiendo la mediación y comunicación de las variables eléctricas como corriente, voltaje, energía activa, energía reactiva, factor de potencia y otras variables eléctricas en circuitos de hasta tres fases.

En base a las investigaciones previas se busca la aplicación de sistemas monitoreo, con el propósito de adquirir datos y simplificar la toma de decisiones fundamentadas en circunstancias variadas, cada trabajo utiliza **IoT** como herramienta crucial para recopilar datos y visualización en interfaces graficas. Utilizan sensores y dispositivos conectados para recopilar información precisa y pertinente en campos especializados como el comercio de energía, la agricultura, la eficiencia energética, todos buscan contribuir para una gestión más eficaz y sostenible en sus áreas de estudio. También comparten una relación al implementar **LoRa** brindando una comunicación inalámbrica de larga distancia y bajo consumo energético.

Justificación

En los países de desarrollo, se han enfocado en fortalecer el sector agroindustrial con la finalidad de mejorar sus alimentos en términos de calidad, incremento en la eficiencia productiva y la generación de oportunidades laborales. La clave para lograr estos objetivos radica en la tecnificación de los procesos agrícolas, donde se busca gestionar los recursos de manera eficiente y optimizar la producción de alimentos basándose en datos recopilados de parámetros ambientales.

La importancia y los alcances de un sistema de monitoreo en sectores agrícolas son notables, especialmente en áreas que aún no han implementado esta tecnología. Este enfoque brinda numerosos beneficios para los agricultores y las comunidades rurales del cantón Gualaceo.

Permite la optimización de recursos primordiales como el agua, al proporcionar datos en tiempo real que permiten un riego más eficiente y ajustes precisos en las operaciones agrícolas según las necesidades reales de los cultivos, esto no solo reduce los costos para los agricultores, sino que también contribuye a la conservación de recursos escasos. El acceso al historial de los datos recopilados permite tomar decisiones informadas en un entorno agrícola en constante cambio, lo que es esencial en la región andina donde la variación climática es una realidad. Además, estos datos pueden promover la investigación y proponer nuevas prácticas agrícolas para el desarrollo de tecnologías innovadoras.

En general, para un sistema de monitoreo remoto eficaz de parámetros ambientales, es necesario utilizar tecnologías de comunicación como [LoRa](#), integrar una variedad de sensores climáticos, garantizar la eficiencia energética, crear una interfaz de usuario y priorizar el almacenamiento de información. Además, para que el sistema sea

escalable y adaptable a diversos contextos para garantizar la precisión de las mediciones, ya que son características esenciales para hacer frente a una variedad de necesidades cambiantes de monitoreo del clima.

Por esta razón, el sistema IoT propuesto para el monitoreo remoto de variables ambientales como temperatura, radiación ultravioleta (UV) y humedad del suelo también utiliza módulos LoRa para la transmisión de información a largo alcance, alta tolerancia a las interferencias y bajo consumo de energía. La información se recopila, almacena y se muestra en una interfaz gráfica con el propósito previsto de observar el cambio en cada variable en un huerto agrícola a lo largo del tiempo.

Objetivos

Objetivo General

- Desarrollar un sistema IoT para el monitoreo remoto de parámetros ambientales en un huerto agrícola de hortalizas.

Objetivos específicos:

- Diseñar un dispositivo electrónico para la monitorización en tiempo real de la temperatura ambiente, humedad del suelo y radiación UV.
- Implementación del sistema de comunicación con tecnología LoRa.
- Desarrollar una plataforma para el monitoreo de los parámetros ambientales.
- Validar las mediciones obtenidas, mediante pruebas de campo.

Introducción

La medición y recolección de variables climáticas es de gran importancia en los huertos agrícolas, para garantizar el proceso de cultivación. Con esto se logra la planificación, operación y mantenimiento efectivos de los huertos agrícolas. Para esto primero se analiza cada variable climática que influyente en estos huertos agrícolas y para qué son importantes de evaluar como la temperatura ambiente, radiación UV y humedad del suelo.

Estas variables desempeñan roles importantes en un huerto agrícola. La temperatura influye en la fase de desarrollo de las plantas y la fotosíntesis. La humedad ambiental impacta en la absorción de agua y la transpiración, esencial para el equilibrio hídrico de las plantas. La radiación UV puede tener efectos beneficiosos en la producción de compuestos útiles, pero un exceso puede resultar perjudicial para las células vegetales. La humedad del suelo es crucial para el provisión de agua y nutrientes para para los cultivos.

El desarrollar un sistema para el monitoreo remoto y recolección de datos climáticos basado en IoT se proyecta como una forma efectiva de garantizar el proceso de cultivación. Este sistema, conformado de sensores de precisión, tiene la capacidad de recolectar información precisa sobre diferentes variables climáticas en tiempo real. Al hacer uso de la conectividad IoT, el sistema recopila estos datos y los transmite de manera inalámbrica mediante Wi-Fi a una plataforma en la nube. Este enfoque posibilita la monitorización eficaz y constante de las condiciones climáticas en ubicaciones remotas, permitiendo ajustar estrategias de cultivo y sistemas riego, asegurando un manejo eficiente y sostenible del huerto agrícola.

Para eso se tomará en cuenta el proceso de seis artículos referentes que realizo

cada autor al tema propuesto, analizando el motivo de implementación en cada uno de los proyectos, dando una solución a la problemática que enfrento cada uno, y en base de esa información se realizará una comparativa de cada artículo haciendo énfasis puntos no se tomaron en cuenta, con el objetivo de empezar con el proyecto de titulación cuyo tema es el monitoreo y recolección de datos climáticos basado en [IoT](#).

Capítulo 1

Fundamentación Teórica

En este apartado, se explorarán conceptos teóricos fundamentales relacionados con la tecnología IoT en la agricultura, con el objetivo de proponer una solución al problema planteado. Se examinan los elementos clave que sirven como fundamento para llevar a cabo la investigación. Además, se detallan las tecnologías empleadas en la ejecución del proyecto, tales como: [LoRa](#), Node-RED, InfluxDB y Grafana.

1.1. Agricultura 4.0

Se destaca por la automatización, adquisición y análisis de información, permitiendo una gestión eficiente de recursos, optimizando costos y aumento de la eficiencia productiva. Estos procesos mejoran la eficiencia agrícola en beneficio de los agricultores y la sostenibilidad. Se fundamenta en la fusión de tecnología digital, [IoT](#) y sensores de campo . Estos elementos posibilitan la optimización y gestión más eficiente y sostenible de las explotaciones agrícolas, reduciendo el consumo de recursos y mejorando la calidad de los productos [3].

La agricultura de precisión, emplea datos y tecnologías para mejorar la eficiencia en las prácticas agrícolas. Esto abarca el uso de información meteorológica y del suelo, para el control preciso de la aplicación de fertilizantes [3].

1.1.1. Tecnologías utilizadas en la Agricultura 4.0

La integración de tecnologías en la agricultura ofrece múltiples aplicaciones. Se puede emplear para recopilar datos relevantes, facilitando análisis que optimizan el uso de recursos. Además, estas tecnologías posibilitan la automatización y eficiencia en las labores agrícolas, mejorando la gestión de cultivos [7]. Algunas de las tecnologías que se utilizan en la agricultura 4.0 son:

- **Drones:** Se emplean drones para inspeccionar campos agrícolas, equipados con sensores para recabar datos. Usualmente llevan cámaras para capturar imágenes aéreas que proporcionan información sobre la fertilidad del suelo y otros datos útiles sobre el desarrollo de los cultivos [7].
- **Sensores ambientales:** Estos dispositivos proporcionan datos cruciales como temperatura, humedad y calidad del suelo, informando sobre las necesidades específicas de los cultivos. Esta información permite ajustar estrategias agrícolas, como la dosificación precisa de fertilizantes, optimizando así el uso de recursos y mejorando la producción agrícola [8].
- **Internet de las cosas (IoT):** Se concreta al conectar dispositivos, facilitando la transferencia instantánea de datos entre ellos. Esta interconexión posibilita la automatización y la toma de decisiones respaldada por información o análisis de datos en diversos ámbitos [8].
- **Robótica:** En la agricultura 4.0, la automatización mediante brazos robóticos o vehículos autónomos permite realizar tareas de mantenimiento en campos y monitorear cultivos con sensores, facilitando la toma de decisiones [7].

1.1.2. Beneficios de la Agricultura 4.0

La incorporación de tecnologías en la Agricultura 4.0 aporta notables beneficios:

- Optimiza la producción, reduciendo costos y aumentando rendimientos.
- Mejora la sostenibilidad al gestionar eficientemente pesticidas y fertilizantes.

- La digitalización facilita un acceso inmediato a datos esenciales y permite la monitorización rápida de los cultivos.
- La automatización y manejo de información aceleran las actividades agrícolas, logrando eficiencia y rapidez.

Estas ventajas posicionan a la Agricultura 4.0 como un enfoque que no solo incrementa la productividad, sino que también promueve la gestión ambientalmente consciente y la toma de decisiones informada en tiempo real [3].

1.2. Huerto agrícola casero

Es un espacio destinado al cultivo de vegetales, hierbas y plantas comestibles, ubicada en ambientes residenciales como se muestra en la figura 1.1. Concebido para aprovechar espacios como balcones, patios o terrazas, su propósito principal es suministrar alimentos frescos y orgánicos directamente a los hogares. Aunque los huertos caseros pueden variar en dimensiones y diseño, generalmente adoptan prácticas de cultivo sostenible, como el uso de fertilizantes orgánicos y métodos de riego eficientes. Estos huertos son óptimos para el cultivo de variedades adaptadas a las condiciones locales y pueden incorporar técnicas de jardinería como la rotación de cultivos. Este enfoque promueve una conexión directa con la producción alimentaria y fomenta hábitos más saludables y sostenibles [9].



Figura 1.1: Huerto agrícola casero [Fuente: Autor]

1.2.1. Requerimientos ambientales del cultivo

En este apartado se exponen los requisitos de cultivo para las hortalizas para una óptima producción como son los parámetros climáticos y edafoclimáticos. Adicionalmente, se analizará cada variable del entorno ya que afectan en el proceso de desarrollo y productividad de los cultivos agrícolas. Estas variables influyen en la etapa de crecimiento mediante la fotosíntesis, respiración, transpiración, captación de nutrientes y agua, para la etapa de floración de las hortalizas. Cada cultivo o variedad tiene sus propios requerimientos de estas variables para alcanzar un óptimo crecimiento [10].

1.2.2. Parámetros climáticos

Temperatura ambiente

Desempeña una función crucial en el progreso y rendimiento en los procesos metabólicos de las plantas, como la brotación de las semillas, crecimiento, fotosíntesis y floración [11].

En la tabla 1.1 se representa una referencia con los rangos de temperatura y su impacto en cultivos.

Tabla 1.1: Rangos de temperatura e impacto en cultivos [11]

Temperatura (°C)	Impacto en Cultivos
<0	Peligro de heladas, daño a cultivos sensibles al frío
0–5	Frío extremo, riesgo de daño en cultivos delicados
6–10	Frío moderado, algunos cultivos pueden ser afectados
11–15	Rango fresco, adecuado para muchos cultivos
16–21	Temperatura agradable, óptima para muchos cultivos
22–27	Cálido, buen crecimiento para la mayoría de los cultivos
28–32	Cálido a caliente, algunos cultivos pueden sufrir estrés
>32	Calor extremo, riesgo de daño y estrés térmico en cultivos

Cada cultivo tiene un rango óptimo de temperatura para realizar estos procesos de forma eficiente, en el caso de condiciones extremas ya sea que la temperatura sea

demasiadamente alta o baja, ocasiona un estrés en las plantas lo que causa su bajo rendimiento [11].

En la tabla 1.2 representa una referencia con los rangos óptimos de temperatura para cada proceso para los cultivos que mayormente siembran en los huertos caseros .

Tabla 1.2: Rango óptimo de temperatura [11]

Cultivo	Crecimiento (°C)	Fotosíntesis (°C)	Floración (°C)
Col	15 - 20	15 - 25	15 -20
Brócoli	15 - 20	15 - 25	15 -20
Lechuga	10 - 18	15 - 20	10 - 15
Zanahoria	15 - 20	15 - 25	15 - 20
Maíz	18 - 30	25 - 30	18 - 24

Radiación UV

Es la fuente de energía para la etapa de crecimiento y producción de materia orgánica. Se compone de diferentes longitudes de onda, de las cuales la más importante es la RFA, que abarca el espectro visible de 100 a 400 nm. La cantidad de radiación solar que incidente a las plantas varía según la estación del año, hora, nubosidad y el tipo de cubierta vegetal [12].

La gestión adecuada de la radiación UV en la agricultura, implica monitoreo y estrategias de protección para optimizar la producción y preservar la salud de los cultivos como se representa en la tabla 1.3.

Tabla 1.3: Índice de radiación ultravioleta (UV) y su impacto en cultivos [12]

Índice UV	Riesgos para Cultivos
0-2	Los cultivos generalmente no se ven afectados significativamente
3-5	Se recomienda precaución y protección para ciertos cultivos
6-7	Se deben tomar medidas de protección, como sombreado o riego adicional
8-10	Se necesitan precauciones intensivas, como sombreado y riego frecuente
11+	Se deben tomar medidas extremas para proteger los cultivos

En la tabla 1.4 representa la cantidad óptima de radiación solar para los cultivos hortícolas depende de la especie, pero generalmente se sitúa entre 1.15 y 2.30 mW/cm² [12].

Tabla 1.4: Rango óptimo de radiación UV [12]

Cultivo	Intensidad de luz UV óptima (mW/cm ²)
Col	0.5 - 1.0
Brócoli	0.5 - 1.0
Lechuga	0.3 - 1.0
Zanahoria	0.5 - 1.5
Maíz	1.0 - 2.0

1.2.3. Parámetros edafoclimáticos

Humedad del suelo

Representa el porcentaje del suministro de agua disponible que contiene la superficie de la plantación, esta variable influye en el crecimiento y rendimiento, ya que afecta en los procesos de captación de componentes nutritivos y crecimiento de raíces [13]. La medición de esta variable permite realizar la optimización del uso correcto del agua para gestionar de mejorar manera los sistemas de riego [14].

En la tabla 1.5 representa una referencia con los rangos de humedad del suelo y su impacto en cultivos.

Tabla 1.5: Rangos de humedad del suelo y su impacto en cultivos [13]

Humedad del Suelo	Impacto en Cultivos
<20 %	Sequedad extrema, riesgo de estrés hídrico para los cultivos
20-30 %	Sequedad, algunos cultivos pueden experimentar estrés hídrico
30-50 %	Rango óptimo para la mayoría de los cultivos
50-70 %	Humedad adecuada, condiciones favorables para el crecimiento
70-80 %	Humedad alta, riesgo de enfermedades fúngicas y de raíces
>80 %	Exceso de humedad, riesgo de anegamiento y asfixia de raíces

En cambio la tabla 1.6 representa los rangos óptimos del porcentaje ideal para los cultivos que mayormente siembran en los huertos caseros.

Tabla 1.6: Rango óptimo de humedad del suelo [13]

Cultivo	Humedad del suelo óptima (%)
Col	60 - 80
Brócoli	60 - 80
Lechuga	70 - 90
Zanahoria	60 - 80
Maíz	50 - 70

1.3. Internet de las Cosas (IoT)

Enlaza objetos comunes con Internet, permitiendo así el intercambio de datos y la ejecución de tareas inteligentes. Estos objetos pueden adquirir y enviar información debido a la incorporación de sensores y componentes de comunicación. Después de eso, estos datos se procesan y examinan para obtener información valiosa [15]. IoT ofrece una variedad de aplicaciones, desde hogares y ciudades inteligentes hasta industrias y salud. IoT mejora la eficiencia, la seguridad y la calidad de vida al cambiar la forma en que interactuamos con nuestro entorno. Esta conexión sigue progresando, prometiendo un futuro cada vez más conectado y automatizado [5].

1.3.1. Propiedades del IoT

Permiten la realización de funciones específicas a través del uso predominante de software y hardware [15]. A continuación, se proporcionan detalles sobre los aspectos fundamentales del IoT:

- **Conectividad:** Implica la interconexión de dispositivos permitiendo interacciones fluidas y recopilación de datos.
- **Servicios vinculados:** La conexión entre objetos físicos y su representación virtual permite la ejecución de servicios con propósitos específicos.

- **Heterogeneidad:** Característica de variedad en términos de tipos de plataformas de interacción [15].

1.3.2. Arquitectura del IoT

Como se visualiza en la figura 1.2, la estructura se compone de tres categorías: sensores, Gateway y plataforma en la nube. En el primer nivel, los sensores capturan datos del entorno. En cambio en el segundo nivel, la puerta de enlace facilita las comunicaciones y realiza acciones locales antes de enviar datos a la nube. Por último, en el tercer nivel la plataforma en la nube, se encarga de diversas tareas, como la gestión de equipos, almacenamiento, procesamiento de datos, y soporte de aplicaciones [15].

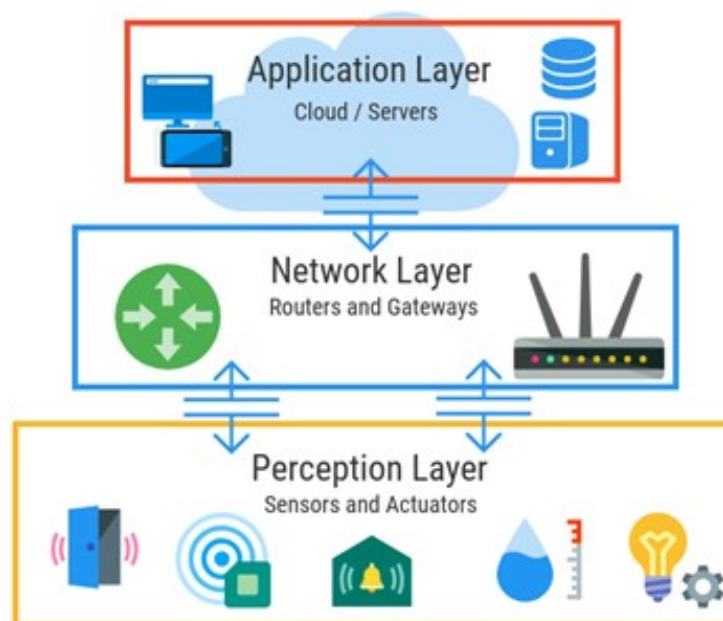


Figura 1.2: Arquitectura de una red IoT [15]

1.3.3. Tecnologías para IoT

Las tecnologías de conectividad inalámbrica son fundamentales en IoT. Wi-Fi ofrece rápida conexión pero alcance limitado. Zigbee y Z-Wave son protocolos de malla eficientes para la automatización del hogar. LoRa permite comunicación a larga distancia con baja potencia [16]. NB-IoT se centra en dispositivos de bajo consumo conectados

a redes celulares. La tecnología 5G promete velocidades ultra rápidas y baja latencia. Estas opciones permiten a los dispositivos IoT comunicarse eficientemente, adaptándose a diversas necesidades de conectividad [17].

La Tabla 1.7 proporciona un desglose de los atributos clave de las tecnologías más utilizadas en en aplicaciones IoT .

Tabla 1.7: Atributos de los principales [17]

Propiedades	Wifi	Zigbee	LoRa
Tasa de transferencia	600 Mbps	20-250 Kbps	0.3-50 Kbps
Banda de transmisión	2.4 GHz y 5 GHz	915 MHz y 2.4 GHz	868 MHz-915 MHz
Potencia de transmisión	15 dBm-20 dBm	0 dBm-20 dBm	5 dBm-30 dBm
Rango	32-40 m	10-100 m	10 km
Modulación	BPSK-QPSK	O-QPSK	LoRa

1.4. IoT en la Agricultura

Ofrece soluciones tecnológicas para mejorar la eficiencia y gestión de cultivos en entornos caseros. Mediante sensores y dispositivos conectados, los agricultores caseros pueden monitorear condiciones ambientales para el crecimiento de las plantaciones. Además, la existencia de aplicaciones y plataformas IoT proporcionan datos en tiempo real y alertas, facilita a los usuarios el seguimiento y ajuste en base a las condiciones para optimizar el rendimiento de sus cultivos [3].

1.4.1. Aplicaciones en Agricultura

Varias aplicaciones del IoT están cambiando la gestión de cultivos como:

- **Supervisión y gestión del riego:** A través de sensores conectados, los agricultores acceden a la información de forma instantánea las necesidades hídricas del suelo de las plantas. Esto facilita la optimización del riego, evitando el desperdicio de agua, una consideración crucial en áreas con escasez de recursos hídricos. La aplicación

de esta tecnología permite una gestión más eficiente y sostenible del recurso, contribuyendo al uso responsable del agua en la agricultura [18].

- **Monitoreo de las condiciones climáticas:** Sensores conectados registran datos como temperatura, humedad y radiación solar. Estos datos son vitales para programar el tiempo de cosecha adecuado y saber el momento óptimo para realizar siembras, riegos, mejorando así la eficiencia y productividad de los cultivos [19].

La implementación del IoT en la agricultura, a través del monitoreo de riego y condiciones climáticas cómo se ilustra en la figura 1.3, ofrece a los agricultores datos en tiempo real para tomar decisiones informadas en el cultivo [19].



Figura 1.3: Monitoreo de variables [3]

1.4.2. Beneficios en la Agricultura

Mediante la conexión de dispositivos inteligentes a Internet, los agricultores mejoran eficiencia, productividad y rentabilidad [18]. A continuación, se describen algunos beneficios clave de la implementación:

- **Aumento de la eficacia y rendimiento:** Los agricultores pueden supervisar y gestionar aspectos críticos de sus cultivos de forma remota. Sensores y dispositivos

conectados proporcionan datos instantáneos sobre las variables climáticas de los cultivos. Esta información permite decisiones fundamentadas y una gestión eficaz de recursos como el riego y la fertilización [7].

- **Disminución de gastos:** Reduce costos al maximizar la utilización de recursos hídricos y fertilizantes. El monitoreo remoto permite detectar problemas tempranos, evitando pérdidas, ahorrando tiempo y contribuyendo a la eficiencia financiera [3].
- **Mejoría en la calidad de las cosechas:** Los dispositivos conectados permiten un monitoreo constante, garantizando condiciones ambientales y prácticas agrícolas óptimas para obtener cultivos de alta calidad. La información detallada sobre cada cultivo posibilita ajustar técnicas de cultivo y manejo según necesidades específicas, mejorando su calidad y rendimiento [15].

1.5. Tecnología LoRa

1.5.1. LoRa

Esta tecnología se caracteriza por su alcance extenso en el tema de comunicación inalámbrica y su eficiencia energética. Emplea la técnica de modulación CSS que puede ser generada por los chips transceptores LoRa de Semtech [16], como se puede ver en la figura 1.9.

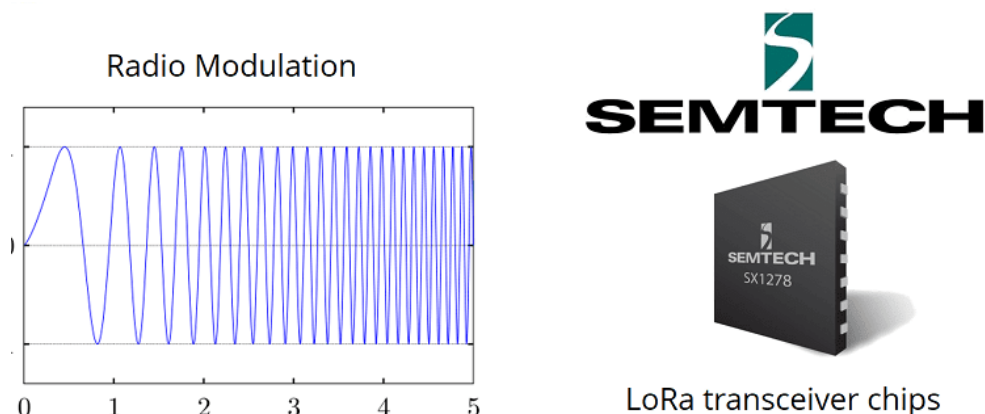


Figura 1.4: Técnica de modulación [16]

La tecnología LoRa utiliza bandas de frecuencias para su correcta operación, dependiendo de la región geográfica y las regulaciones locales tal como se visualiza en la figura 1.5.

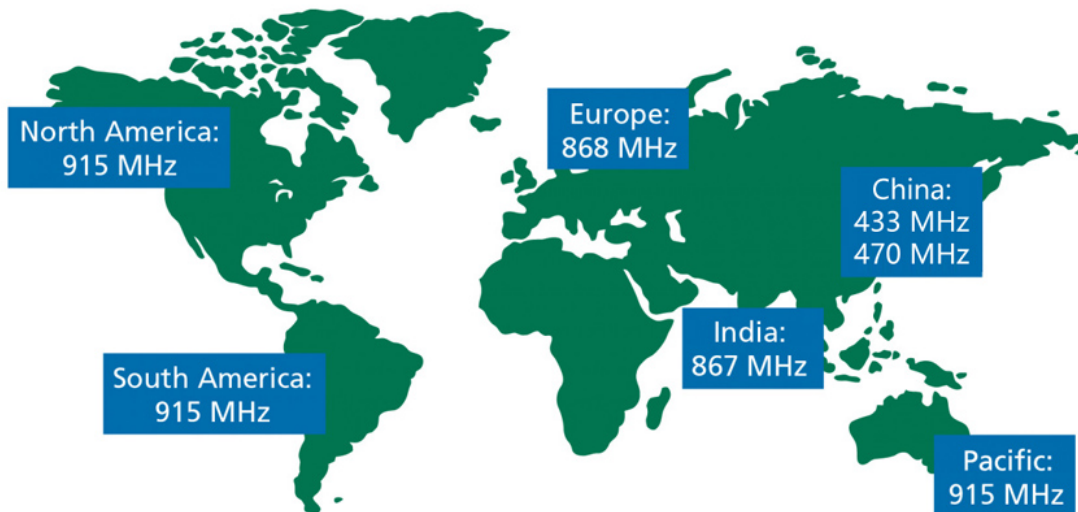


Figura 1.5: Frecuencias LoRa [20]

Posibilita la transmisión de datos a grandes distancias, siendo particularmente idónea para situaciones que demandan una extensa cobertura y un consumo energético reducido, por ejemplo los sistemas de monitoreo a distancia en zonas agrícolas, redes de sensores urbanos, rastreo de activos y diversas otras aplicaciones [20]. Al operar en estas bandas de frecuencia, LoRa proporciona una solución confiable para la conectividad inalámbrica en proyectos IoT diversificados [21].

Ventajas:

- Gran capacidad de resistencia a las interferencias.
- Gran capacidad de recepción de datos con alta sensibilidad (-168 dB).
- Amplio rango de cobertura de 10 a 20 kilómetros.
- Limitada capacidad de transferencia de datos (hasta 255 bytes).
- Enlace directo entre dos puntos de comunicación [22].

1.5.2. LoRaWAN

Se trata de un protocolo de red para dispositivos IoT que utiliza tecnología LoRa se representa en una topología de tipo estrella como se muestra en la figura 1.8, que consiste en la conexión de nodos finales entre sí mediante un Gateway o varios, que dirigen la información a un servidor de red [21]. Se compone de por cuatro elementos principales:

- **Dispositivos de red:** Los nodos son dispositivos terminales que envían y reciben datos a través del punto de acceso central.
- **Puerta de enlace:** Es el responsable de la recepción y transmisión de datos desde y hacia los nodos de la red.
- **Servidor de red:** Se enlaza con múltiples gateways mediante una conexión TCP/IP de forma inalámbrica, además elimina los mensajes duplicados.
- **Servidor de aplicaciones:** Recopila y analiza los datos provenientes de los nodos finales, luego decide las acciones a tomar por el sistema [21].

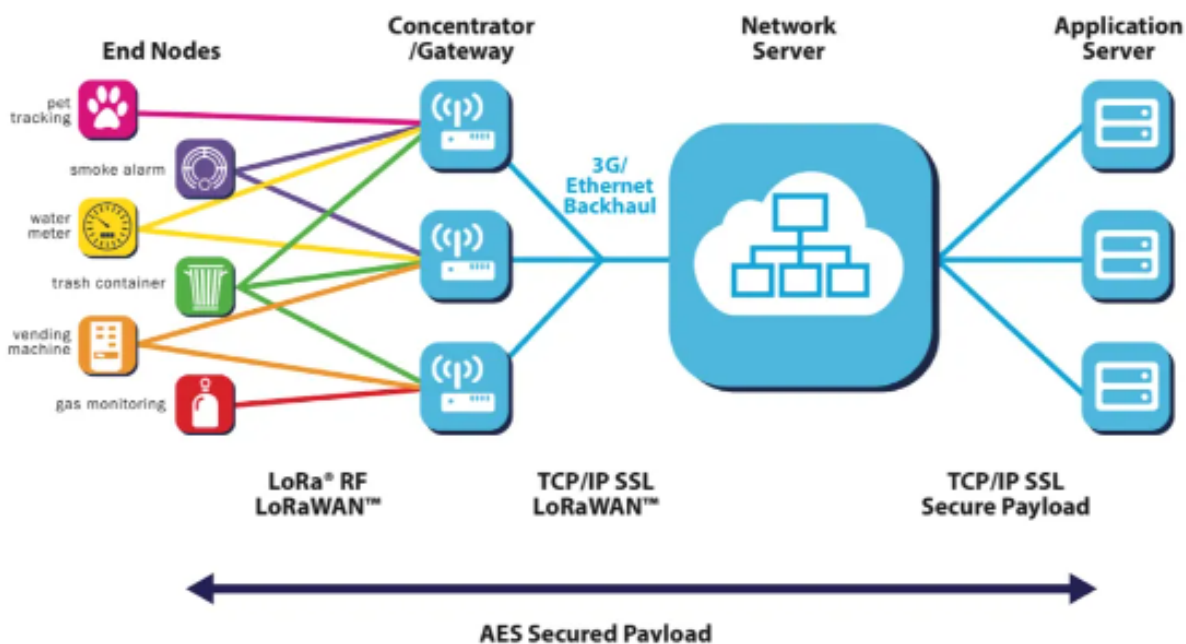


Figura 1.6: Arquitectura LoRaWAN [21]

Existen tres categorías de clases para los nodos:

- **Clase A:** La clase más ampliamente respaldada en prácticamente todos los dispositivos, brinda el ahorro de energía máximo al entrar en modo de escucha solo después de enviar información al gateway.
- **Clase B:** Estos dispositivos operan con ventanas de recepción basadas en intervalos de tiempo predefinidos con el gateway.
- **Clase C:** Consume más energía ya que permanece en modo de escucha continua, solo transmitiendo cuando es necesario. Es adecuada para dispositivos con alimentación externa [23].

Ventajas:

- Comunicaciones seguras bidireccionales con cifrado de extremo a extremo.
- Amplio rango de cobertura de 10 a 20 kilómetros.
- Conexión de múltiples sensores y dispositivos a redes públicas o privadas.
- Velocidades de datos reducidas.
- Frecuencia de transmisión baja, movilidad limitada y servicios de localización mínimos [22].

1.5.3. Topologías

La disposición o estructura de una red, conocida como topología, define cómo los nodos se organizan para promover una comunicación efectiva, que cumplen la función de identificar la vía más eficaz para la transmisión de datos [24]. Existen dos tipos:

Punto a Punto (P2P)

Dos dispositivos LoRa establecen una conexión directa entre ellos. Uno desempeña el papel de emisor (transmitiendo datos) y el otro asume el rol de receptor (recibiendo datos). La comunicación es directa y simple entre estos dos nodos [25].



Figura 1.7: Conexión punto a punto [25]

Multipunto

Facilita la conexión directa entre varios nodos, permitiendo la comunicación sin depender de un punto central. Cada nodo puede interactuar directamente con otros, brindando flexibilidad y eficiencia en la transferencia de información dentro de la red [26].

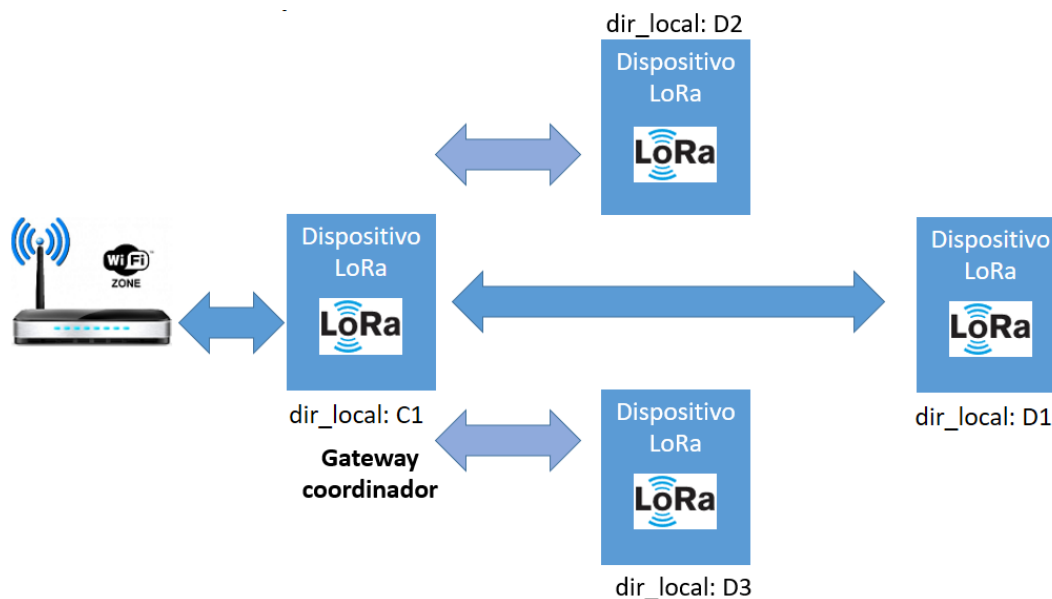


Figura 1.8: Conexión multipunto [25]

Estas topologías ofrecen flexibilidad para adaptarse a diferentes necesidades y desafíos de implementación. Al seleccionar una topología, es esencial considerar factores como la distancia entre nodos, la redundancia de la red y la escalabilidad. La elección dependerá del entorno específico y los requisitos de la aplicación en la red LoRa [24].

1.5.4. Modelo OSI

Representa de una manera general cómo se estructuran las funciones de comunicación en las redes que utilizan la tecnología LoRa [27].

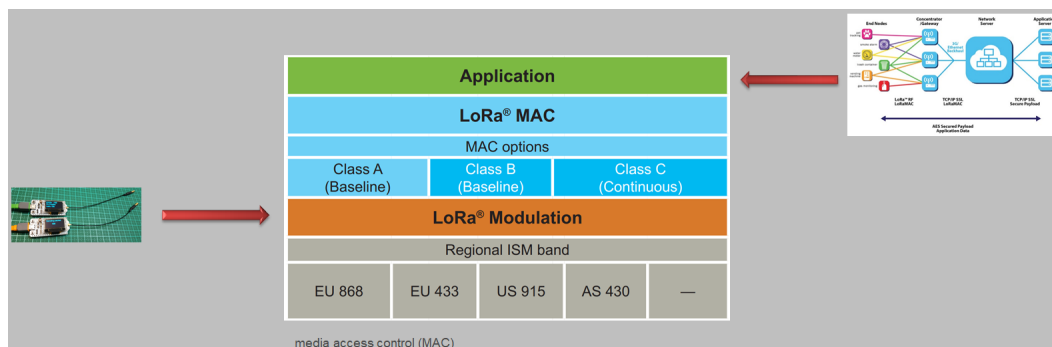


Figura 1.9: Modelo OSI [27]

- **Capa física:** Abarca la modulación de espectro ensanchado para la comunicación de información a grandes distancias de forma inalámbrica. Se enfoca en la eficiencia de la transmisión, adaptándose a diversas condiciones del canal [27].
- **Capa de red y enlace :** Governa la interconexión de dispositivos LoRa y una infraestructura central, gestionando aspectos como el acceso al medio, la seguridad y el enrutamiento [27].

1.5.5. LoRa en la agricultura

Se destaca como una aplicación específica de IoT en la producción agrícola, la integración de la tecnología LoRa en la agricultura convencional está transformando el panorama futuro de la agricultura a nivel mundial [28].

Aplicaciones

- **Invernadero inteligente:** La conectividad inalámbrica LoRa permite el control automático remoto, recolectando datos ambientales en tiempo real para optimizar el rendimiento y reducir la gestión manual [28].
- **Agricultura de precisión:** Enfocada en tratamientos precisos para cultivos, la implementación de sensores LoRa permite la medición precisa de variaciones en

el campo, permitiendo ajustes específicos de insumos según las condiciones de crecimiento [28].

- **Seguimiento de las condiciones climáticas:** El monitoreo climático es esencial en la agricultura inteligente, ya que las condiciones atmosféricas afectan significativamente desde el inicio del proceso de siembra hasta la recolección de los cultivos. La tecnología LoRa y sensores inteligentes permiten prever información meteorológica, brindando datos procesables para una gestión agrícola más eficiente y mejorando los rendimientos [28].

1.6. MQTT

Es un procedimiento de mensajería para IoT, que facilita la comunicación entre dispositivos mediante su estructura cliente-servidor facilita la transferencia de información entre dispositivos de manera asíncrona. En este sistema, los dispositivos publican mensajes en temas específicos, y los receptores se suscriben a temas de interés. Facilita la transmisión eficiente de datos en tiempo real, asegurando una conexión confiable a través del protocolo TCP/IP [29].

El broker MQTT actúa como el intermediario central, gestionando la entrega de mensajes entre dispositivos editores y suscriptores, permite una comunicación bidireccional y escalable en entornos IoT, donde múltiples dispositivos necesitan intercambiar datos de manera eficiente entre dispositivos [30].

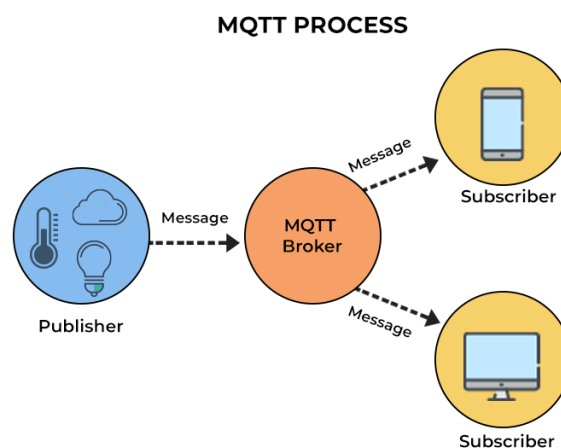


Figura 1.10: Arquitectura de MQTT [29]

1.7. Node-RED

Se trata de un sistema de libre acceso que facilita el desarrollo de aplicaciones IoT y flujos de trabajo de automatización [15]. Diseñado con una interfaz visual intuitiva, Node-RED brinda a los usuarios conectar equipos y plataformas de nube de manera eficiente. Sus nodos predefinidos, que representan funciones específicas, se pueden arrastrar y soltar para construir flujos de trabajo de forma fácil para la automatización y la interconexión de dispositivos. Además, es altamente personalizable, lo que lo hace adecuado para proyectos de diferentes escalas y complejidades. Node-RED se ha convertido en una herramienta esencial para desarrolladores y entusiastas de IoT en la creación de soluciones inteligentes y conectadas [31].

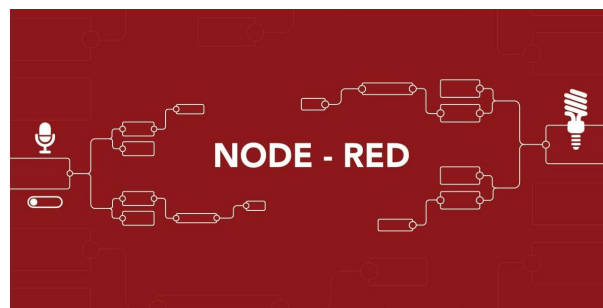


Figura 1.11: Node-RED [31]

1.8. InfluxDB

Es un sistema que administra archivos de registros temporales, diseñado específicamente para almacenar, consultar y visualizar datos temporales, como métricas y eventos de monitoreo. A diferencia de MySQL, InfluxDB se centra en la eficiencia de almacenamiento y recuperación lo que facilita el manejo de grandes volúmenes de datos temporales[32].

Es especialmente adecuado para entornos donde la recopilación y análisis de datos de series temporales son cruciales, como en sistemas de monitoreo, IoT y aplicaciones relacionadas con el tiempo [15].

InfluxDB utiliza un lenguaje de consulta llamado InfluxQL y ofrece una interfaz HTTP y API para trabajar con la base de datos. Su arquitectura escalable y su enfoque en

datos temporales hacen que sea una elección popular para proyectos que requieren un rendimiento optimizado en el manejo de datos temporales [32].

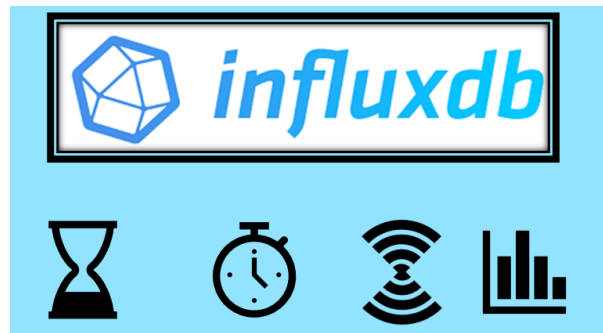


Figura 1.12: InfluxDB [32]

1.9. Grafana

Grafana se posiciona como una página web de libre acceso destinada a la representación de datos, esencial en el análisis de información, su función primordial consiste en establecer conexiones y presentar datos guardados en distintas fuentes de información, que abarcan desde bases de datos hasta y servicios ubicados en la nube [33].

La adaptabilidad de Grafana surge de su capacidad para personalizar y organizar la visualización de acuerdo con las necesidades únicas del usuario. Además, es posible configurar alertas que notifiquen a los usuarios cuando se alcancen umbrales o condiciones específicos, lo que permite respuestas rápidas a los cambios de datos [33].



Figura 1.13: Grafana [33]

Capítulo 2

Marco Metodológico

La metodología se describe en detalle presentando una sinopsis de los componentes electrónicos empleados en la creación del sistema, junto con la manera en que se comunican entre sí, el protocolo de comunicación inalámbrica utilizado, elaboración del circuito impreso y diseño 3D de la caja de implementación.

2.1. Esquema de bloques del sistema

La figura 2.1 ofrece un esquema visual del diagrama de bloques que rige el funcionamiento integral del sistema. El esquema operativo se basa en un enfoque de tres fases esenciales para el monitoreo de variables:

- **Primera etapa:** Se implementan sensores específicos para recoger con exactitud las variaciones de temperatura, humedad del suelo y radiación UV.
- **Segunda etapa:** Posteriormente, estos datos son transmitidos de manera inalámbrica a través de la tecnología LoRa desde el nodo final hasta el Gateway LoRa que funciona como mediador, habilitando la interconexión entre el dispositivo y la red.
- **Tercera etapa:** Finalmente, mediante Node-RED se gestiona la información de los sensores a un almacén de información en la nube donde se registra cada parámetro recopilado. Esta información se presenta de forma accesible mediante

el desarrollo de una plataforma de visualización a través de la interfaz web de Grafana, facilitando el monitoreo remoto de las condiciones del huerto.

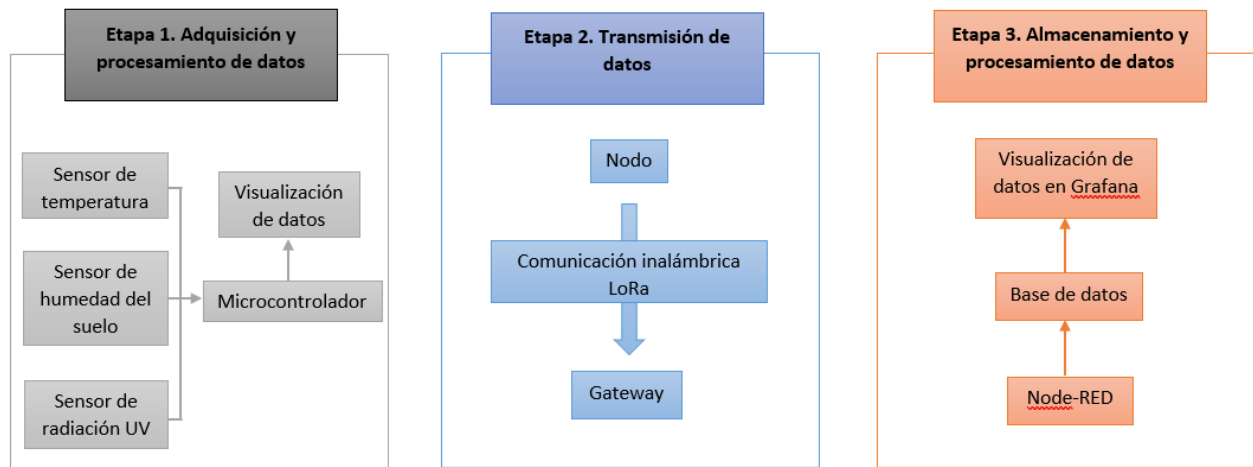


Figura 2.1: Esquema visual[Fuente: Autor]

2.2. Componentes electrónicos

2.2.1. Heltec Wi-Fi LoRa 32 (V2)

El microcontrolador ESP32 Heltec es ampliamente reconocido debido a su avanzada funcionalidad y alta eficacia. Es parte de la familia ESP32, que es conocida por su conectividad inalámbrica y capacidades de procesamiento potentes [34]. Está fabricado por la empresa Heltec Automation y está diseñado para combinar las ventajas del ESP32 con una pantalla OLED y un módulo LoRa en un solo dispositivo compacto y eficiente [35].

Cuenta con un display OLED, esta característica se torna especialmente valiosa en proyectos de IoT, ya que proporciona información sin la necesidad de visualizar en el computador los datos del puerto serial [35].

Incorpora un módulo LoRa para la transmisión de datos entre dispositivos, especialmente en aquellas regiones donde la disponibilidad de Wi-Fi o señal de telefonía móvil puede ser limitada. Como resultado, se incrementan las posibilidades de aplicación del ESP32 Heltec en situaciones donde la cobertura presenta un desafío [36].

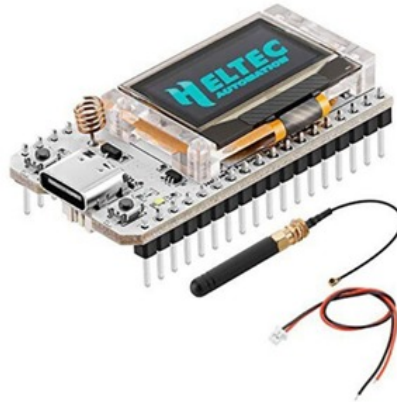


Figura 2.2: Módulo WiFi LoRa 32 V3 SX1262 de 915 MHz ESP32 Bluetooth OLED [35]

Características:

- Basado en el microcontrolador ESP32-S3FN8 (Xtensa® 32-bit LX7 dual-core hasta 240 MHz).
- LoRa chip: SX1262.
- Banda de frecuencia de soporte: 863 - 928 MHz.
- Distancia de comunicación abierta: 2.8 Km.
- WiFi: 802.11 b/g/n.
- Interfaz SH1.25-2 para batería de litio (Li-Po) de 3.7V.
- Potencia de salida de 20 dBm.
- Sensibilidad del receptor: -139 dBm.
- Voltaje de funcionamiento: 3.3 a 7 V.
- Consumo de corriente : 70 mA.
- Pantalla OLED de 0.96".
- 36 pines (28 GPIO) [35].

2.2.2. Sensor de Temperatura Ambiente DHT22 AM2302

Tiene una medición de temperatura altamente preciso y confiable. Su rango de temperatura puede ser de -40°C a 80°C , su diseño compacto permiten una fácil integración en proyectos de IoT, estaciones meteorológicas y sistemas de control de climatización. Siendo así una elección popular para el monitoreo ambiental y el control de condiciones en interiores y exteriores [37].



Figura 2.3: Sensor de Temperatura Ambiente (DHT21 AM2302) [37]

Características:

- Voltaje de alimentación 5V.
- El consumo de corriente es de 3 mA.
- Señal de salida digital.
- El rango de temperatura es de -40 a 80°C con una precisión de $\pm 0.5^{\circ}\text{C}$.
- La humedad varía de 20 a 90 % con una precisión de $\pm 2\%$ de humedad relativa.
- Dimensión aproximada 60x25 mm [38].

2.2.3. Sensor de Humedad del Suelo HD-38

Es esencial en la área de la agricultura y jardinería, proporciona mediciones precisas de la humedad del suelo, lo que ayuda a los agricultores y jardineros a

determinar cuándo regar las plantas de manera eficiente. Su facilidad de integración, durabilidad y capacidad para resistir las condiciones del suelo lo convierten en una herramienta valiosa para proyectos de riego automático y agricultura de precisión [39].



Figura 2.4: Sensor HD-38 [39]

Características:

- Rango de medición de humedad del suelo: Depende del modelo específico, pero típicamente de 0% a 100% de humedad.
- Las puntas del sensor son resistentes a la oxidación.
- Interfaz de comunicación: Analógica o digital, según el modelo.
- Voltaje de alimentación: 3.3V a 12V.
- El consumo de corriente es de 30 mA.
- Sensibilidad de salida digital ajustable por potenciómetro [40].

2.2.4. Sensor de Radiación UV ML8511

Su capacidad para medir la intensidad de la radiación UV, es crucial para aplicaciones de protección solar. Se utiliza en dispositivos portátiles de protección solar, estaciones meteorológicas y sistemas de monitoreo ambiental. También se aplica en

la agricultura para monitorear la radiación **UV** que puede influir en la fotosíntesis y desarrollo de cultivos [41].

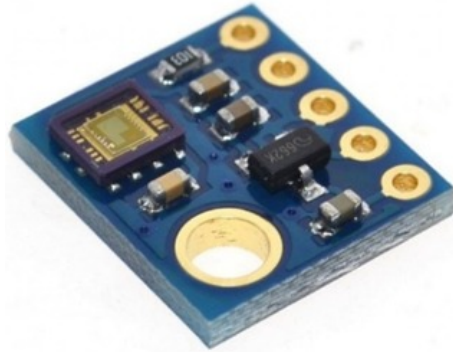


Figura 2.5: Sensor ML8511 [41]

Características:

- Rango de medición de radiación UV-A: 280 nm a 320 nm.
- Rango de medición de radiación UV-B: 320 nm a 390 nm.
- La salida analógica está relacionada linealmente con la intensidad **UV** (mW/cm²).
- Voltaje de alimentación: 3.3V a 5V.
- Consumo de corriente : 1 mA.
- Tipo de salida: analógica.
- Salida de voltaje analógica lineal: 1V a 3V.
- Consumo ultrabajo de energía.
- Dimensiones aproximadas: 13.5x12 mm [42].

2.2.5. Elevador de voltaje MT3608

El módulo DC-DC elevador Step Up MT3608 es un dispositivo regulable que convierte una fuente de voltaje más bajo en uno más alto. Con capacidad de hasta 2A,

utiliza el chip MT3608 para regular eficientemente el voltaje de salida. Este módulo es ajustable mediante un potenciómetro que permite establecer la salida deseada. Sus aplicaciones comunes incluyen alimentar dispositivos electrónicos con tensiones mayores a partir de fuentes de baja tensión, como baterías. Compacto y versátil, el MT3608 es útil para proyectos electrónicos donde se requiere un voltaje elevado de manera eficiente y controlada [43].



Figura 2.6: Módulo DC-DC Elevador Step Up regulable MT3608 2A [43]

Características:

- Corriente máxima de salida: 2A (Se recomienda no superar 1A para uso continuo).
- Voltaje de entrada: 2V a 24V.
- Voltaje máximo de salida: 28V.
- Eficiencia máxima: 93 %.
- Potenciómetro para ajustar el voltaje de salida.
- Dimensiones aproximadas: 37x17x14 mm [44].

2.2.6. Cargador Baterías Litio TP4056

Consta con conector USB Tipo C para alimentación, su chip principal, el TP4056, regula la corriente y voltaje durante la carga. Incorpora una protección contra sobrecarga

y sobredescarga, garantizando una carga segura, de igual forma indicadores LED sobre el estado de carga, este módulo facilita la carga eficiente y segura de baterías LiPo, se sigue las especificaciones del fabricante para un uso adecuado [45].



Figura 2.7: Módulo TP4056 [45]

Características:

- Módulo de carga lineal de baterías de litio con protección dual.
- Precisión de carga: 1.5 %.
- Voltaje de entrada: 4.5 a 5.5 V.
- Voltaje de carga completa: 4.2 V.
- Corriente máxima: 1A.
- Temperatura de funcionamiento: -10 °C a 85 °C.
- Tamaño aproximado: 25x19x10 mm [46].

2.2.7. Batería Recargable 18650

La batería recargable 18650 es un dispositivo de almacenamiento de energía, comúnmente utilizada en dispositivos electrónicos portátiles. Conformada por celdas de iones de litio, ofrece una corriente estable y alta densidad energética [47]. Su recargabilidad la hace económica y ecológica. Sin embargo, requiere cargadores

adecuados y cuidado para un uso seguro [45]. Su popularidad radica en su capacidad de proporcionar una fuente de energía confiable y duradera en aplicaciones diversas.



Figura 2.8: Batería 18650 [47]

Características:

- Voltaje: 3.7 V.
- Capacidad nominal: 2000 mAh.
- Peso: 43 gramos.
- Dimensiones máximas: 65x18 mm.
- Tasa de descarga reducida.
- Temperatura de uso: 0 a 60 °C.
- Vida útil: 10 años.
- Material: iones de litio [47].

2.2.8. Panel Fotovoltaico

Es un dispositivo que transmuta mediante células fotovoltaicas la energía solar en electricidad. Estas células están dispuestas en un módulo, creando un panel. Ofreciendo eficiencia y versatilidad en la generación de energía renovable, contribuyendo así a un enfoque más sostenible para abastecer a las necesidades energéticas requeridas [48].



Figura 2.9: Panel solar [48]

Características:

- Tensión nominal máxima: 5V.
- Corriente nominal máxima: 0-80 mA.
- Potencia máxima: 0,75 W.
- Tamaño: 99x69 mm.
- Grosor: 2,5 mm.
- Vida útil: 5-10 años.
- Material: policristalino [48].

2.3. Huerto agrícola

En la comunidad de Chicaguiña el cantón Gualaceo existe un huerto agrícola donde se produce diversas variedades de hortalizas para la comercialización local. Como se puede observar en la figura 2.10 se aprecia la ubicación del huerto agrícola.

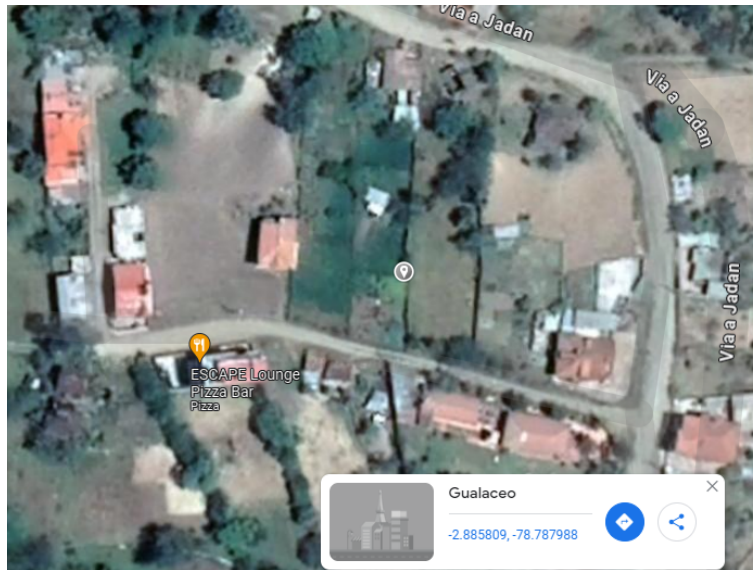


Figura 2.10: Localización del huerto agrícola [Fuente: Google Maps]

Se seleccionó este sitio específico para llevar a cabo las pruebas de los dispositivos debido a su diversidad en secciones con cultivos de hortalizas como se puede ver en la figura 2.11. Además, se eligió por su ubicación remota, coincidiendo con la ubicación prevista del receptor, con el fin de verificar la funcionalidad de intercambio de datos en la zona.



Figura 2.11: Huerto agrícola casero [Fuente: Autor]

2.4. Diagrama de comunicaciones

2.4.1. Diagrama de comunicación LoRa

Para iniciar la interacción entre el transmisor y el receptor, la topología adecuada para este caso es el Punto-a-Punto como se muestra en la imagen 2.12. En el cual, el nodo recopila la información de las variables ambientales mediante los sensores y la transmite al receptor a través de la comunicación inalámbrica con tecnología LoRa, funcionando a 915 MHz frecuencia conforme a las regulaciones de Ecuador. Posteriormente, esta información se dirige a la página de Grafana para la creación de paneles interactivos y gráficos, utilizando la información almacenada desde InfluxDB.

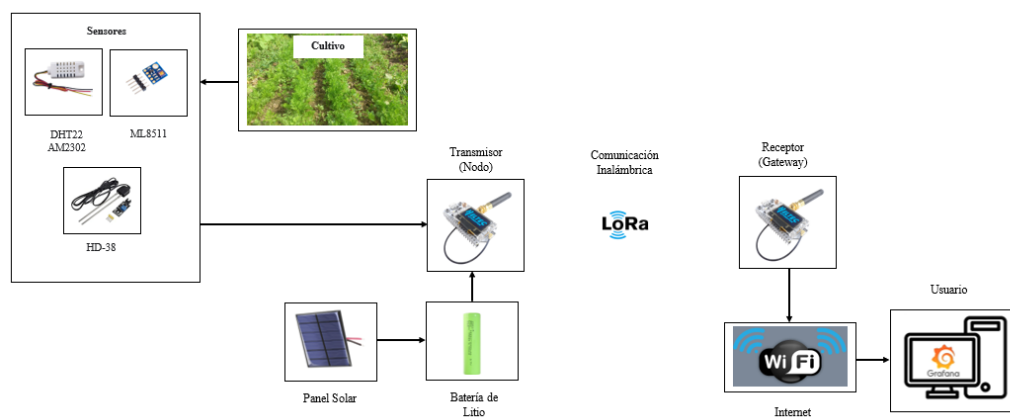


Figura 2.12: Sistema de comunicación inalámbrica de la red [Fuente: Autor]

2.4.2. Comunicación del ESP32 LoRa Heltec V2 hacia los sensores

El ESP32 Lora Heltec V2 se responsabiliza de recolectar la información captada por los sensores, donde se comunican con el microcontrolador a través de los pines analógicas y digitales tal como se evidencia en la figura 2.13. Los datos capturados se almacenan en variables específicas, ya sea como enteros o vectores de bytes, dependiendo de los requerimientos de cada sensor. Posteriormente, estos datos son preparados para su transmisión a través de la red LoRa, facilitando su monitoreo remoto y la visualización de los datos de las variables ambientales.

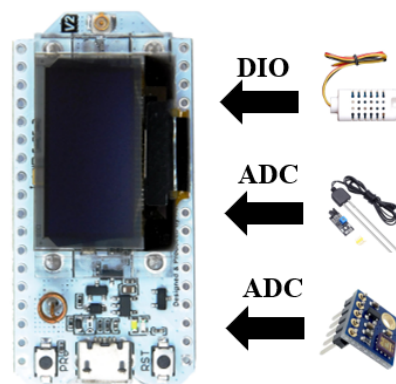


Figura 2.13: Comunicación del ESP32 LoRa Heltec V2 hacia los sensores [Fuente: Autor]

2.4.3. Suministro energético mediante energía solar

Se implementó un sistema solar fotovoltaico de la figura 2.14 para suministrar energía a diversos módulos y sensores. Esto alimenta a una batería de ion de litio de 3.7 voltios, la cual a su vez está conectada a un elevador de voltaje que proporciona una salida de 5V.

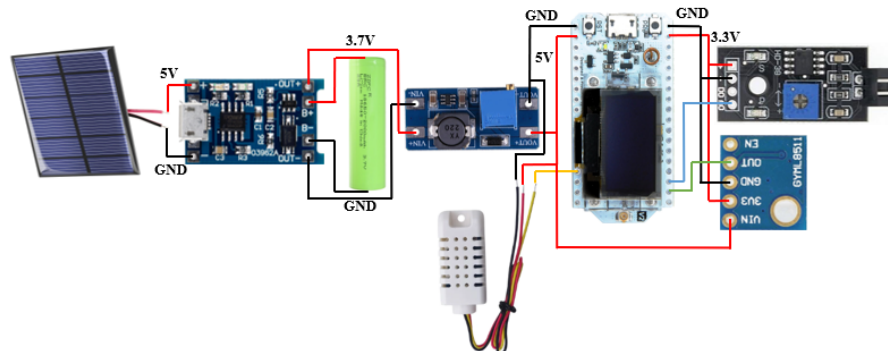


Figura 2.14: Esquema de suministro eléctrico mediante energía solar [Fuente: Autor]

2.5. Placa PCB

El diseño del circuito para el prototipo se realizó mediante el uso del software KiCad 6.0. En este caso, se organiza las tarjetas de programación, módulos y sensores. El objetivo era establecer las interconexiones requeridas entre los sensores y el microcontrolador. La figura 2.15 ilustra el montaje de las pistas para la interconexión de

los componentes electrónicos a utilizar en el circuito.

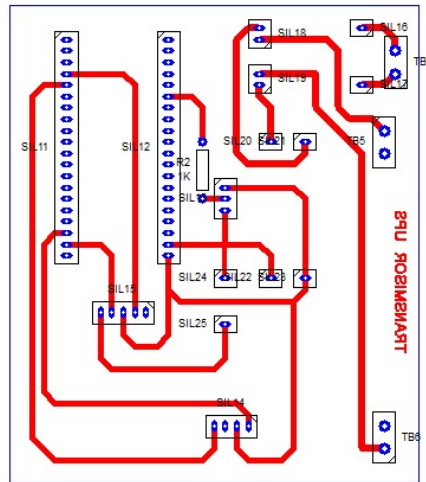


Figura 2.15: Vista 2D de la placa [Fuente: Autor]

La placa desarrollada ha sido configurada con dimensiones adecuadas para acomodar los componentes electrónicos esenciales, esto se puede ver reflejado en la ilustración 2.16.

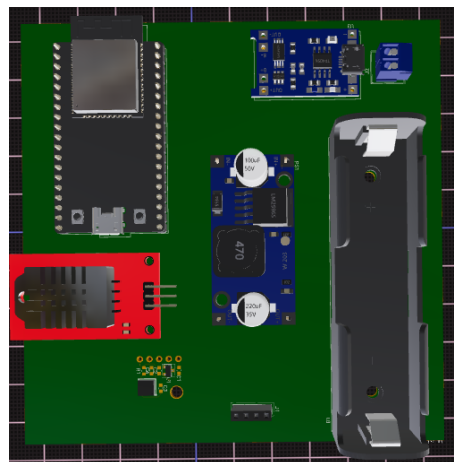


Figura 2.16: Vista 3D de la placa [Fuente: Autor]

2.6. Diseño y Fabricación de protección para dispositivos

Se creó un diseño en el software Fusion360 para fabricar en impresión 3D una caja destinada a la protección tanto del dispositivo transmisor como del receptor de la placa. Se consideró al momento del diseñar el tamaño y posición de la placa PCB, panel solar,

batería, un orificio para la antena del modulo LoRa y un soporte en la parte posterior para la colocación en una superficie sólida. Se puede evidenciar en la imagen 2.17.

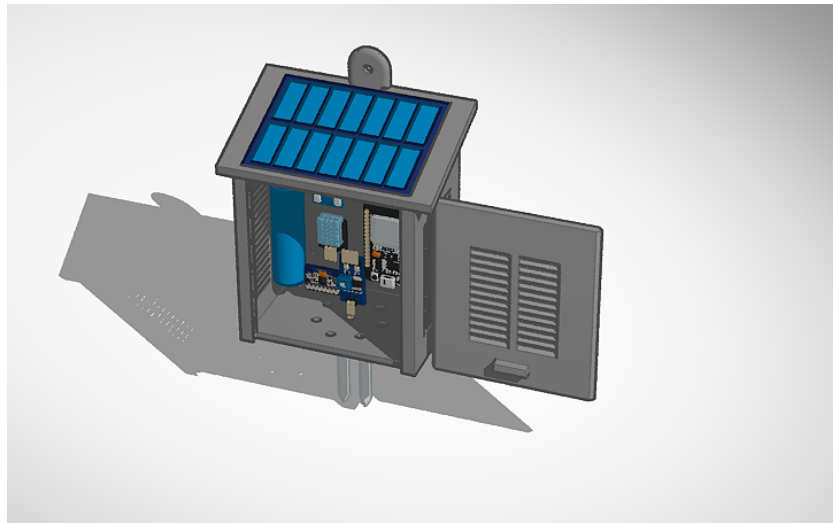


Figura 2.17: Estructura para el dispositivo transmisor [Fuente: Autor]

Igualmente, se creó un diseño para resguardar el dispositivo receptor. La imagen 2.18 proporciona una vista previa de la caja, destacando la presencia de una abertura específica para la antena LoRa. De este modo, la caja funciona como un nodo de comunicación que establecerá conexión con el receptor.

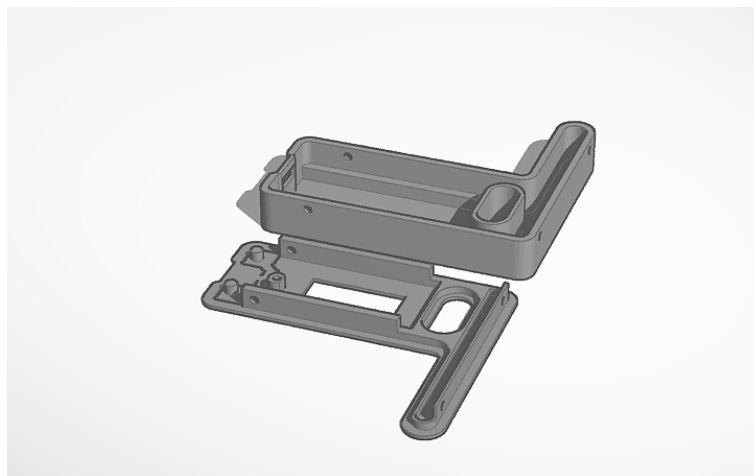


Figura 2.18: Estructura para el receptor [Fuente: Autor]

Capítulo 3

Implementación y Análisis de Resultados

3.1. Diseño del dispositivo electrónico

3.1.1. Interconexión de los componentes mediante una placa de pruebas

La identificación de los puertos la placa Heltec WiFi LoRa 32(V2) se visualiza en la placa misma o en la documentación del fabricante. Los puertos son etiquetados con nombres específicos para facilitar su uso y conexión como se puede ver en la figura 3.1.

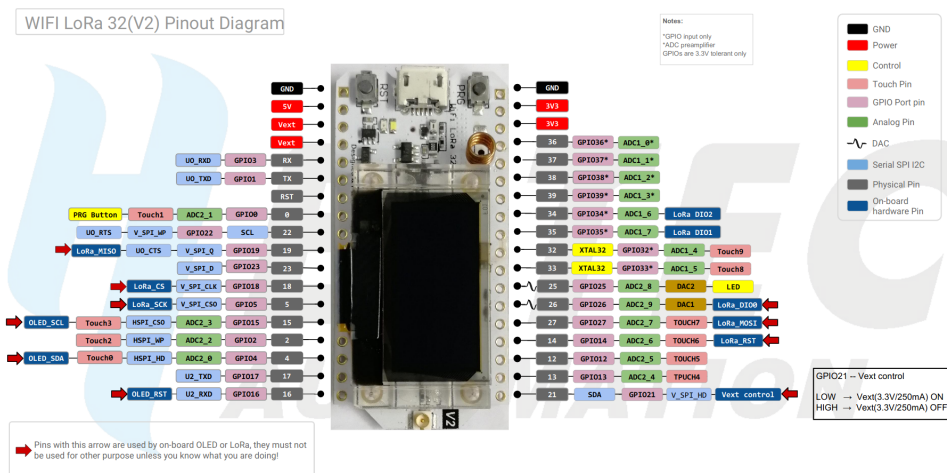


Figura 3.1: Identificación de puertos del microcontrolador [35]

La conexión de los sensores de la figura 3.2 y comprobación del funcionamiento se realizó en una placa de pruebas (protoboard), en primera instancia se define los puertos del microcontrolador a cada sensor y modulo que se va a utilizar.

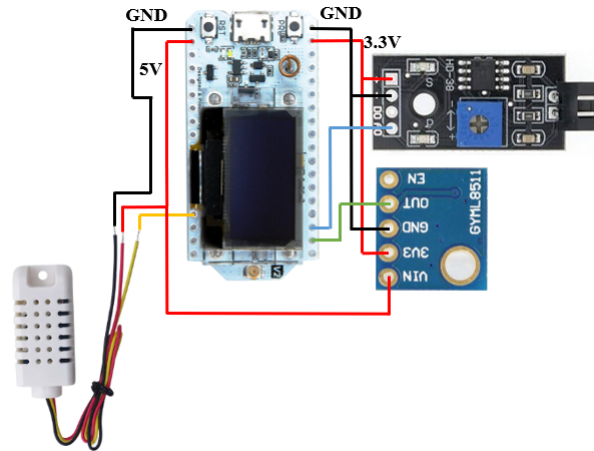


Figura 3.2: Esquema de conexiones de los sensores con el módulo Heltec WiFi LoRa 32(V2) [Fuente: Autor]

Una vez definido los pines de conexión se procede en el armado del circuito en la protoboard como se muestra en la figura 3.3, para la comprobación del funcionamiento de los sensores con la programación, hay que recalcar que para la parte de alimentación se utilizó un sistema solar fotovoltaico de 5V de la figura 2.14.

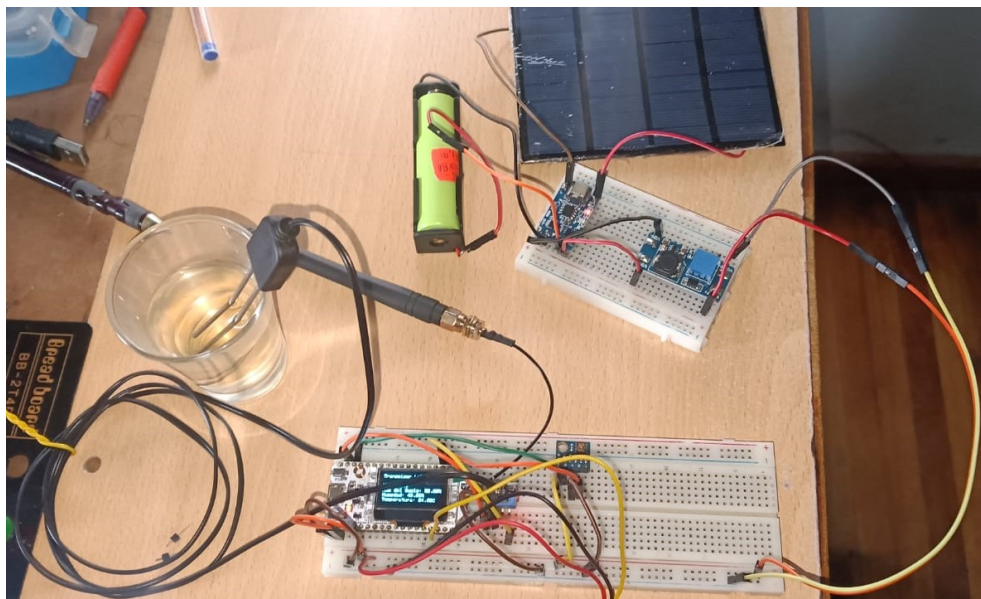


Figura 3.3: Conexión de dispositivos [Fuente: Autor]

Fuente de alimentación

La utilización de una batería de litio de 3.7 V con alimentación solar es crucial en aplicaciones remotas debido a su eficiencia, esta combinación ofrece una fuente de energía sostenible y renovable. Esto permite la implementación de sistemas autónomos y ecoamigables en áreas remotas.

Para calcular el tiempo de operación de una batería de litio, se utiliza la fórmula de Digital Key Electronics [49]:

$$\text{Tiempo de Operación (horas)} = \frac{\text{Capacidad de la Batería (mAh)}}{\text{Corriente de Consumo del Dispositivo (mA)}} \quad (3.1)$$

Los parámetros de esta fórmula son:

- **Energía almacenada de la batería:** Este valor representa el potencial energético acumulado de la batería, expresada en miliamperios-hora (mAh).
- **Corriente de Consumo del Dispositivo:** Este parámetro indica la corriente eléctrica que el dispositivo consume mientras está en funcionamiento, medido en miliamperios (mA).

Se seleccionó una batería de 3.7 V y 2000 mA para alimentar el dispositivo transmisor, basándose en las especificaciones de los módulos y sensores. Se tomó en cuenta la corriente de consumo, la cual se estima en 104 mA, según la datos específicos en la tabla 3.1.

Tabla 3.1: Corriente de consumo de los componentes electrónicos [Fuente: Autor]

Componente	Consumo de Corriente
Sensor Dht22	3 mA
Sensor HD-38	30 mA
Sensor ML8511	1 mA
Heltec LoRa Wifi 32 (V2)	70 mA

$$\text{Tiempo de Operación (horas)} = \frac{2000 \text{ mAh}}{104 \text{ mA}} \approx 19,23 \text{ horas} \quad (3.2)$$

Este cálculo proporciona una estimación del tiempo durante el cual la batería podría alimentar el dispositivo con una corriente de consumo constante de 104mA. Es fundamental considerar que esta es una aproximación simplificada y que la vida útil real puede variar debido a factores como la descarga no lineal de la batería, cambios en la corriente de consumo y la degradación natural de la batería con el tiempo.

3.1.2. Diagrama de flujo del dispositivo transmisor

En la figura 3.4 se observa el gráfico de procesos del dispositivo transmisor especificando cada paso para la realización del código de programación para la obtención de la información de los sensores, establecer la comunicación inalámbrica con tecnología LoRa para la transmisión de datos y visualización tanto en el monitor serial como en la pantalla OLED.

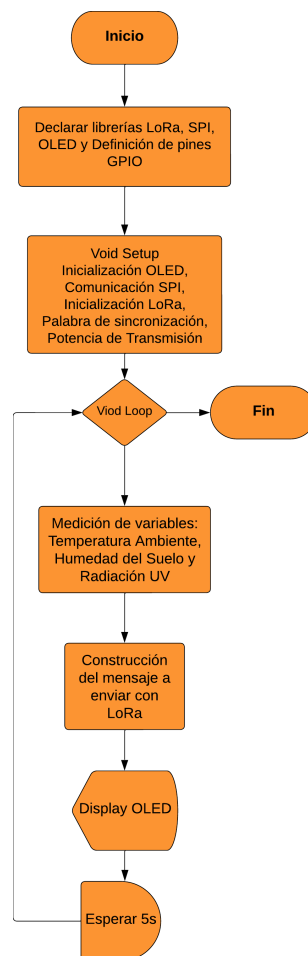


Figura 3.4: Lógica de programación del dispositivo transmisor [Fuente: Autor]

3.1.3. Programación del sensor DHT22

Para usar el sensor DHT22 en Arduino IDE implica incluir la biblioteca DHT, configurar el pin de conexión y definir el modelo del sensor como se muestra en la figura 3.5.

```
//Libreria para el sensor DHT22
#include <DHT.h>
#define DHTTYPE DHT22 // Tipo de sensor DHT
DHT dht(2, DHTTYPE); // Identificación del tipo de sensor
```

Figura 3.5: Librería del sensor DHT [Fuente: IDE Arduino]

En el bucle principal, ejecuta la función de lectura la temperatura y humedad. Si cumple la condición de la lectura del sensor, los valores se imprimen en la ventana del monitor serial, la función encapsula la lógica de lectura y manejo de errores como se observa en la figura 3.6.

```
// Medir la temperatura y la humedad con el sensor DHT22
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```

Figura 3.6: Medición de la temperatura y humedad [Fuente: IDE Arduino]

3.1.4. Programación del sensor HD-38

Las líneas de código de la figura 3.7 se realiza la medición del sensor HD-38 mediante sus dos electrodos metálicos. Donde envía una señal analógica que varía según la resistencia del suelo y se ajusta los valores para expresar en un rango del 0 al 100, proporcionando así una representación proporcional.

```
// Medir la humedad del suelo
int soilHumidityValue = analogRead(SOIL_HUMIDITY_PIN);
float soilHumidity = map(soilHumidityValue, 4095, 0, 0, 100);
```

Figura 3.7: Medición del porcentaje de humedad del suelo [Fuente: IDE Arduino]

3.1.5. Programación del sensor ML85-11

Las líneas de código de la figura 3.8 mide el nivel de radiación UV a través de un pin analógico para calcular el voltaje de salida proporcional a la intensidad y luego lo mapea a un rango específico para obtener la intensidad UV en términos de niveles de 1.0 a 15.0.

```
// Medir el nivel de radiación UV
int uvLevel = averageAnalogRead(uvSensorPin);
int refLevel = averageAnalogRead(REF_3V3);
float outputVoltage = 3.3 / refLevel * uvLevel;
//Convierta el voltaje a un nivel de intensidad UV.
float uvIntensity = mapfloat(outputVoltage, 0.87, 3.2, 1.0, 15.0);
```

Figura 3.8: Medición de la radiación ultravioleta (UV) [Fuente: IDE Arduino]

3.2. Comunicación inalámbrica con tecnología LoRa

3.2.1. Selección de la topología de comunicación

La topología punto a punto entre el emisor y receptor se puede ver en la figura 3.9, donde se destaca por su eficiencia y simplicidad.

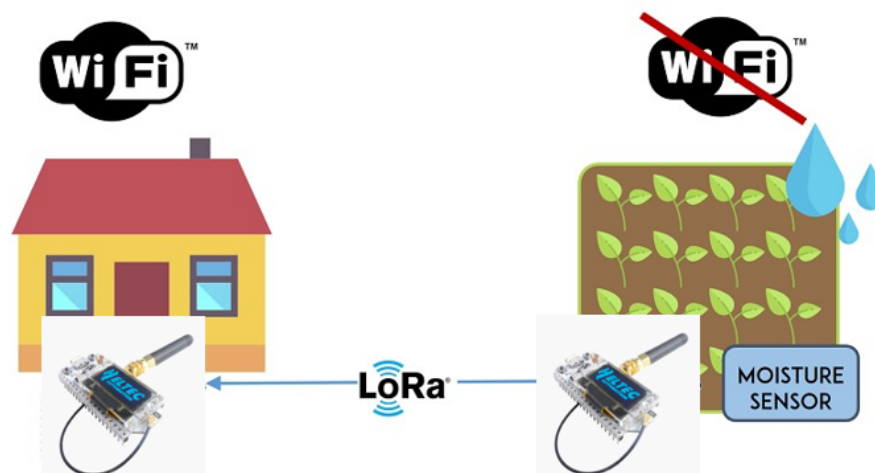


Figura 3.9: Topología P2P [24]

En este enfoque, ambos dispositivos establecen una conexión directa para intercambiar datos de manera unidireccional, el transmisor equipado con sensores de

temperatura, humedad del suelo y radiación UV, transmite información al receptor que utiliza un único microcontrolador para recibir y registrar los datos sin depender de conexiones adicionales.

3.2.2. Programación del dispositivo transmisor

La figura 3.10 se describe el código del dispositivo transmisor, primero se debe incorporar bibliotecas esenciales para gestionar el módulo LoRa y para la interacción con una pantalla OLED integrada. También se debe definir los pines específicos de conexión con el módulo LoRa, como SCK, MISO, MOSI, SS, RST y DIO0. Además, establecer la frecuencia de operación en 915 MHz, adaptándose a la ubicación del proyecto. Estas configuraciones y librerías proporcionan la base necesaria para la implementación de la comunicación inalámbrica y la interfaz con la pantalla OLED.

```
//Librerías para el modulo LoRa
#include <LoRa.h>
#include <SPI.h>

//Libraries para comunicar y dibujar en la pantalla OLED integrada
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//Debemos definir los pines que seran utilizadas por el modulo LoRa
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

//Definimos la frecuencia de operacion segun nuestra ubicacion
#define BAND 915E6
```

Figura 3.10: Configuraciones básicas de comunicación del dispositivo transmisor [Fuente: IDE Arduino]

En la figura 3.11 se configuran inicialmente en el **void setup()** la comunicación SPI del modulo LoRa y de la interfaz con la pantalla OLED. Luego, se intenta iniciar la comunicación en la frecuencia de banda seleccionada. Si surge algún problema, se emite un mensaje de error causando la interrupción del programa. En caso de una inicialización exitosa, se muestra un mensaje positivo tanto en el puerto serial como en la pantalla

OLED, seguido de una breve espera de dos segundos.

```
SPI.begin(SCK, MISO, MOSI, SS); //Definimos pines para la comunicacion SPI
LoRa.setPins(SS, RST, DIO0); //Configuramos el LoRa para enviar

if (!LoRa.begin(BAND)) { //Intenta transmitir en la banda seleccionada
  Serial.println("Error iniciando LoRa"); //Si no puede transmitir, marca error en la inicializacion
  while (1);
}

Serial.println("Inicio exitoso de LoRa!"); //Mensaje de todo bien en puerto serial
display.setCursor(0,10);
display.print("Inicio exitoso de LoRa!"); //Mensaje de todo bien en pantalla OLED
display.display();
delay(2000); //Esperamos un par de segundos
```

Figura 3.11: Inicialización de la comunicación LoRa y Pantalla OLED [Fuente: IDE Arduino]

Establece de igual forma la palabra de sincronización para la comunicación LoRa en 0xF3 y configura la potencia de transmisión del módulo LoRa a 20 dBm como se muestra en la figura 3.12. Estas configuraciones son cruciales para garantizar la sincronización y la eficacia de la transmisión.

```
// Configura la palabra de sincronización para la comunicación LoRa
LoRa.setSyncWord(0xF3);
// Configura la potencia de transmisión del módulo LoRa a 20 dBm
LoRa.setTxPower(20);
```

Figura 3.12: Palabra de sincronización [Fuente: IDE Arduino]

En las líneas de programación de la figura 3.13, se crea un mensaje que incorpora datos de varios sensores, como la temperatura ambiente, radiación UV, humedad del suelo. Se utiliza un array de bytes llamado **messageData** para almacenar estos datos mediante la función **memcpy** en el **void loop()**. Posteriormente, se utiliza la comunicación LoRa para iniciar y enviar el mensaje construido.

```
// Construir el mensaje a enviar
byte messageData[24];
memcpy(&messageData[0], &uvVoltage, sizeof(float));
memcpy(&messageData[4], &soilHumidity, sizeof(float));
memcpy(&messageData[8], &temperature, sizeof(float));
memcpy(&messageData[12], &humidity, sizeof(float));

// Enviar el mensaje
LoRa.beginPacket();
LoRa.write(messageData, sizeof(messageData));
LoRa.endPacket();
```

Figura 3.13: Construcción y envío de datos de los sensores [Fuente: IDE Arduino]

3.2.3. Diagrama de flujo del ESP32 Heltec LoRa (Receptor)

En la figura 3.14 se muestra un esquema de flujo del dispositivo receptor como Gateway LoRa especificando cada paso para la realización de la lógica de programación para establecer la comunicación con tecnología LoRa de manera inalámbrica, para recibir la información de los sensores, seguido de la conexión a la red WiFi y el servidor MQTT, y finaliza con un bucle para la recepción de información.

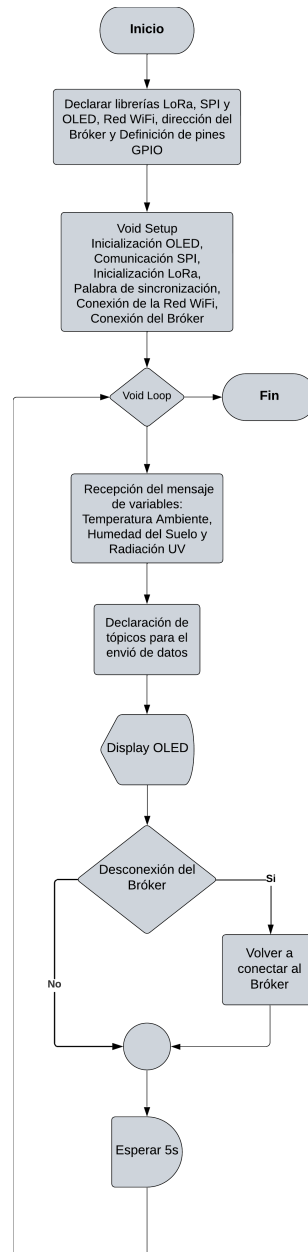


Figura 3.14: Lógica de programación del dispositivo receptor [Fuente: Autor]

3.2.4. Programación del dispositivo receptor como Gateway LoRa

En la programación del dispositivo receptor, se implementó las librerías para establecer la conectividad WiFi y para implementar funcionalidades MQTT en el programa, seguido de las librerías del módulo LoRa, interfaz OLED y establecimiento de la frecuencia de operación en 915 MHz como se puede ver en la figura 3.15.

```
//Librerías para el módulo LoRa
#include <LoRa.h>
#include <SPI.h>

//Librerías para comunicar y dibujar en la pantalla OLED integrada
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//Configuración de Conexión WiFi y Cliente MQTT
#include <WiFi.h>
#include <PubSubClient.h>

//Debemos definir los pines que serán utilizadas por el módulo LoRa
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

//Definimos la frecuencia de operación según nuestra ubicación
#define BAND 915E6
```

Figura 3.15: Configuraciones básicas de comunicación del dispositivo receptor [Fuente: IDE Arduino]

Adicionalmente, en la inicialización del programa se configuran los parámetros esenciales para MQTT, como las credenciales de la red WiFi (SSID y contraseña) y el servidor MQTT al que el cliente se conectará como se muestra en la figura 3.16, facilitando la conexión para implementar la función para capturar y analizar información.

```
// Configuración de MQTT

// Definir el nombre de la red WiFi y la contraseña
const char* ssid = "Mario Penaranda";
const char* password = "santabarbara_64";

// Definir la dirección del servidor MQTT
const char* mqtt_server = "broker.emqx.io";

// Crear una instancia de cliente WiFi
WiFiClient espClient;

// Crear una instancia de cliente MQTT utilizando la biblioteca PubSubClient
PubSubClient client(espClient);
```

Figura 3.16: Conexión WiFi y cliente MQTT [Fuente: IDE Arduino]

En este caso en el receptor, el código configura la palabra de sincronización para la comunicación LoRa en 0xF3 como se puede ver en la figura 3.17. Esta configuración garantiza que el receptor esté sincronizado correctamente con el transmisor para una comunicación efectiva a través de LoRa.

```
// Configura la palabra de sincronización para la comunicación LoRa
LoRa.setSyncWord(0xF3);
```

Figura 3.17: Palabra de sincronización (Receptor) [Fuente: IDE Arduino]

En esta sección de código de la figura 3.18, se inicia la comunicación SPI y se configuran los pines para el módulo LoRa en la función **void setup()**. Luego, se establece la comunicación LoRa en la frecuencia de banda específica. En caso de que ocurra un fallo al momento de inicializar el módulo, se emite una notificación de fallo causando la interrupción del código. Si la inicialización es exitosa, se imprime un mensaje de confirmación en el puerto serial y en una pantalla OLED. Estas acciones garantizan una configuración adecuada y exitosa del módulo LoRa para la transmisión inalámbrica en el dispositivo.

```
// Inicia la comunicación SPI con pines específicos
SPI.begin(SCK, MISO, MOSI, SS);

// Configura los pines para el módulo LoRa
LoRa.setPins(SS, RST, DIO0);

// Intenta iniciar la comunicación LoRa en la frecuencia de banda especificada
if (!LoRa.begin(BAND)) {
  Serial.println("Error iniciando LoRa");
  while (1);
}

// Imprime un mensaje de inicio exitoso en el puerto serial
Serial.println("Inicio exitoso de LoRa!");
// Configura la posición del cursor en la pantalla OLED
display.setCursor(0, 30);
// Imprime un mensaje de inicio exitoso en la pantalla OLED
display.print("Inicio exitoso de LoRa!");
// Muestra el contenido de la pantalla OLED
display.display();
delay(2000);
```

Figura 3.18: Inicialización y configuración del módulo LoRa [Fuente: IDE Arduino]

Seguido en el **void setup()** la conexión WiFi se inicializa a través de la función **setup_wifi()**. Luego, se configura la conexión MQTT, especificando el servidor y el puerto, asignando una función de retorno de llamada **callback**. Y por último, se intenta reconectar al servidor MQTT para asegurar una conexión estable y efectiva como se muestra en la figura 3.19.

```
// Inicialización de la conexión WiFi
setup_wifi();

// Inicialización de la conexión MQTT
client.setServer(mqtt_server, 1883); // Establece el servidor MQTT y el puerto
client.setCallback(callback); // Asigna la función de retorno de llamada
reconnect(); // Intenta reconectar al servidor MQTT
```

Figura 3.19: Parametrización de conexión WiFi y MQTT [Fuente: IDE Arduino]

En la función **void loop()**, se tiene que verificar la presencia de un paquete LoRa recibido y, en caso de que sea así, se procesan los datos contenidos. Se crea un array de bytes para almacenar la información y se lee el paquete en su totalidad. Los valores específicos, se los sensores, se extraen del array. Además, se obtiene la intensidad de señal RSSI del paquete LoRa. Finalmente, se publican los datos en los tópicos MQTT deseados, facilitando la transmisión de información a otros dispositivos o sistemas, el código de todas estas acciones se muestra en la figura 3.20.

```
// Obtiene el tamaño del paquete recibido a través de LoRa
int tamañoPaquete = LoRa.parsePacket();

if (tamañoPaquete) {
  Serial.print("Paquete recibido ");
  // Ajusta el tamaño del array según la longitud de tus datos
  byte messageData[24];
  while (LoRa.available()) {
    LoRa.readBytes(messageData, sizeof(messageData));
  }

  // Extrae los valores del array de bytes
  float uvVoltage, soilHumidity, temperature, humidity;
  // Copia los datos del array de bytes a las variables correspondientes
  memcpy(&uvVoltage, &messageData[0], sizeof(float));
  memcpy(&soilHumidity, &messageData[4], sizeof(float));
  memcpy(&temperature, &messageData[8], sizeof(float));
  memcpy(&humidity, &messageData[12], sizeof(float));

  // Obtiene el valor de intensidad de señal RSSI del paquete LoRa
  int rssi = LoRa.packetRssi();
  Serial.print(" con RSSI ");
  Serial.println(rssi);

  // Publica los datos en los tópicos MQTT deseados
  publishData(temperature, humidity, uvVoltage, soilHumidity);
}
```

Figura 3.20: Procesamiento de datos desde paquetes LoRa y publicación en MQTT [Fuente: IDE Arduino]

Seguido se tiene que verifica si el cliente MQTT no está conectado y, en caso afirmativo, se procede a reconectar utilizando la función `reconnect()`. Posteriormente, se manejan los mensajes entrantes mediante la función `client.loop()`. Este fragmento de código de la figura 3.21 asegura una conexión persistente al servidor y permite gestionar de manera continua los mensajes entrantes, garantizando así una comunicación estable y efectiva con el servidor MQTT.

```
// Realiza la conexión MQTT si es necesario
if (!client.connected()) {
  reconnect();
}

// Maneja los mensajes MQTT entrantes
client.loop();
```

Figura 3.21: Gestión continua de conexión y mensajes MQTT [Fuente: IDE Arduino]

En la función `void publishData()` de la figura 3.22, se implementa la publicación de datos asociados a la temperatura, humedad, voltaje UV y humedad del suelo en tópicos específicos. Cada variable se envía de manera individualizada, estructurando la información para su seguimiento remoto. Esta estrategia agiliza la transmisión de datos en un contexto del IoT, mejorando la eficiencia en la comunicación sin enfatizar demasiado en el protocolo MQTT.

```
void publishData(float temperature, float humidity, float uvVoltage, float soilHumidity) {
  // Publica cada variable por separado en su tópico MQTT correspondiente
  client.publish("temperature_topic", String(temperature).c_str());
  client.publish("humidity_topic", String(humidity).c_str());
  client.publish("uv_topic", String(uvVoltage).c_str());
  client.publish("soil_humidity_topic", String(soilHumidity).c_str());
}
```

Figura 3.22: Publicación eficiente de datos en tópicos específicos [Fuente: IDE Arduino]

En la función `setup_wifi()` de la figura 3.23, se da inicio al procedimiento de conexión a la red WiFi utilizando las credenciales proporcionadas (SSID y contraseña). Mientras espera que se establezca la conexión, se imprime un indicador visual en el puerto serial. Una vez conectado con éxito, se imprime la dirección IP asignada al dispositivo, proporcionando información clave sobre la conexión WiFi.

```
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Conectando a WiFi");

  // Inicia el proceso de conexión a la red WiFi
  WiFi.begin(ssid, password);

  // Espera hasta que se establezca la conexión
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Imprime la información de la conexión exitosa
  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("Dirección IP: ");
  Serial.println(WiFi.localIP());
}
```

Figura 3.23: Configuración y conexión a la red WiFi [Fuente: IDE Arduino]

Por último, en la función **void reconnect()** de la figura 3.24, intenta reconectar el dispositivo al servidor MQTT. Se utiliza un bucle que continúa hasta que la conexión es exitosa. Durante cada intento, genera un ID de cliente único, intenta conectarse y verifica el estado de la conexión. Si la conexión falla, muestra el código de retorno (rc) y espera 5 segundos antes de intentar nuevamente, el objetivo es mantener una conexión persistente al servidor.

```
void reconnect() {
  // Intenta conectar al servidor MQTT
  while (!client.connected()) {
    Serial.print("Intentando conexión MQTT...");

    // Genera un ID de cliente único
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);

    // Intenta conectar con el ID de cliente generado
    if (client.connect(clientId.c_str())) {
      Serial.println("Conectado al servidor MQTT");
    } else {
      Serial.print("falló, rc=");
      Serial.print(client.state());
      Serial.println(" intentando de nuevo en 5 segundos");
      delay(5000);
    }
  }
}
```

Figura 3.24: Reconexión al servidor [Fuente: IDE Arduino]

3.2.5. Envío de datos

El dispositivo transmisor mediante comunicación LoRa, se establece una conexión persistente mediante un bucle de reconexión. Este enfoque simplificado garantiza el envío eficiente de información a través de los módulos LoRa, transmitiendo así las variables climáticas del entorno y visualizadas en la pantalla como se puede ver en la figura 3.25.

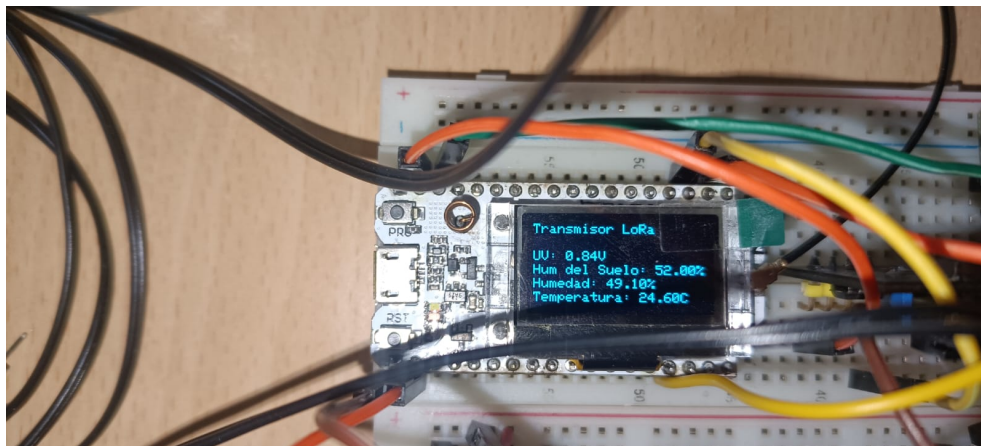


Figura 3.25: Envío de información de la variables [Fuente: Autor]

3.2.6. Recepción de datos

El dispositivo receptor busca mantener de igual forma una conexión constante mediante un bucle de reconexión. Al recibir datos, estos se procesan y luego se publican en tópicos específicos, facilitando la monitorización remota como se puede ver en la figura 3.26.



Figura 3.26: Modulo receptor [Fuente: Autor]

El enfoque de la reconexión asegura una transmisión efectiva de información, priorizando la conexión a la red WiFi con la dirección IP y la publicación de datos en tópicos al servidor MQTT esto se puede ver en la figura 3.27.

```

15:54:37.139 -> WiFi conectado
15:54:37.139 -> Dirección IP:
15:54:37.171 -> 192.168.88.105
15:54:37.171 -> Intentando conexión MQTT...Conectado al servidor MQTT
15:54:37.918 -> Received message: Temperature: 27.30 °C - Humidity: 41.30 % - UV Voltage: 2.92 mW/cm^2 - Soil Humidity: 0.00 %
15:54:38.088 -> Received message: Temperature: 27.40 °C - Humidity: 41.90 % - UV Voltage: 3.22 mW/cm^2 - Soil Humidity: 0.00 %
15:54:40.618 -> Received message: Temperature: 27.50 °C - Humidity: 41.60 % - UV Voltage: 2.68 mW/cm^2 - Soil Humidity: 0.00 %
15:54:43.145 -> Received message: Temperature: 27.50 °C - Humidity: 40.80 % - UV Voltage: 2.56 mW/cm^2 - Soil Humidity: 0.00 %
15:54:45.702 -> Received message: Temperature: 27.50 °C - Humidity: 39.90 % - UV Voltage: 2.89 mW/cm^2 - Soil Humidity: 0.00 %
15:54:48.197 -> Received message: Temperature: 27.50 °C - Humidity: 39.40 % - UV Voltage: 3.02 mW/cm^2 - Soil Humidity: 0.00 %
15:54:50.756 -> Received message: Temperature: 27.50 °C - Humidity: 39.00 % - UV Voltage: 2.42 mW/cm^2 - Soil Humidity: 0.00 %
15:54:53.295 -> Received message: Temperature: 27.50 °C - Humidity: 38.60 % - UV Voltage: 2.77 mW/cm^2 - Soil Humidity: 0.00 %
15:54:55.811 -> Received message: Temperature: 27.60 °C - Humidity: 38.30 % - UV Voltage: 3.18 mW/cm^2 - Soil Humidity: 0.00 %
15:54:58.374 -> Received message: Temperature: 27.60 °C - Humidity: 38.10 % - UV Voltage: 3.08 mW/cm^2 - Soil Humidity: 88.00 %
15:55:00.888 -> Received message: Temperature: 27.60 °C - Humidity: 38.00 % - UV Voltage: 2.57 mW/cm^2 - Soil Humidity: 80.00 %
15:55:03.446 -> Received message: Temperature: 27.60 °C - Humidity: 37.90 % - UV Voltage: 3.46 mW/cm^2 - Soil Humidity: 78.00 %
15:55:05.982 -> Received message: Temperature: 27.60 °C - Humidity: 37.90 % - UV Voltage: 2.51 mW/cm^2 - Soil Humidity: 78.00 %
15:55:08.528 -> Received message: Temperature: 27.60 °C - Humidity: 37.80 % - UV Voltage: 2.64 mW/cm^2 - Soil Humidity: 75.00 %
15:55:11.046 -> Received message: Temperature: 27.70 °C - Humidity: 37.80 % - UV Voltage: 3.11 mW/cm^2 - Soil Humidity: 75.00 %
15:55:13.596 -> Received message: Temperature: 27.70 °C - Humidity: 37.60 % - UV Voltage: 2.57 mW/cm^2 - Soil Humidity: 74.00 %
15:55:16.142 -> Received message: Temperature: 27.70 °C - Humidity: 37.50 % - UV Voltage: 2.63 mW/cm^2 - Soil Humidity: 73.00 %
15:55:18.653 -> Received message: Temperature: 27.70 °C - Humidity: 37.40 % - UV Voltage: 2.77 mW/cm^2 - Soil Humidity: 73.00 %
15:55:21.173 -> Received message: Temperature: 27.80 °C - Humidity: 37.40 % - UV Voltage: 2.45 mW/cm^2 - Soil Humidity: 74.00 %
15:55:23.703 -> Received message: Temperature: 27.80 °C - Humidity: 37.40 % - UV Voltage: 2.45 mW/cm^2 - Soil Humidity: 72.00 %
15:55:26.269 -> Received message: Temperature: 27.70 °C - Humidity: 37.20 % - UV Voltage: 2.60 mW/cm^2 - Soil Humidity: 72.00 %
15:55:28.804 -> Received message: Temperature: 27.80 °C - Humidity: 37.20 % - UV Voltage: 3.03 mW/cm^2 - Soil Humidity: 72.00 %

```

Figura 3.27: Recepción y publicación de datos [Fuente: IDE Arduino]

3.3. Plataforma de visualización

3.3.1. Habilitación de los puertos de comunicación

Es un procedimiento esencial para permitir la interacción entre dispositivos y sistemas.

La habilitación de los puertos de comunicación de entrada es un paso esencial para permitir la recepción eficaz de datos en un sistema o red. Este proceso implica configurar el firewall para permitir la entrada de información a través de puertos específicos esto se puede ver en la figura 3.28. Esto es fundamental para garantizar una comunicación fluida y sin obstáculos, posibilitando la interacción eficiente entre dispositivos y sistemas conectados en una red.

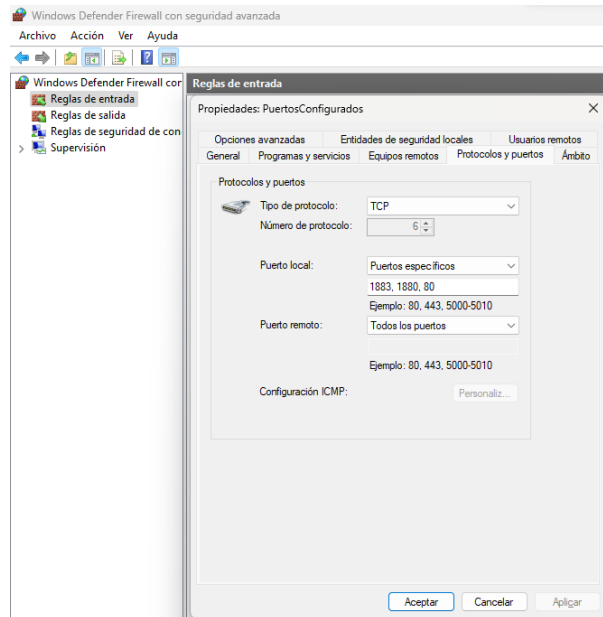


Figura 3.28: Configuración de los puertos de entrada [Fuente: Autor]

En cambio la activación de los puertos de comunicación de salida es esencial para permitir el envío eficiente de datos desde un sistema o red hacia destinos externos. De igual manera este proceso implica configurar el firewall para permitir la transmisión de información a través de puertos específicos esto se puede ver en la figura 3.29, posibilitando la interacción efectiva entre el sistema y otros servicios fuera de la red local.

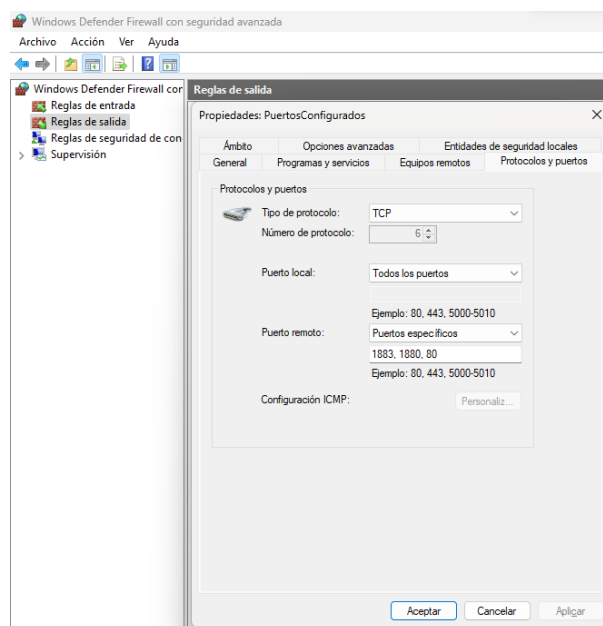
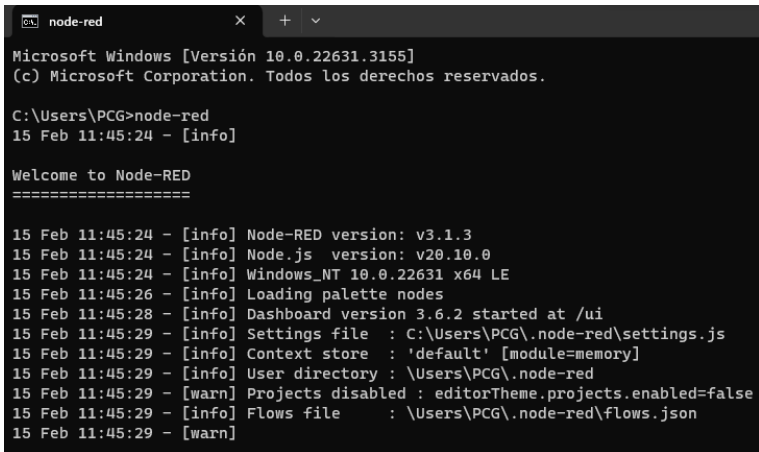


Figura 3.29: Configuración de los puertos de salida [Fuente: Autor]

3.3.2. Configuración de Node-RED y Broker MQTT

La habilitación de Node-RED mediante la línea de comandos (cmd) del la figura 3.30, se realiza para permitir su ejecución y configuración de manera sencilla. Este proceso implica acceder a la interfaz de línea de comandos y utilizar comandos específicos para iniciar Node-RED. Al ejecutar los comandos adecuados, se activa la plataforma, lo que facilita su utilización y personalización.



```
node-red
Microsoft Windows [Versión 10.0.22631.3155]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\PCG>node-red
15 Feb 11:45:24 - [info]

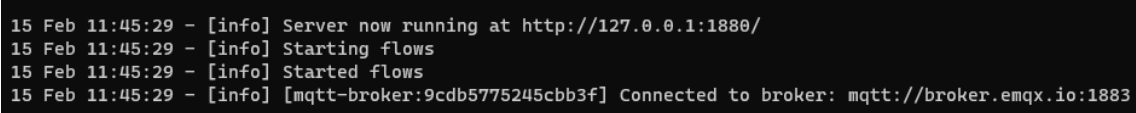
Welcome to Node-RED
=====

15 Feb 11:45:24 - [info] Node-RED version: v3.1.3
15 Feb 11:45:24 - [info] Node.js version: v20.10.0
15 Feb 11:45:24 - [info] Windows_NT 10.0.22631 x64 LE
15 Feb 11:45:26 - [info] Loading palette nodes
15 Feb 11:45:28 - [info] Dashboard version 3.6.2 started at /ui
15 Feb 11:45:29 - [info] Settings file : C:\Users\PCG\.node-red\settings.js
15 Feb 11:45:29 - [info] Context store : 'default' [module=memory]
15 Feb 11:45:29 - [info] User directory : \Users\PCG\.node-red
15 Feb 11:45:29 - [warn] Projects disabled : editorTheme.projects.enabled=false
15 Feb 11:45:29 - [info] Flows file : \Users\PCG\.node-red\flows.json
15 Feb 11:45:29 - [warn]
```

Figura 3.30: Inicialización de Node-RED [Fuente: Autor]

Esta acción es esencial para desplegar y aprovechar las capacidades para el desarrollo de flujos de trabajo, interfaces visuales, y gestión de aplicaciones IoT.

En la figura 3.31 se puede visualizar la conexión del broker MQTT desde la interfaz de línea de comandos (cmd) de Node-RED, estableciendo así una vinculación efectiva entre los dos servicios.



```
15 Feb 11:45:29 - [info] Server now running at http://127.0.0.1:1880/
15 Feb 11:45:29 - [info] Starting flows
15 Feb 11:45:29 - [info] Started flows
15 Feb 11:45:29 - [info] [mqtt-broker:9cdb5775245cbb3f] Connected to broker: mqtt://broker.emqx.io:1883
```

Figura 3.31: Conexión de Node-RED al broker MQTT [Fuente: Autor]

3.3.3. Conexión al Broker MQTT

La utilización de nodos MQTT de la figura 3.32 es fundamental para la gestión de información mediante el servidor, para la suscripción a tópicos específicos y publicación

de información. Al incorporar nodos los usuarios pueden diseñar flujos visuales para gestionar la entrada y salida de datos de manera eficiente.

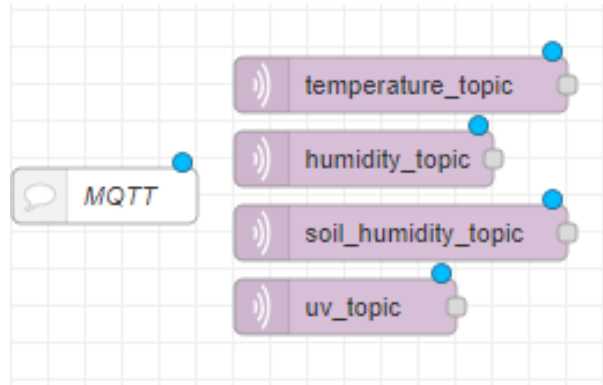


Figura 3.32: Nodos de publicación MQTT [Fuente: Node-RED]

La configuración de nodos implica ajustar parámetros específicos para vincular de manera efectiva el servidor MQTT con el puerto de enlace, la dirección del servidor y la definición de las suscripciones o publicaciones a tópicos MQTT. Además, de especificar la calidad de servicio (QoS), como se visualiza en la ilustración 3.33.

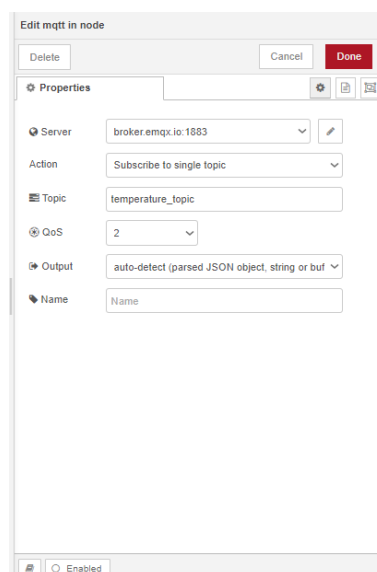


Figura 3.33: Establecimiento de nodos MQTT [Fuente: Node-RED]

Adicionalmente, en la figura 3.34 es esencial hacer ajustes de las configuraciones del servidor utilizando el localhost. En este caso particular, se emplea la dirección local broker.emqx.io y el puerto 1883.

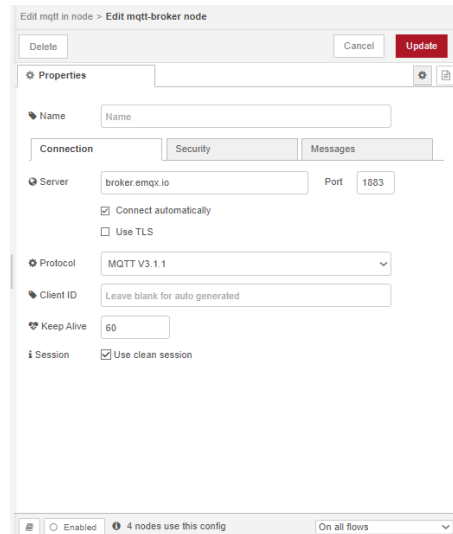


Figura 3.34: Configuración del servidor y puerto [Fuente: Node-RED]

En la configuración de cada nodo de la figura 3.32, se sigue el mismo enfoque. Se realiza la suscripción a tópicos específicos para cada magnitud, definiendo identificadores correspondientes para las variables y para el nodo final.

3.3.4. Configuración de una Dashboard en Node-RED

La configuración de un panel de visualización en Node-RED se lleva a cabo para crear una interfaz gráfica que permita monitorear y controlar datos en tiempo real.

Para configurar la interfaz, es esencial definir los nodos responsables de mostrar la información. En un principio, se organizó en cuatro conjuntos, representando las variables climáticas a monitorear como se visualiza en la ilustración 3.35.

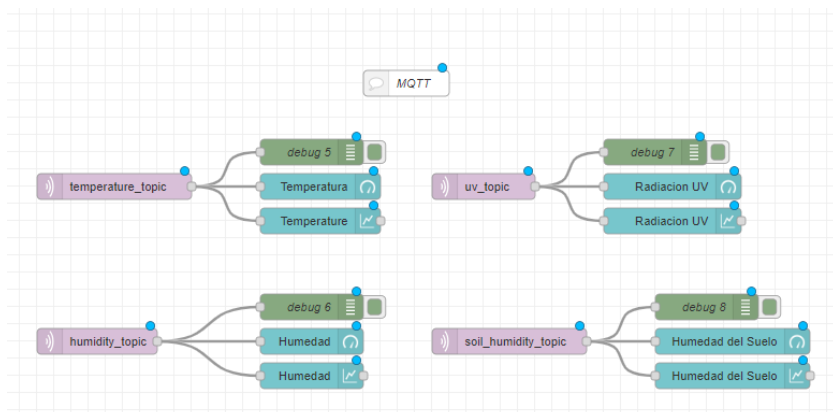


Figura 3.35: Configuración los nodos de la interfaz gráfica [Fuente: Node-RED]

En este grupo, se creó un indicador y un gráfico de línea, facilitando la separación y visualización de cada magnitud. La figura 3.36 ilustra el grupo creado, con su respectiva tabla de variables.

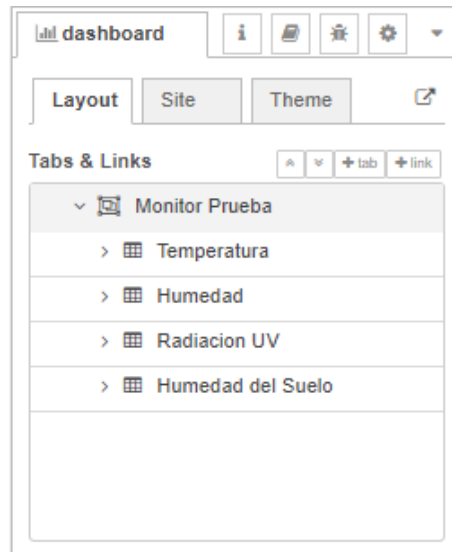


Figura 3.36: Configuración de la interfaz gráfica [Fuente: Node-RED]

La figura 3.37 el panel de visualización diseñada, que incluye indicadores para cada magnitud y presenta una gráfica temporal que permite evaluar el comportamiento de los valores a lo largo del tiempo.

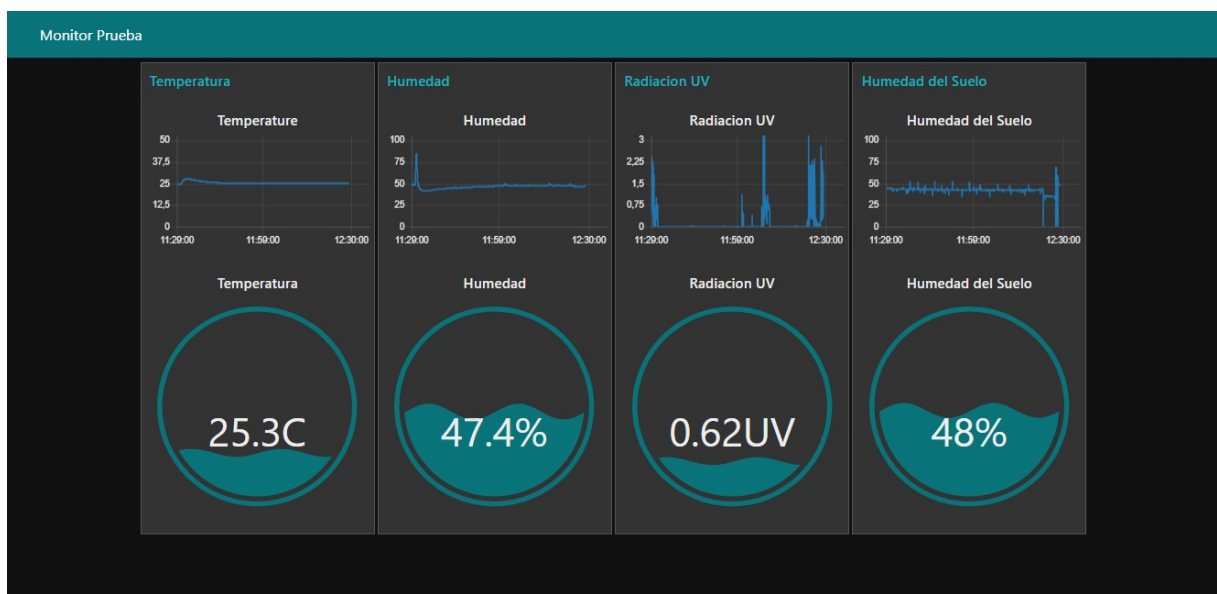
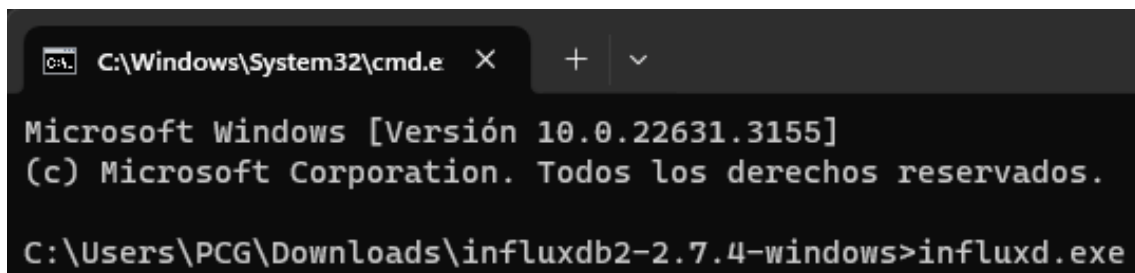


Figura 3.37: Diseño de la interfaz para el sistema de supervisión [Fuente: Node-RED]

3.3.5. Comunicación con InfluxDB

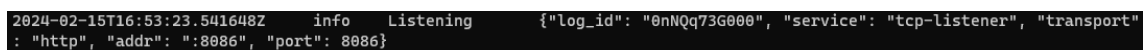
La información transmitida por el nodo final al gateway se presenta en la interfaz de la figura 3.37, pero los datos son instantáneos y no hay historial registrado. Es por eso que se estableció una base de datos de InfluxDB, para guardar las mediciones. El primer paso fue la inicialización mediante el comando (cmd), como se muestra en la figura 3.38 para preparar y optimizar la base de datos, brindando así una infraestructura sólida para el registro y evaluación eficientes de la información temporal.



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Versión 10.0.22631.3155]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\PCG\Downloads\influxdb2-2.7.4-windows>influxd.exe
```

Figura 3.38: Inicialización de InfluxDB [Fuente: Autor]

La conexión a través del puerto de enlace 8086 con InfluxDB facilita el envío eficiente de información entre sistemas como se muestra en la figura 3.39. Este puerto actúa como interfaz para la comunicación, permitiendo el acceso y almacenamiento efectivo de información en InfluxDB.



```
2024-02-15T16:53:23.541648Z info Listening {"log_id": "0nNqQ73G000", "service": "tcp-listener", "transport": "http", "addr": ":8086", "port": 8086}
```

Figura 3.39: Puerto de enlace de InfluxDB [Fuente: Autor]

La vinculación desde el menú principal de InfluxDB en el ordenador de la figura 3.40, proporciona acceso directo y sencillo a diversas funcionalidades, facilitando la gestión de bases de datos, usuarios y consultas desde una interfaz intuitiva.

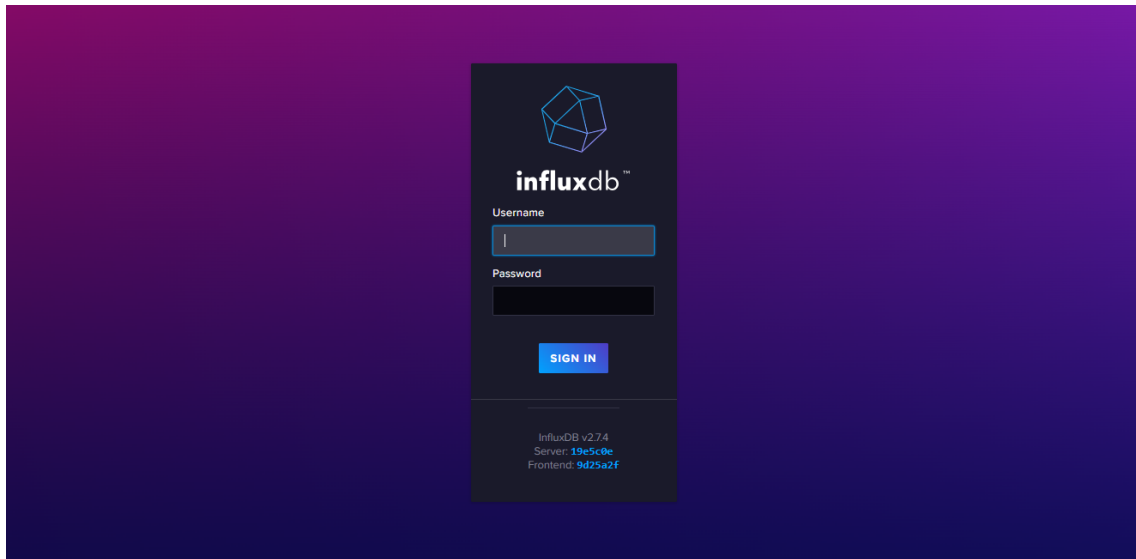


Figura 3.40: Menu de InfluxDB [Fuente: InfluxDB]

Al inicio de sesión por defecto, se creó de una organización en InfluxDB que implica establecer un entorno estructurado para gestionar y compartir datos. Esta acción se puede visualizar en la figura 3.41, con esto nos permite organizar eficientemente la información, asignar permisos y colaborar en proyectos, brindando una administración organizada y colaborativa dentro del entorno.

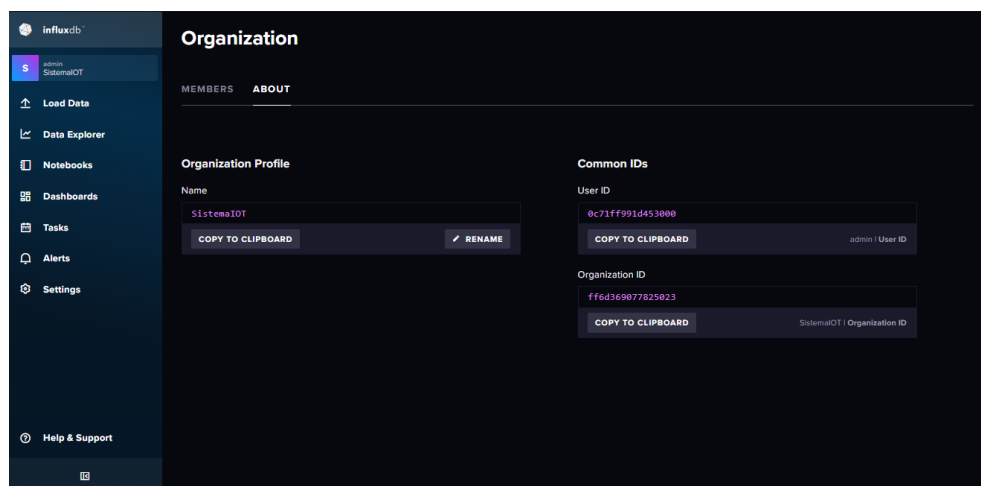


Figura 3.41: Organización en InfluxDB [Fuente: InfluxDB]

Seguido creamos de un bucket en InfluxDB como se visualiza en la ilustración 3.42, que consiste en establecer un contenedor para almacenar datos con características específicas, como la retención de datos y las políticas de almacenamiento. Esta acción

simplifica la organización y gestión eficiente de datos, permitiendo la configuración de reglas personalizadas para la retención y eliminación de información, optimizando así la administración de series temporales.

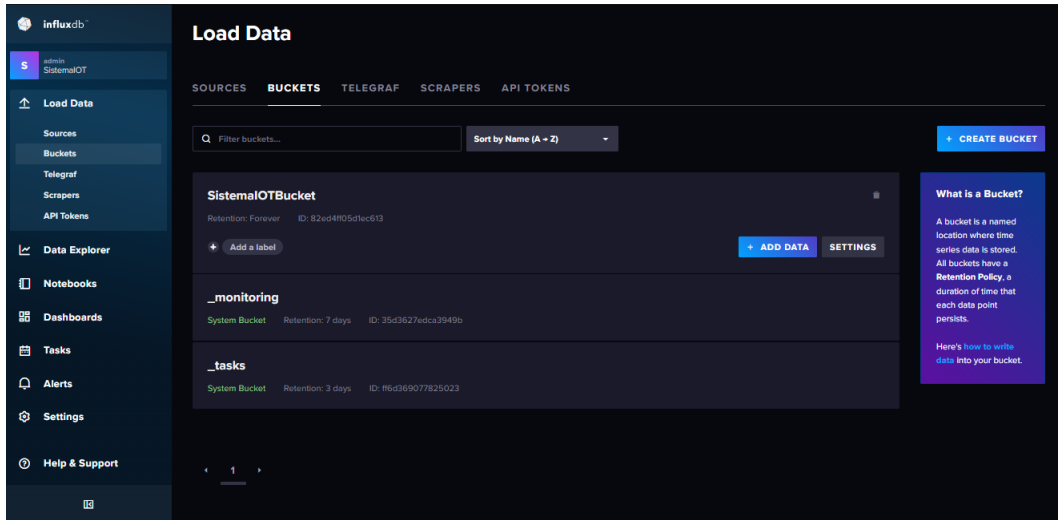


Figura 3.42: Creación del Bucket [Fuente: InfluxDB]

La conexión de InfluxDB con Node-RED facilita la integración eficiente de datos en entornos IoT, para almacenar y recuperar datos de manera estructurada. En la figura 3.43 se presenta la configuración necesaria para la operación del nodo InfluxDB.

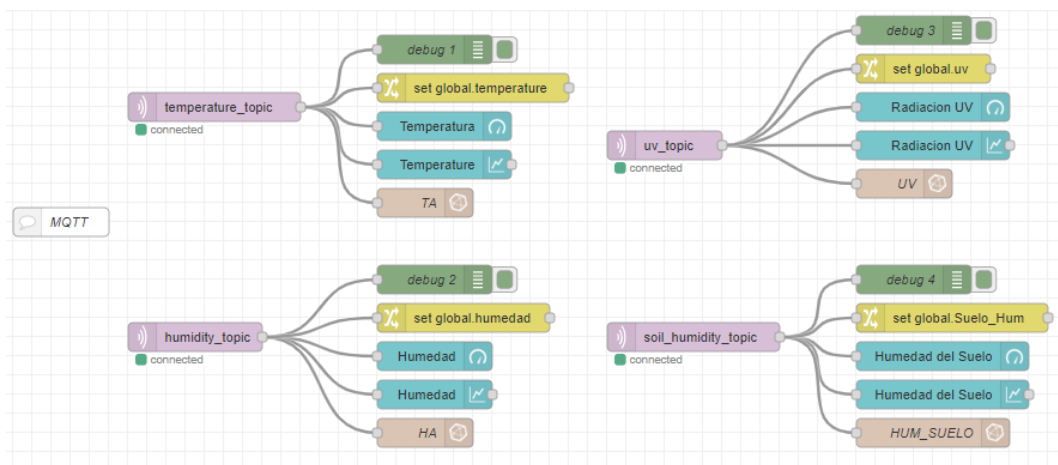


Figura 3.43: Ajustes del nodo InfluxDB [Fuente: InfluxDB]

En la configuración del nodo InfluxDB, se incluye la especificación de la organización, el bucket y la medición de la variable como se visualiza en la ilustración 3.44. Esto permite una gestión precisa de la información almacenada en series de datos

temporales. La organización determina el contexto de los datos, el bucket representa el contenedor de datos y la medición define la variable específica a almacenar. Esta configuración garantiza una estructura ordenada, eficiente para almacenar y recuperar datos relevantes.

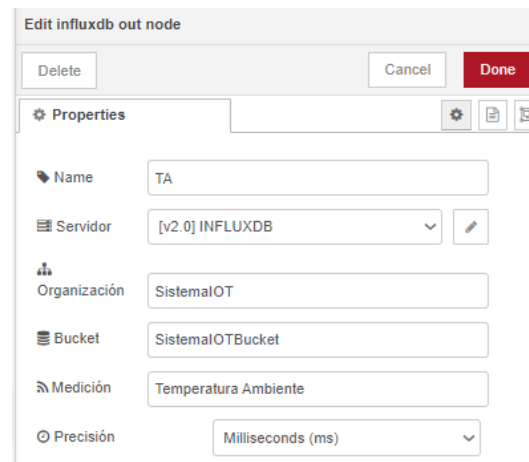


Figura 3.44: Propiedades del nodo InfluxDB [Fuente: Node-RED]

En la figura 3.45 se establece el ajuste de la dirección del servidor InfluxDB, el puerto y las credenciales en Node-RED, permitiendo una comunicación fluida entre ambos sistemas.

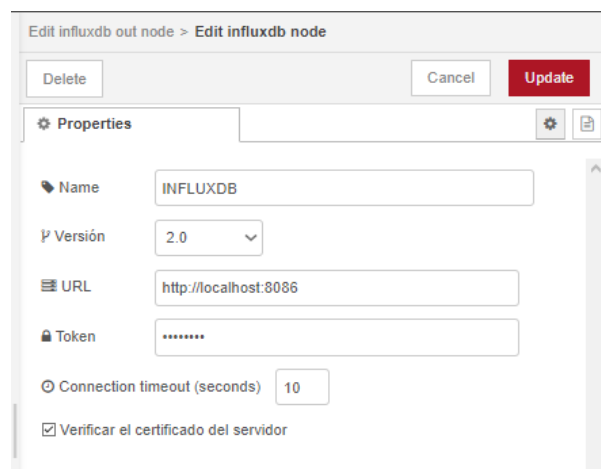


Figura 3.45: Propiedades del servidor InfluxDB [Fuente: InfluxDB]

De la misma forma se realiza la configuración de cada nodo InfluxDB de la figura 3.32, se sigue el mismo enfoque. Se realiza la configuración de los detalles de conexión, como la dirección del servidor, el puerto y las credenciales de acceso.

El proceso de registro de datos en InfluxDB para la temperatura, humedad, radiación UV y humedad del suelo. Cada conjunto de datos climáticos se registra y almacena según esta estructura, asegurando una gestión eficiente y accesible para el análisis y monitorización a largo plazo.

La figura 3.46 representa el registro de la variable ambiental de temperatura ambiente creando un historial de los datos almacenados.

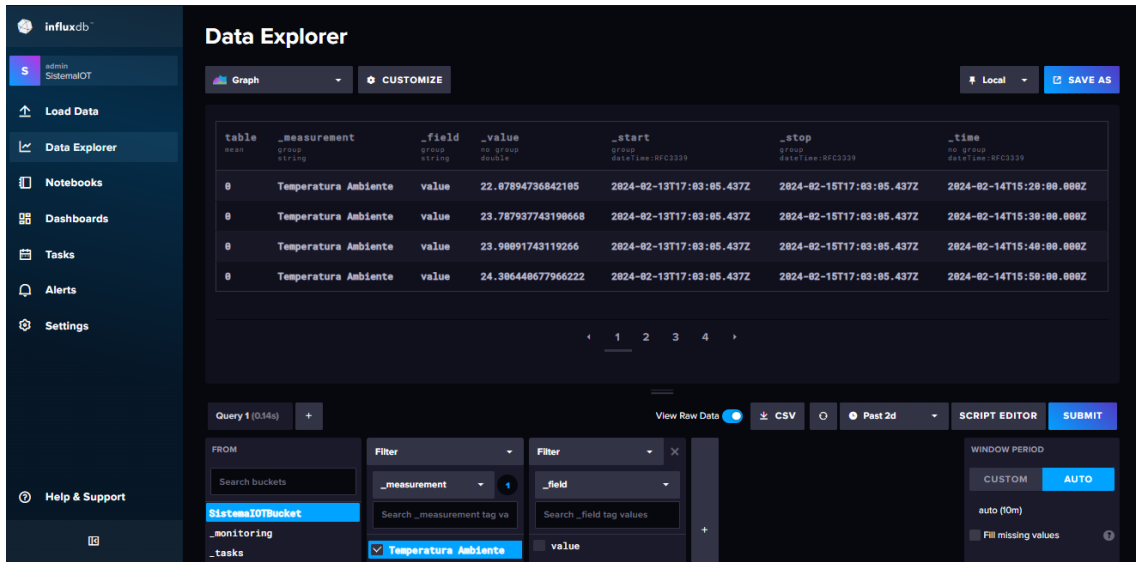


Figura 3.46: Registro de datos de la temperatura ambiente [Fuente: InfluxDB]

La figura 3.47 representa el registro de la variable ambiental de humedad ambiental creando un historial de los datos almacenados.

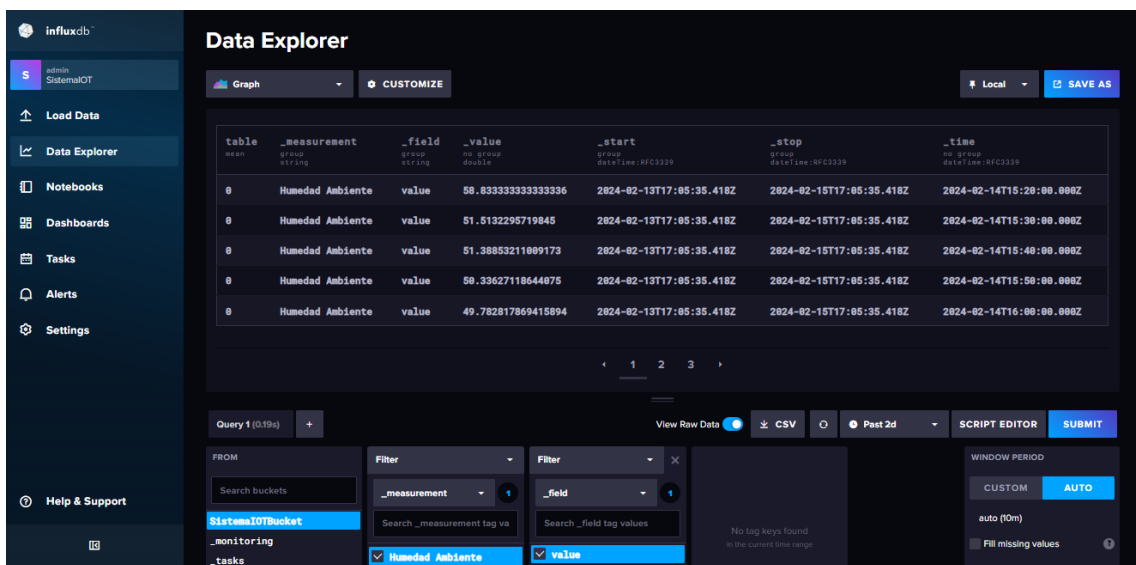


Figura 3.47: Registro de datos de la humedad ambiente [Fuente: InfluxDB]

La figura 3.48 representa el registro de la variable ambiental de radiación solar (UV) creando un historial de los datos almacenados.

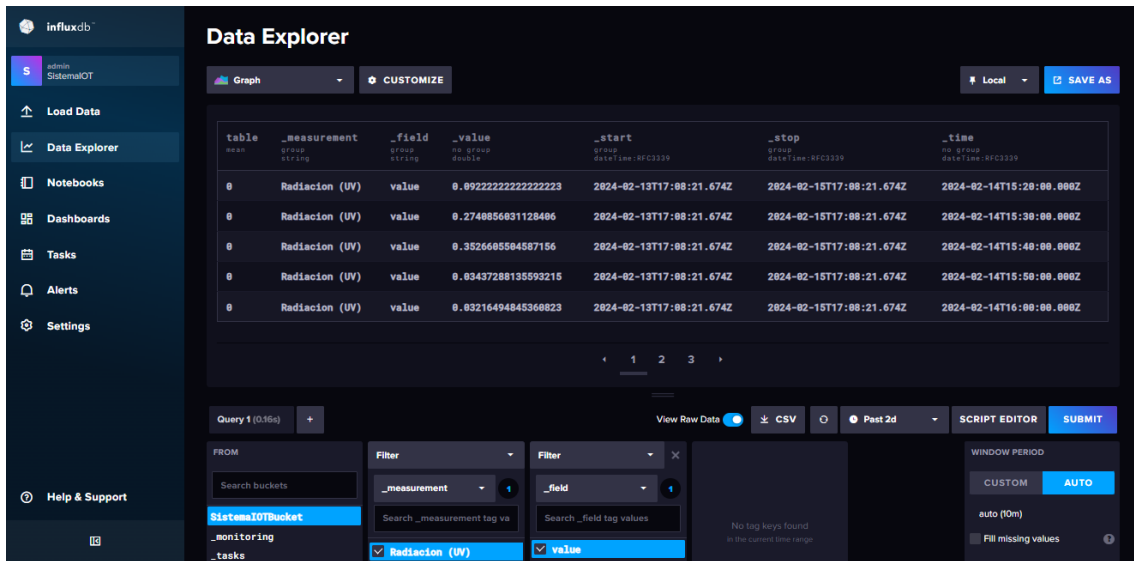


Figura 3.48: Registro de datos de la radiación solar (UV) [Fuente: InfluxDB]

Y por último en la figura 3.49 representa el registro de la variable ambiental de humedad del suelo creando un historial de los datos almacenados.

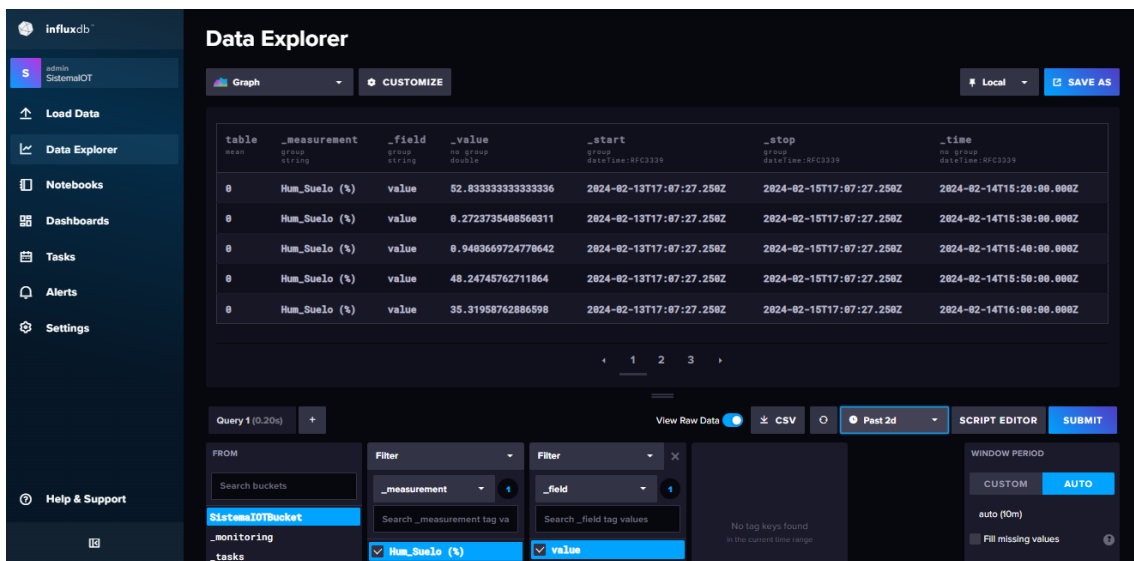


Figura 3.49: Registros de humedad del suelo [Fuente: InfluxDB]

3.3.6. Implementación de una interfaz gráfica en Grafana

Grafana nos crea visualizaciones comprensibles y efectivas a partir de los datos almacenados. Configurando paneles y tableros, se organiza la información para facilitar la interpretación. Esta implementación permite a los usuarios explorar, comprender fácilmente patrones y tendencias en conjuntos de información, optimizando la elección de acciones mediante los entornos de monitoreo y análisis. La figura 3.50, ilustra la configuración inicial predeterminada de Grafana

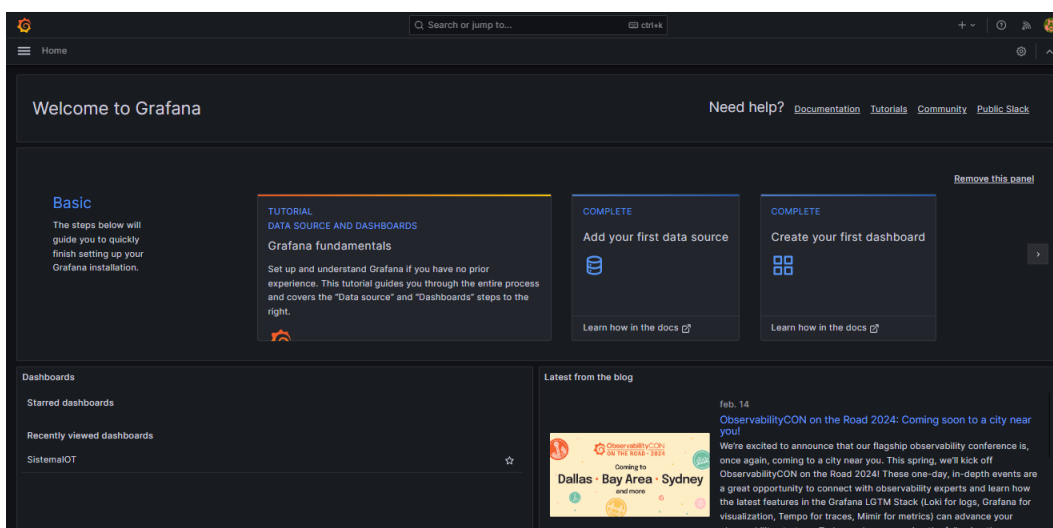


Figura 3.50: Menu principal de Grafana [Fuente: Grafana]

La conexión entre InfluxDB y Grafana se realiza configurando una fuente de datos en Grafana hacia InfluxDB como se puede ver en la figura 3.51. Esta vinculación permite la visualización de datos almacenados en InfluxDB mediante paneles personalizables, facilitando el análisis y monitoreo eficiente de información climática u otros conjuntos de datos.

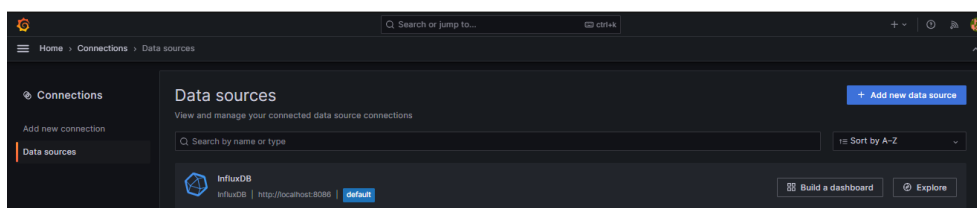


Figura 3.51: Conexión de InfluxDB [Fuente: Grafana]

La conexión entre el almacenamiento de información y Grafana se establece mediante HTTP en InfluxDB como se muestra en la figura 3.52. Al configurar se posibilita la recuperación y visualización de datos almacenados.

La comunicación HTTP facilita la integración fluida entre ambas plataformas, permitiendo una interpretación y monitoreo eficientes de datos climáticos u otros conjuntos de información.

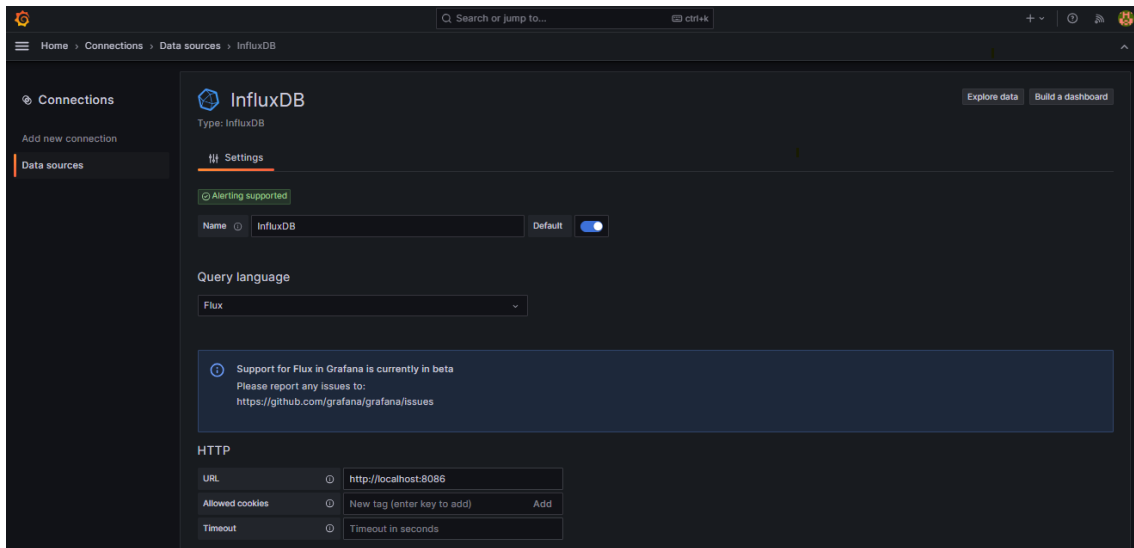


Figura 3.52: Integración HTTP entre InfluxDB y Grafana [Fuente: Grafana]

En la figura 3.53 se configura de la organización y el bucket en InfluxDB. Esta configuración permite una recuperación precisa de datos almacenados en InfluxDB, facilitando la creación de dashboards y paneles personalizables.

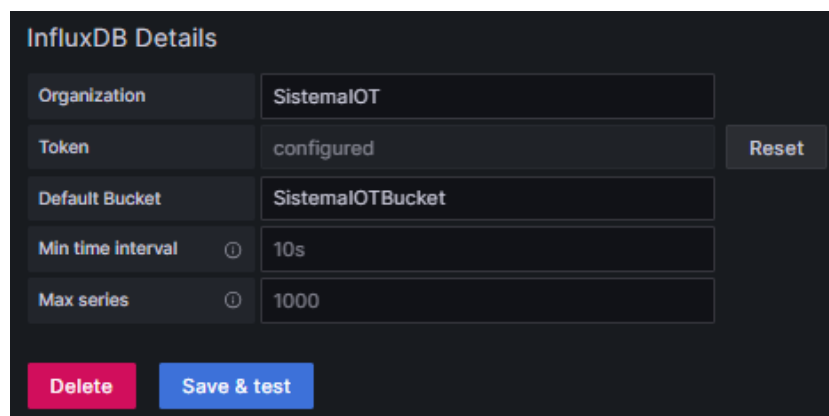


Figura 3.53: Configuración de la organización y bucket de InfluxDB en Grafana [Fuente: Grafana]

En la figura 3.54 se implementa el código de consulta de InfluxDB que se integra con Grafana para recuperar datos específicos. Filtra los datos del bucket "SistemaIOTBucket" por la medida "Temperatura Ambiente", seleccionando el valor del dato y representación como indicador.

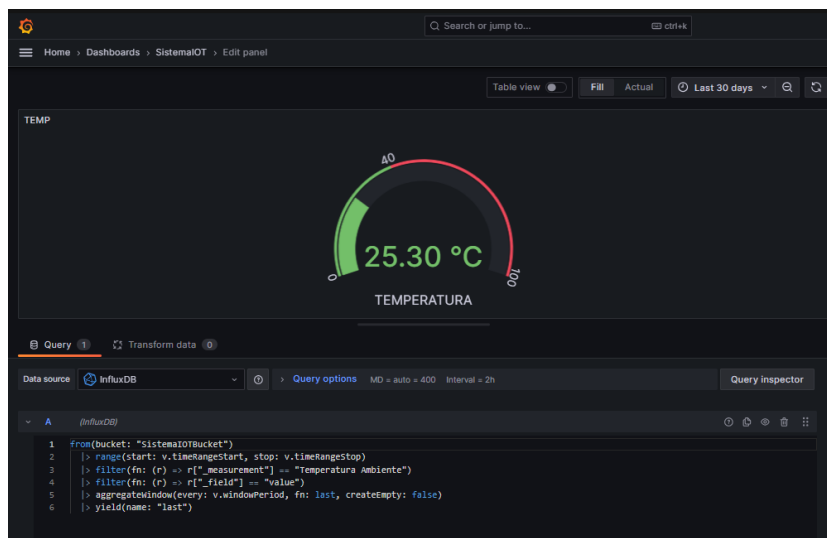


Figura 3.54: Desarrollo del gráfico indicador en Grafana [Fuente: Grafana]

Se sigue el mismo proceso para la variable de Temperatura Ambiente, pero en este caso se representará mediante un gráfico lineal tal como se representa en la figura 3.55.

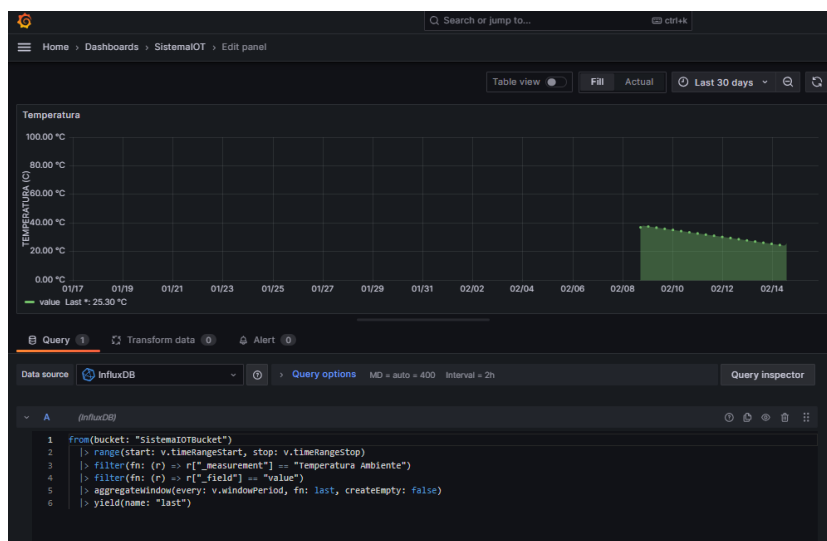


Figura 3.55: Gráfico lineal en Grafana [Fuente: Grafana]

Este proceso permite visualizar la última temperatura registrada en el intervalo

de tiempo especificado en Grafana, proporcionando una herramienta eficiente para el análisis y monitoreo de datos climáticos.

El mismo procedimiento se realiza para cada variable y se organiza en el menú principal como se puede ver en la figura 3.56.

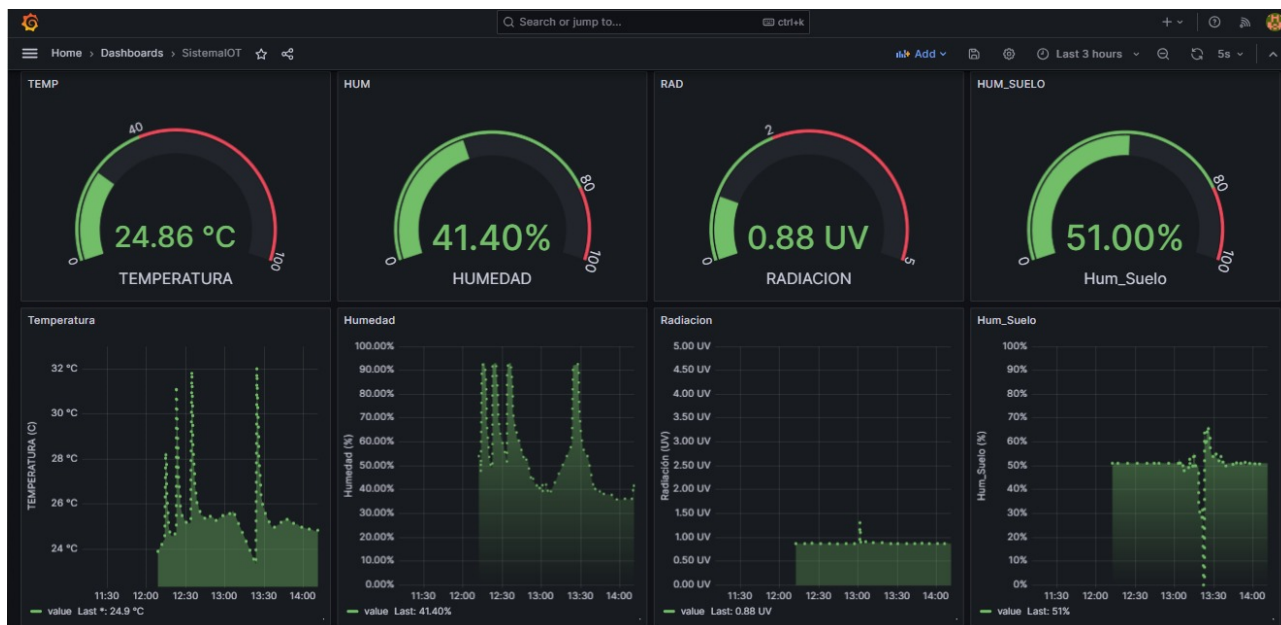


Figura 3.56: Plataforma de visualización en Grafana [Fuente: Grafana]

3.4. Implementación y validación del sistema

3.4.1. Elaboración y montaje del circuito impreso

Para la elaboración de la placa PCB se utilizó el método del planchado, es un proceso eficiente y accesible para crear circuitos impresos. Este enfoque combina la creación de un diseño en software, la impresión del diseño en papel fotográfico y la transferencia térmica a una placa. La posterior aplicación de productos químicos y la perforación completan el proceso como se puede ver en la figura 3.57. Esta técnica casera ofrece una alternativa viable para fabricar prototipos de PCBs de manera económica y personalizada.

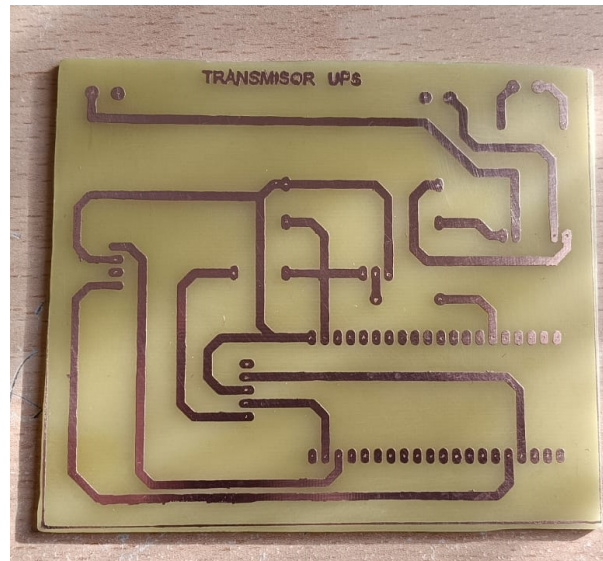


Figura 3.57: Elaboración de la placa PCB [Fuente: Autor]

El montaje de los componentes electrónicos en la placa PCB es una fase crucial del desarrollo, requiere precisión y atención para soldar cada componente en su ubicación designada como se muestra en la figura 3.62, para garantizar el funcionamiento confiable del dispositivo. Esta etapa involucra la conexión de sensores, módulos, y otros elementos, siguiendo el esquemático diseñado.

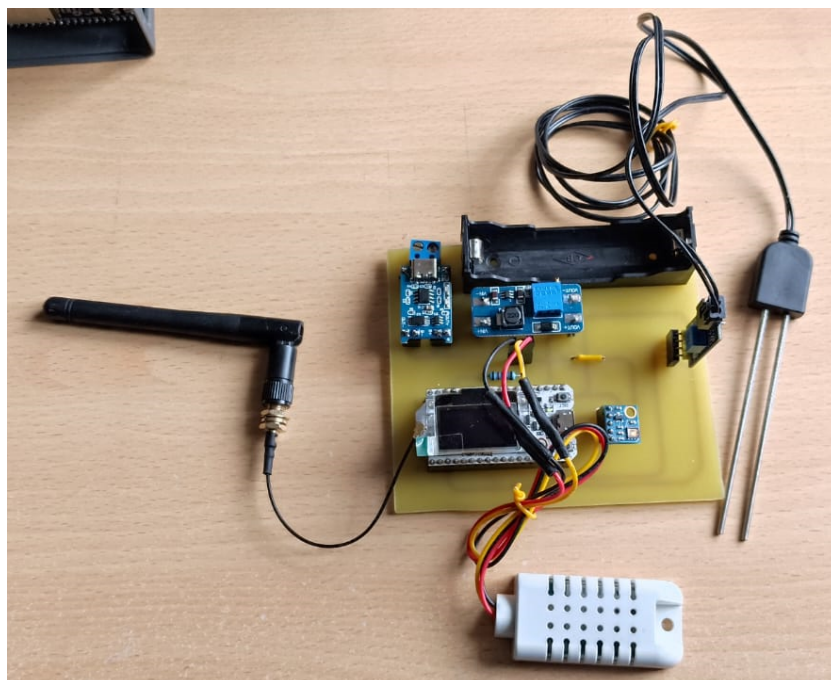


Figura 3.58: Montaje de la placa PCB [Fuente: Autor]

3.4.2. Diseño 3D

Dimensiones (mm) de la estructura de la caja del dispositivo transmisor:

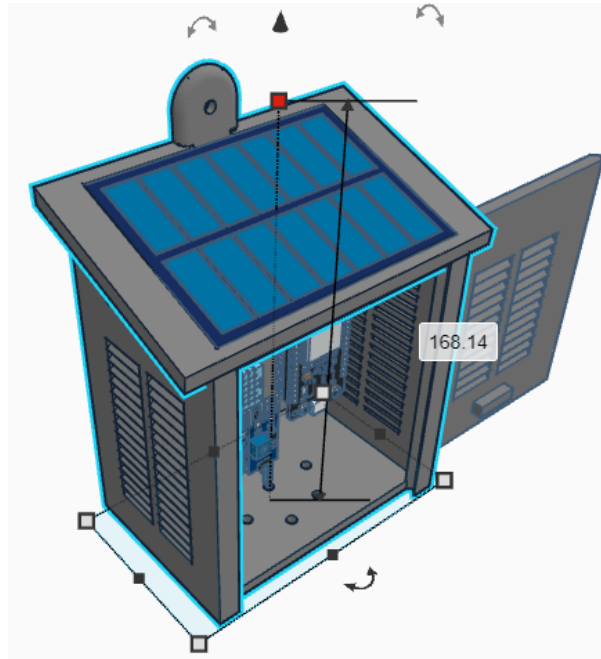


Figura 3.59: Dimensiones (mm) de la base [Fuente: Fusion360]

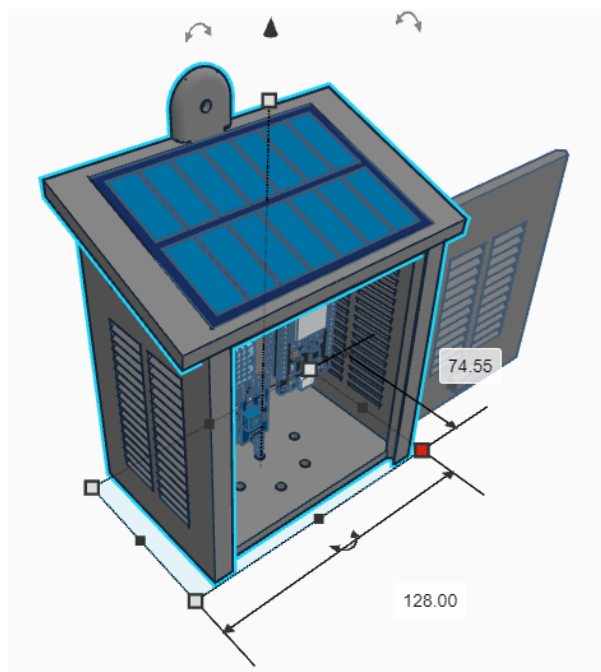


Figura 3.60: Dimensión (mm) de la altura [Fuente: Fusion360]

Dimensiones (mm) de la estructura de la caja del dispositivo receptor:

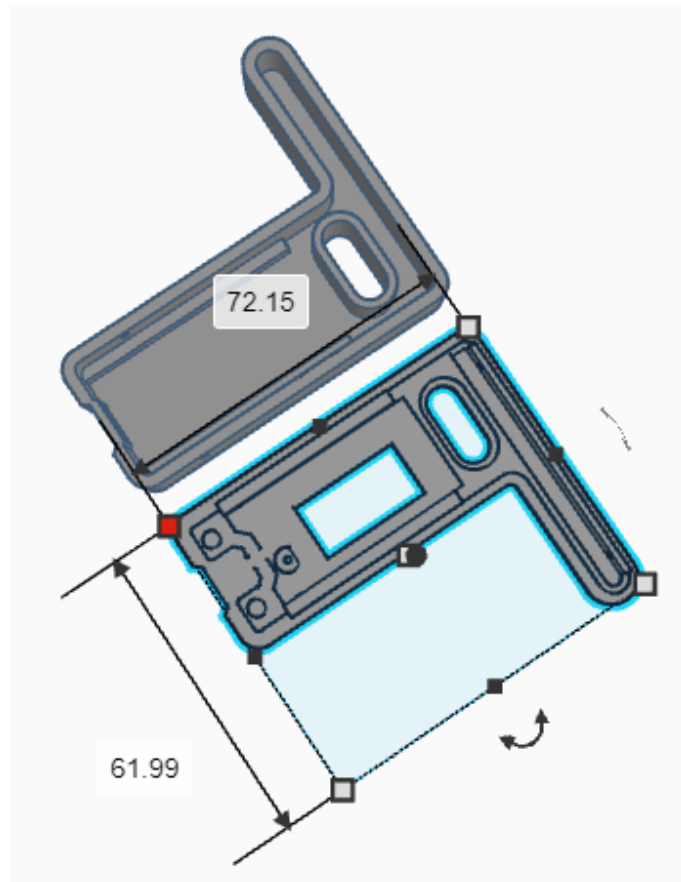


Figura 3.61: Dimensiones (mm) de la base [Fuente: Fusion360]

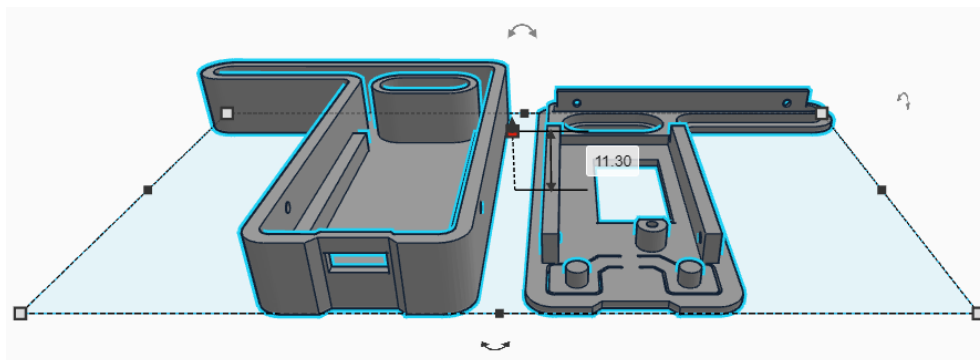


Figura 3.62: Dimensión (mm) de la altura [Fuente: Fusion360]

3.4.3. Impresión 3D

Para la impresión de la estructura para el receptor y transmisor involucra la utilización de tecnologías de impresión 3D. Este proceso implica la creación de modelos

tridimensionales en software como se muestra en la figura 3.63, seguido por la impresión capa por capa de los componentes utilizando materiales como plástico.

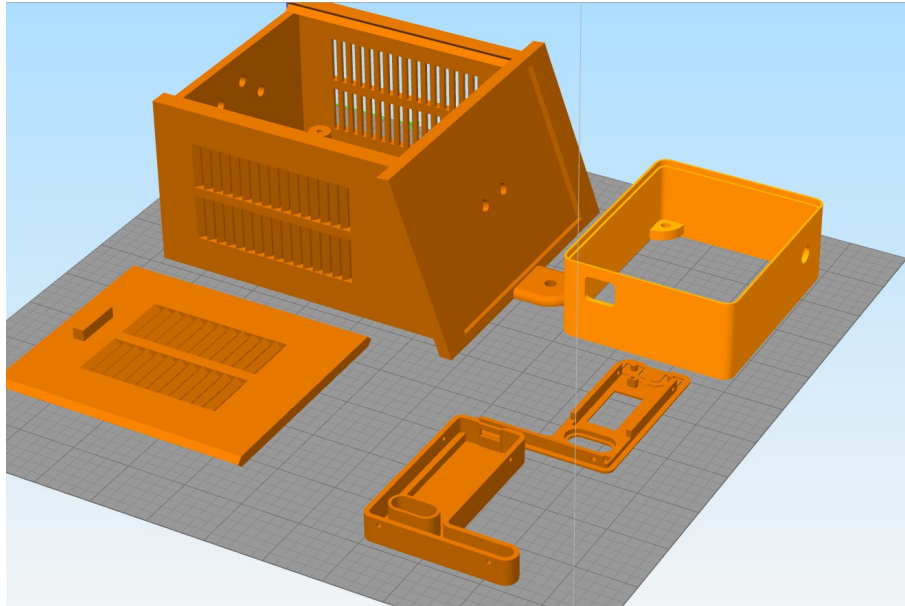


Figura 3.63: Modelos 3D de las estructuras [Fuente: Autor]

La impresión 3D proporciona flexibilidad en el diseño y la personalización de la carcasa como se puede ver en la figura 3.64, permitiendo adaptarla a las necesidades específicas del proyecto.



Figura 3.64: Impresión en 3D de las estructuras [Fuente: Autor]

3.4.4. Montaje de la estructura del dispositivo transmisor y receptor

En el dispositivo transmisor como se puede ver en la figura 3.65, la placa PCB incorpora de manera compacta los sensores de temperatura, radiación UV y humedad del suelo, junto con el microcontrolador para el procesamiento y envío de datos.

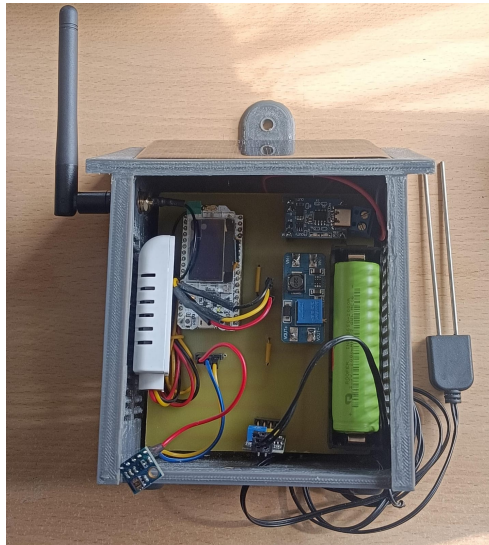


Figura 3.65: Visualización interna del dispositivo transmisor [Fuente: Autor]

Además, la estructura 3D impresa proporciona soporte físico y protección esto se puede ver en la figura 3.66.



Figura 3.66: Montaje del dispositivo transmisor [Fuente: Autor]

En el receptor como se puede ver en la figura 3.67, únicamente se dispone de un microcontrolador que se encarga de recibir y procesar los datos provenientes del transmisor.



Figura 3.67: Visualización interna del dispositivo receptor [Fuente: Autor]

La placa PCB y estructura 3D impresa en el receptor están simplificadas ya que no requieren sensores adicionales como se puede ver en la figura 3.69, centrándose en la funcionalidad de recepción y conexión con la plataforma IoT.



Figura 3.68: Montaje del dispositivo receptor [Fuente: Autor]

3.4.5. Aplicación en el huerto agrícola

La colocación estratégica del dispositivo transmisor que se visualiza en la figura 3.69, juega un papel fundamental en la efectividad del monitoreo. Ubicado en una posición central, garantiza la captura precisa de datos climáticos esenciales.



Figura 3.69: Montaje del dispositivo transmisor en el huerto agrícola [Fuente: Autor]

La placa PCB se integra en la estructura 3D impresa como se muestra en la figura 3.66, asegurando su resistencia a condiciones ambientales adversas. La conexión inalámbrica permite una comunicación fluida con el receptor. Esta disposición posibilita un monitoreo detallado de la temperatura, humedad del suelo y radiación UV en tiempo real. El dispositivo contribuye a la optimización de las condiciones ambientales para maximizar el rendimiento agrícola.



Figura 3.70: Colocación del sensor en la tierra [Fuente: Autor]

3.4.6. Recepción y almacenamiento de información de cada variable climática

La visualización de datos en Grafana durante los últimos 30 días como se puede ver en la figura 3.71 permite un análisis histórico de los parámetros ambientales registrados por el dispositivo receptor. Con gráficos temporales y paneles personalizados, los usuarios pueden identificar patrones, tendencias y variaciones en la temperatura, humedad y radiación UV.

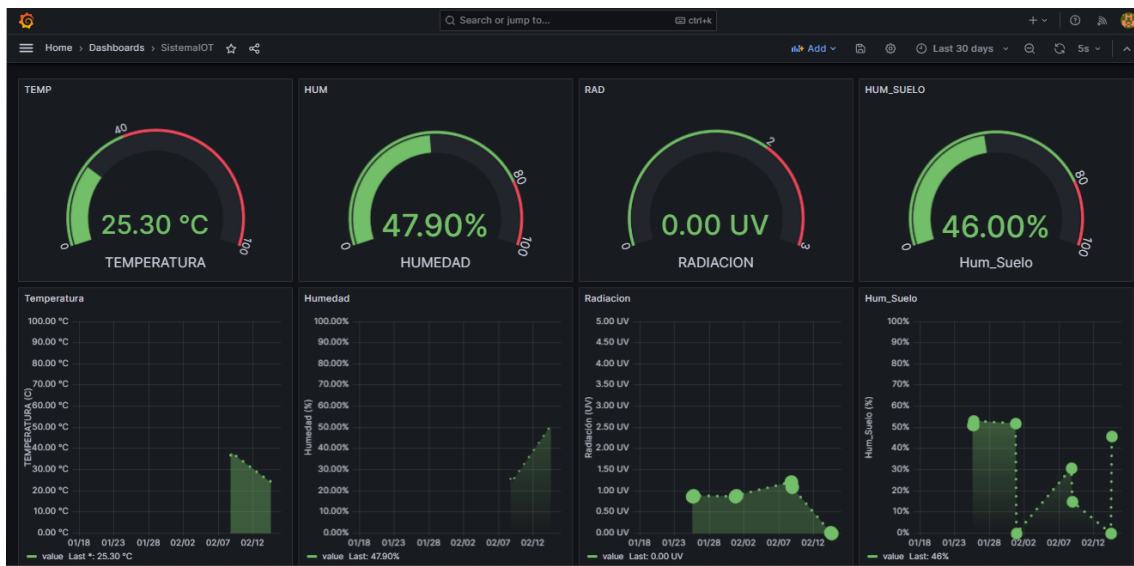


Figura 3.71: Historial ambiental de los últimos 30 días [Fuente: Autor]

Esta funcionalidad proporciona una perspectiva a largo plazo sobre el entorno del huerto agrícola, permitiendo ajustes precisos en las prácticas agrícolas según las condiciones climáticas previas. La representación a lo largo del tiempo en Grafana ofrece una herramienta valiosa para evaluar el impacto estacional en el cultivo y optimizar las estrategias agrícolas en función de los datos recopilados.

Capítulo 4

Conclusiones y Recomendaciones

4.1. Conclusiones

- El desarrollo del dispositivo de medición, se logró crear un sistema integral capaz de monitorear la temperatura, radiación UV y humedad del suelo. Usando la combinación de tecnologías avanzadas, proporciona una herramienta efectiva para la gestión ambiental y agrícola. Este dispositivo ofrece información precisa y en tiempo real, permitiendo una toma de decisiones informada. Su diseño compacto y fácil implementación lo convierte en una solución versátil para diversos contextos, destacando su contribución a la monitorización ambiental y el mejor aprovechamiento de recursos en diferentes aplicaciones.
- De los sistemas de comunicación inalámbrica se utilizó LoRa a 915MHz entre el dispositivo transmisor y receptor, la elección de la banda de frecuencia proporciona una cobertura amplia de transmisión y de bajo consumo de energía en los dispositivos, vital en entornos agrícolas. La capacidad de transmitir datos a largas distancias sin comprometer la calidad de la señal mejora la conectividad en áreas extensas. Este logro fortalece la funcionalidad del dispositivo de medición, asegurando una transmisión eficaz de datos ambientales y respaldando la toma de decisiones informada en diversos escenarios, destacando su relevancia en aplicaciones agrícolas y medioambientales.
- La creación de una plataforma de visualización en Grafana, conectada a la base

de datos InfluxDB gestionada por Node-RED, optimiza la presentación de las variables ambientales. La integración fluida entre el receptor y Grafana permite una representación gráfica dinámica y detallada de la información recopilada. La combinación de tecnologías ofrece una interfaz intuitiva para analizar y comprender los datos en tiempo real, facilitando la toma de decisiones informadas. Esta solución robusta y eficiente proporciona una visión completa de las mediciones, mejorando la capacidad de monitoreo y gestión en entornos agrícolas y medioambientales.

- La prueba exitosa del dispositivo transmisor en un huerto agrícola de hortalizas valida su eficacia en tiempo real. Al monitorear la temperatura, radiación UV y humedad del suelo, los beneficios son evidentes ya que facilita la toma de decisiones basada en datos, permitiendo ajustes precisos en el riego y la protección contra condiciones climáticas adversas. Optimiza la productividad al garantizar condiciones óptimas para el crecimiento de las hortalizas. Además, al prevenir pérdidas y maximizar recursos, este sistema de monitoreo contribuye a la sostenibilidad agrícola y a la eficiencia en la gestión de cultivos

4.2. Recomendaciones

- Mejorar la eficiencia energética del dispositivo permitiría una vida útil más prolongada de la batería, optimizando la operación continua en entornos agrícolas. La implementación de tecnologías de bajo consumo y paneles solares con mayor capacidad de alimentación podría ser clave para una autonomía mejorada.
- Reforzar la seguridad en la comunicación LoRa podría ser vital para proteger los datos sensibles. Implementando protocolos de encriptación sólidos y autenticación mejoraría la integridad de la transmisión, asegurando la confidencialidad de la información.
- Ampliar las opciones de personalización en Grafana permitiría adaptar la interfaz a las necesidades específicas del usuario. Proporcionar herramientas intuitivas para análisis y generación de informes con un enfoque más completo.

- Incorporar sensores adicionales, como medidores de nutrientes del suelo, podría ofrecer una visión más completa de las condiciones agrícolas. Integrar alertas automáticas ante condiciones críticas agilizaría la respuesta del agricultor, mejorando la capacidad de gestión del cultivo.

Glosario

API Interfaz de Programación de Aplicaciones – Application Programming Interface.

CSS Espectro Ensanchado por Chirp – Chirp Spread Spectrum.

FAO Organización de las Naciones Unidas para la Agricultura y la Alimentación.

HTTP Protocolo de Transferencia de Hipertexto – Hypertext Transfer Protocol.

IoT Internet de las cosas – Internet Of Things.

LoRa De largo Alcance – Long Range.

LoRaWan LoRa Wide Area Network.

MQTT Transporte de telemetría de colas de mensajes – Message Queuing Telemetry Transport.

P2P De igual a igual – Peer to Peer.

RF Frecuencia de radio – Radio Frequency.

RFA Radiación Fotosintéticamente Activa – Photosynthetically Active Radiation.

TCP/IP Protocolo de Control de Transmisión/Protocolo de Internet – Transmission Control Protocol/Internet Protocol.

UV Radiación Ultravioleta – Ultraviolet Radiation.

Wi-Fi Wireless Fidelity.

Referencias

- [1] A. M. De Vats y A. C. Biel, *WORLD GOVERNMENT SUMMIT*, 2018.
- [2] *FOOD WASTE INDEX REPORT*. 2021, ISBN: 9789280738681.
- [3] K. A. Dubin, «IoT en Agricultura,» 2021. dirección: www.movildat.comwww.movildat.com.
- [4] M. J. A. Baig, M. T. Iqbal, M. Jamil y J. Khan, «Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol,» *Energy Reports*, vol. 7, págs. 5733-5746, nov. de 2021, ISSN: 23524847. DOI: 10.1016/j.egyrs.2021.08.190.
- [5] J. G. M. A, T. E. Mathew, K. S. N et al., «Microclimate monitoring system for irrigation water optimization using IoT,» *Measurement: Sensors*, vol. 27, jun. de 2023, ISSN: 26659174. DOI: 10.1016/j.measen.2023.100727.
- [6] N. E. Guevara, Y. H. Bolaños, J. P. Diago y J. M. Segura, «Development of a low-cost IoT system based on LoRaWAN for monitoring variables related to electrical energy consumption in low voltage networks,» *HardwareX*, vol. 12, oct. de 2022, ISSN: 24680672. DOI: 10.1016/j.ohx.2022.e00330.
- [7] *Agricultura 4.0. Nuevas tecnologías en la agricultura | ATRIA Innovation*. dirección: <https://www.atriainnovation.com/agricultura-4-0-nuevas-tecnologias-en-la-agricultura/>.
- [8] M. Wandel, A. Luis y N. Yáñez, «NUEVAS TECNOLOGÍAS Y AGRICULTURA 4.0: IMPACTO EN LOS RECURSOS HUMANOS DE LA INDUSTRIA AGRÍCOLA EN CENTROAMÉRICA,» 2021.
- [9] Y. de la Cruz Elizondo Juan Camilo Fontalvo Buelvas, «Manual de iniciación al huerto casero: Una guía para producir alimentos saludables,» 2020.

- [10] A. Héctor, F. Mafla, C. Delsa et al., «© MANEJO E INTERPRETACIÓN DE VARIABLES DEL CLIMA Y DE AGROMETEOROLOGÍA desde la finca del agricultor, ante los efectos de la ola invernal,» dirección: www.produmedios.org.
- [11] *Importancia de la Radiación Solar en la Producción Bajo Invernadero* | Intagri S.C. dirección: <https://www.intagri.com/articulos/horticultura-prottegida/importancia-de-la-radiacion-solar-en-la-produccion-bajo-invernadero>.
- [12] J. M. Torres, «RELACIÓN DE LA RADIACIÓN SOLAR CON LA PRODUCCIÓN DE PLANTAS: AGROPRODUCTIVAS Relationship between solar radiation and plant production: agro-productive plants,» *Recursos Naturales, Bolivia*, vol. 9, pág. 2022, ISSN: 2518-6868. DOI: 10.53287/oqym7033yy88k.
- [13] *Temperatura Del Suelo Y Crecimiento De Los Cultivos*. dirección: <https://eos.com/es/blog/temperatura-del-suelo/>.
- [14] *Temperatura del Suelo y su importancia*. dirección: <https://www.tipsytemasagronicos.com/temperatura-del-suelo-y-su-importancia/>.
- [15] J. Salazar e Y. S. Silvestre, «INTERNET DE LAS COSAS,» dirección: <http://www.techpedia.eu>.
- [16] A. Wang, *LoRa contra Zigbee: La mejor tecnología para la conectividad IoT*, 2022. dirección: <https://www.mokolora.com/es/lora-vs-zigbee-which-is-better/>.
- [17] A. wang, *Comparación entre LoRa y otras tecnologías inalámbricas*, 2021. dirección: <https://www.mokolora.com/es/lora-and-wireless-technologies/>.
- [18] *IoT en Agricultura: Conectando Campos Inteligentes*. dirección: <https://www.sembrarsaber.com.ar/iot-en-agricultura/>.
- [19] *Tareas del huerto – Botanical-online*. dirección: <https://www.botanical-online.com/cultivo/tareas-huerto>.
- [20] *Frequency Plans by Country | The Things Network*. dirección: <https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country/>.
- [21] X. Muñoz, *Tecnología LoraWan – DSET ENERGY*, 2019. dirección: <http://www.dset-energy.com/2019/07/12/descripcion-lorawan/>.
- [22] *Tecnología LoRA y LoRAWAN - Catsensors*. dirección: <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>.

- [23] *Haciendo IoT con LoRa: Capitulo 2.- Tipos y Clases de Nodos* | by Sabas | Beelan | Medium. dirección: <https://medium.com/beelan/haciendo-iot-con-lora-capitulo-2-tipos-y-clases-de-nodos-3856aba0e5be>.
- [24] A. Bickley, *Tecnología LoRa para sistemas IoT*, 2016. dirección: <https://www.comunicacionesinalambricashoy.com/tecnologia-lora-sistemas-iot/>.
- [25] *2.1 LoRa Multipunto – Esquema – GIRNI*. dirección: <http://blog.espol.edu.ec/girni/lora-multipunto-esquema/>.
- [26] H. Alonso, *LoRaWAN - Parte 2 - Topología y dispositivos - El bosque de silicio*. dirección: <https://elbosquedesilicio.es/lorawan2/>.
- [27] *ESP32 LoRa Changing Frequency : 16 Steps (with Pictures) - Instructables*. dirección: <https://www.instructables.com/ESP32-Lora-Changing-Frequency/>.
- [28] *Cima 9 Aplicaciones de agricultura inteligente LoRa a tener en cuenta*. dirección: <https://www.mokolora.com/es/lora-smart-agriculture-applications/>.
- [29] *MQTT Working, Types, Applications*. dirección: <https://www.spiceworks.com/tech/iot/articles/what-is-mqtt/>.
- [30] *MQTT | Aprendiendo Arduino*. dirección: <https://aprendiendoarduino.wordpress.com/2018/11/19/mqtt/>.
- [31] M. Miguel, *NODE-RED COMO HERRAMIENTA VISUAL DE DISPOSITIVOS IOT*, 2021.
- [32] S. Büttrich, «Node-RED, InfluxDB and Grafana for IoT Motivation,» 2018.
- [33] R. J. Hernandez, *Building IoT visualizations using Grafana : power up your IoT projects with Grafana*. 2022, ISBN: 9781803236124.
- [34] *ESP32 with LoRa using Arduino IDE | Random Nerd Tutorials*. dirección: <https://randomnerdtutorials.com/esp32-lora-rfm95-transceiver-arduino-ide/>.
- [35] «WiFi Lora32 Manual,» 2020. dirección: <https://heltec.org>.
- [36] G. Hallberg, *Wireless Communications with Arduino and the ESP32*, 2021.
- [37] P. Docs y D. •. Simulator, «Connecting to a DHTxx Sensor Using a DHTxx Sensor DHT CircuitPython Code • Adafruit CircuitPython Module Install • Wiring • Usage • Example Code,» dirección: <https://adafru.it/uF2>.
- [38] *AM2302 datasheet*. dirección: <https://datasheetspdf.com/pdf-file/942482/ETC/AM2302/1>.

- [39] *Guide for Soil Moisture Sensor YL-69 or HL-69 with the Arduino | Random Nerd Tutorials.* dirección: <https://randomnerdtutorials.com/guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/>.
- [40] *MÓDULO SENSOR DE HUMEDAD DE SUELO HD-38 HIGRÓMETRO ANTICORROSIVO – Grupo Electrostore, 2019.* dirección: <https://grupoelectrostore.com/shop/sensores/temperatura/modulo-sensor-de-humedad-de-suelo-hd-38-higrometro-anticorrosivo/>.
- [41] *UV Sensor ML8511 Arduino for UV Ray Intensity Meter.* dirección: <https://how2electronics.com/uv-sensor-ml8511-arduino-uv-ray-intensity-meter/>.
- [42] *ML8511 datasheet.* dirección: <https://datasheetspdf.com/pdf-file/1188772/LAPIS/ML8511/1>.
- [43] *MT3608 DC To DC Step Up Boost Converter Module Pinout, Datasheet, Specs - Components Info.* dirección: <https://www.componentsinfo.com/mt3608-module-pinout-datasheet/>.
- [44] *MT3608 pdf, MT3608 Description, MT3608 Datasheet, MT3608 view ::: ALLDATASHEET :::* dirección: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1131968/ETC1/MT3608.html>.
- [45] *TP4056 3.7V Li-ion 18650 Battery Charger Module Pinout, Datasheet Details - Components Info.* dirección: <https://www.componentsinfo.com/tp4056-module-pinout-datasheet-details/>.
- [46] *TP4056 pdf, TP4056 Description, TP4056 Datasheet, TP4056 view ::: ALLDATASHEET :::* dirección: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1487471/ETC2/TP4056.html>.
- [47] *Batería Recargable 18650 Litio 3,7 V 2000mah, Picua Outdoor | MercadoLibre.* dirección: https://articulo.mercadolibre.com.ve/MLV-706554494-bateria-recargable-18650-litio-37-v-2000mah-picua-outdoor-_JM.
- [48] *Solar Panel 110x60mm DC 5V 0.75W 80mA Solar Cells Epoxy.* dirección: <https://udvabony.com/product/solar-panel-110-x-60mm/>.
- [49] *Calculador de vida útil de una batería | DigiKey Electronics.* dirección: <https://www.digikey.com/es/resources/conversion-calculators/conversion-calculator-battery-life>.

Apéndices

Apéndice A: Algoritmo del dispositivo transmisor para la obtención y envío de las variables climáticas

```
1 //Librerías para el modulo LoRa
2 #include <LoRa.h>
3 #include <SPI.h>
4
5 //Librerías para comunicar y dibujar en la pantalla OLED integrada
6 #include <Wire.h>
7 #include <Adafruit_GFX.h>
8 #include <Adafruit_SSD1306.h>
9
10 //Debemos definir los pines que seran utilizadas por el modulo LoRa
11 #define SCK 5
12 #define MISO 19
13 #define MOSI 27
14 #define SS 18
15 #define RST 14
16 #define DIO0 26
17
18 //Definimos la frecuencia de operacion segun nuestra ubicacion
19 #define BAND 915E6
20
21 //Libreria para el sensor DHT22
22 #include <DHT.h>
23 #define DHTTYPE DHT22 // Tipo de sensor DHT
24 DHT dht(2, DHTTYPE); // Identificacion del tipo de sensor
```

```

25
26 //Definimos lo pines del microcontrolador que vamos a utilizar
27 #define DHTPIN 2 // Pin digital al que est conectado el sensor DHT22
28 #define UV_SENSOR_PIN 13 // Pin anal gico para el sensor UV
29 #define SOIL_HUMIDITY_PIN 12 // Pin anal gico para el sensor de humedad del suelo
30
31
32 //Definimos los pines necesarios para conectar con pantalla OLED
33 #define ANCHOPANTALLA 128 // El ancho de la pantalla en pixeles es de 128px
34 #define ALTOPANTALLA 64 // El ancho de la pantalla en pixeles es de 64px
35 #define OLED_SDA 4
36 #define OLED_SCL 15
37 #define OLED_RST 16
38
39 Adafruit_SSD1306 display(ANCHOPANTALLA, ALTOPANTALLA, &Wire, OLED_RST);
40
41 //Hardware pin definitions
42 int uvSensorPin = 34; // Pin anal gico para el sensor UV
43 int REF_3V3 = 25; //3.3V power on the Arduino board
44
45 void setup() {
46
47     Serial.begin(115200); //inicia monitor serial
48
49     pinMode(uvSensorPin, INPUT);
50     pinMode(REF_3V3, INPUT);
51
52     pinMode(OLED_RST, OUTPUT); //reseteamos la pantalla OLED para comenzar
53     digitalWrite(OLED_RST, LOW);
54     delay(20);
55     digitalWrite(OLED_RST, HIGH);
56
57     Wire.begin(OLED_SDA, OLED_SCL); //inicia OLED
58     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) { // 0x3C representa
59         Serial.println(F("Fallo iniciando SSD1306"));
60         for(;;); // Si detecta el fallo anterior, detiene el codigo en este apartado ha
61     }

```



```
62
63 display.clearDisplay(); //Borramos pantalla
64 display.setTextColor(WHITE); //Definimos texto color blanco
65 display.setTextSize(1); //Tamaño de fuente a 1 punto
66 display.setCursor(0,0); //Comenzamos a graficar desde coordenadas 0,0
67 display.print("TRANSMISOR LORA ");
68 display.display();
69
70 Serial.println("Prueba de envío LoRa");
71
72 SPI.begin(SCK, MISO, MOSI, SS); //Definimos pines para la comunicación SPI
73 LoRa.setPins(SS, RST, DIO0); //Configuramos el LoRa para enviar
74
75 if (!LoRa.begin(BAND)) { //Intenta transmitir en la banda seleccionada
76     Serial.println("Error iniciando LoRa"); //Si no puede transmitir, marca error
77     while (1);
78 }
79
80 // Configura la palabra de sincronización para la comunicación LoRa
81 LoRa.setSyncWord(0xF3);
82 // Configura la potencia de transmisión del módulo LoRa a 20 dBm
83 LoRa.setTxPower(20);
84
85 Serial.println("Inicio exitoso de LoRa!"); //Mensaje de todo bien en puerto serial
86 display.setCursor(0,10);
87 display.print("Inicio exitoso de LoRa!"); //Mensaje de todo bien en pantalla OLED
88 display.display();
89 delay(2000); //Esperamos un par de segundos
90
91 dht.begin(); // Inicializar el sensor DHT
92 }
93
94 void loop() {
95     // Medir el nivel de radiación UV
96     int uvLevel = averageAnalogRead(uvSensorPin);
97     int refLevel = averageAnalogRead(REF_3V3);
98     float outputVoltage = 3.3 / refLevel * uvLevel;
```

```
99 //Convierta el voltaje a un nivel de intensidad UV.
100 float uvIntensity = mapfloat(outputVoltage, 0.87, 3.2, 1.0, 15.0);
101
102 // Medir la humedad del suelo
103 int soilHumidityValue = analogRead(SOIL_HUMIDITY_PIN);
104 float soilHumidity = map(soilHumidityValue, 4095, 0, 0, 100);
105
106 // Medir la temperatura y la humedad con el sensor DHT22
107 float temperature = dht.readTemperature();
108 float humidity = dht.readHumidity();
109
110 // Construir el mensaje a enviar
111 byte messageData[24];
112 memcpy(&messageData[0], &uvIntensity, sizeof(float));
113 memcpy(&messageData[4], &soilHumidity, sizeof(float));
114 memcpy(&messageData[8], &temperature, sizeof(float));
115 memcpy(&messageData[12], &humidity, sizeof(float));
116
117 // Enviar el mensaje
118 LoRa.beginPacket();
119 LoRa.write(messageData, sizeof(messageData));
120 LoRa.endPacket();
121
122 Serial.print("UV: ");
123 Serial.print(uvIntensity);
124 Serial.print("mW/cm^2, Humedad del suelo: ");
125 Serial.print(soilHumidity);
126 Serial.print("%, Temperatura: ");
127 Serial.print(temperature);
128 Serial.print(" C , Humedad: ");
129 Serial.print(humidity);
130 Serial.println("%");
131
132 display.clearDisplay();
133 display.setCursor(0, 0);
134 display.print("Transmisor LoRa");
135 display.setCursor(0, 20);
```

```
136 display.print("UV: ");
137 display.print(uvIntensity);
138 display.print("V");
139 display.setCursor(0, 30);
140 display.print("Hum del Suelo: ");
141 display.print(soilHumidity);
142 display.print("%");
143 display.setCursor(0, 40);
144 display.print("Humedad: ");
145 display.print(humidity);
146 display.print("%");
147 display.setCursor(0, 50);
148 display.print("Temperatura: ");
149 display.print(temperature);
150 display.print("C");
151 display.display();
152
153 delay(5000);
154 }
155
156
157 //Toma un promedio de lecturas en un pin determinado
158 //Devuelve el promedio
159 int averageAnalogRead(int pinToRead)
160 {
161     byte numberOfReadings = 8;
162     unsigned int runningValue = 0;
163
164     for(int x = 0 ; x < numberOfReadings ; x++)
165         runningValue += analogRead(pinToRead);
166     runningValue /= numberOfReadings;
167
168     return(runningValue);
169 }
170
171 //La función Arduino Map pero para flotadores
172 //De: http://forum.arduino.cc/index.php?topic=3922.0
```

```
173 float mapfloat(float x, float in_min, float in_max, float out_min, float out_max)
174 {
175     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
176 }
```

Listing 1: Código de programación del dispositivo transmisor

Apéndice B: Algoritmo del dispositivo receptor para recepción y publicación de las variables climáticas

```
1 //Librerías para el módulo LoRa
2 #include <LoRa.h>
3 #include <SPI.h>
4
5 //Librerías para comunicar y dibujar en la pantalla OLED integrada
6 #include <Wire.h>
7 #include <Adafruit_GFX.h>
8 #include <Adafruit_SSD1306.h>
9
10 //Configuración de Conexión WiFi y Cliente MQTT
11 #include <WiFi.h>
12 #include <PubSubClient.h>
13
14 //Debemos definir los pines que serán utilizadas por el módulo LoRa
15 #define SCK 5
16 #define MISO 19
17 #define MOSI 27
18 #define SS 18
19 #define RST 14
20 #define DIO0 26
21
22 //Definimos la frecuencia de operación según nuestra ubicación
23 #define BAND 915E6
24
25 //Definimos los pines necesarios para conectar con pantalla OLED
26 #define ANCHOPANTALLA 128 // El ancho de la pantalla en pixeles es de 128px
```

```
27 #define ALTOPANTALLA 64 // El ancho de la pantalla en pixeles es de 64px
28 #define OLED_SDA 4
29 #define OLED_SCL 15
30 #define OLED_RST 16
31
32 Adafruit_SSD1306 display(ANCHOPANTALLA, ALTOPANTALLA, &Wire, OLED_RST);
33
34 // Configuraci n de MQTT
35
36 // Definir el nombre de la red WiFi y la contrase a
37 const char* ssid = "Mario Penaranda";
38 const char* password = "santabarbara_64";
39
40 // Definir la direcci n del servidor MQTT
41 const char* mqtt_server = "broker.emqx.io";
42
43 // Crear una instancia de cliente WiFi
44 WiFiClient espClient;
45
46 // Crear una instancia de cliente MQTT utilizando la biblioteca PubSubClient
47 PubSubClient client(espClient);
48
49 void setup() {
50     Serial.begin(115200);
51
52     pinMode(OLED_RST, OUTPUT);
53     digitalWrite(OLED_RST, LOW);
54     delay(20);
55     digitalWrite(OLED_RST, HIGH);
56
57     Wire.begin(OLED_SDA, OLED_SCL);
58     if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) {
59         Serial.println(F("Fallo iniciando SSD1306"));
60         for(;;);
61     }
62
63     display.clearDisplay();
```

```
64 display.setTextColor(WHITE);
65 display.setTextSize(1);
66 display.setCursor(0, 10);
67 display.print("RECEPTOR LORA ");
68 display.display();
69
70 Serial.println("Prueba de recepci n LoRa");
71
72 // Inicia la comunicaci n SPI con pines espec ficos
73 SPI.begin(SCK, MISO, MOSI, SS);
74
75 // Configura los pines para el m dulo LoRa
76 LoRa.setPins(SS, RST, DIO0);
77
78 // Intenta iniciar la comunicaci n LoRa en la frecuencia de banda especificada
79 if (!LoRa.begin(BAND)) {
80     Serial.println("Error iniciando LoRa");
81     while (1);
82 }
83
84 // Configura la palabra de sincronizaci n para la comunicaci n LoRa
85 LoRa.setSyncWord(0xF3);
86
87 // Imprime un mensaje de inicio exitoso en el puerto serial
88 Serial.println("Inicio exitoso de LoRa!");
89 // Configura la posici n del cursor en la pantalla OLED
90 display.setCursor(0, 30);
91 // Imprime un mensaje de inicio exitoso en la pantalla OLED
92 display.print("Inicio exitoso de LoRa!");
93 // Muestra el contenido de la pantalla OLED
94 display.display();
95 delay(2000);
96
97 // Inicializaci n de la conexi n WiFi
98 setup_wifi();
99
100 // Inicializaci n de la conexi n MQTT
```

```
101 client.setServer(mqtt_server, 1883); // Establece el servidor MQTT y el puerto
102 client.setCallback(callback); // Asigna la función de retorno de llamada
103 reconnect(); // Intenta reconectar al servidor MQTT
104 }
105
106 void loop() {
107     // Obtiene el tamaño del paquete recibido a través de LoRa
108     int tamañoPaquete = LoRa.parsePacket();
109
110     if (tamañoPaquete) {
111         Serial.print("Paquete recibido ");
112
113         // Ajusta el tamaño del array según la longitud de tus datos
114         byte messageData[24];
115         while (LoRa.available()) {
116             LoRa.readBytes(messageData, sizeof(messageData));
117         }
118
119         // Extrae los valores del array de bytes
120         float uvIntensity, soilHumidity, temperature, humidity;
121
122         // Copia los datos del array de bytes a las variables correspondientes
123         memcpy(&uvIntensity, &messageData[0], sizeof(float));
124         memcpy(&soilHumidity, &messageData[4], sizeof(float));
125         memcpy(&temperature, &messageData[8], sizeof(float));
126         memcpy(&humidity, &messageData[12], sizeof(float));
127
128         // Muestra los valores recibidos
129         Serial.print("UV: ");
130         Serial.print(uvIntensity);
131         Serial.print("mW/cm^2, Humedad del suelo: ");
132         Serial.print(soilHumidity);
133         Serial.print("%, Temperatura: ");
134         Serial.print(temperature);
135         Serial.print(" C , Humedad: ");
136         Serial.print(humidity);
137         Serial.println("%");
```

```
138
139 // Obtiene el valor de intensidad de se al RSSI del paquete LoRa
140 int rssi = LoRa.packetRssi();
141 Serial.print(" con RSSI ");
142 Serial.println(rssi);
143
144 // Publica los datos en los t picos MQTT deseados
145 publishData(temperature, humidity, uvIntensity, soilHumidity);
146
147 // Muestra informaci n captada en la pantalla OLED
148 display.clearDisplay();
149 display.setCursor(0, 0);
150 display.print("Receptor LoRa");
151 display.setCursor(0, 20);
152 display.print("UV: ");
153 display.print(uvIntensity);
154 display.print("V");
155 display.setCursor(0, 30);
156 display.print("Hum del Suelo: ");
157 display.print(soilHumidity);
158 display.print("%");
159 display.setCursor(0, 40);
160 display.print("Humedad: ");
161 display.print(humidity);
162 display.print("%");
163 display.setCursor(0, 50);
164 display.print("Temperatura: ");
165 display.print(temperature);
166 display.print("C");
167 display.display();
168 }
169
170 // Realiza la conexi n MQTT si es necesario
171 if (!client.connected()) {
172     reconnect();
173 }
174
```



```
175 // Maneja los mensajes MQTT entrantes
176 client.loop();
177
178 }
179
180 void publishData(float temperature, float humidity, float uvIntensity, float soilHumidity) {
181 // Publica cada variable por separado en su t p ico MQTT correspondiente
182 client.publish("temperature_topic", String(temperature).c_str());
183 client.publish("humidity_topic", String(humidity).c_str());
184 client.publish("uv_topic", String(uvIntensity).c_str());
185 client.publish("soil_humidity_topic", String(soilHumidity).c_str());
186 }
187
188 void setup_wifi() {
189 delay(10);
190 Serial.println();
191 Serial.print("Conectando a WiFi");
192
193 // Inicia el proceso de conexi n a la red WiFi
194 WiFi.begin(ssid, password);
195
196 // Espera hasta que se establezca la conexi n
197 while (WiFi.status() != WL_CONNECTED) {
198 delay(500);
199 Serial.print(".");
200 }
201
202 // Imprime la informaci n de la conexi n exitosa
203 Serial.println("");
204 Serial.println("WiFi conectado");
205 Serial.println("Direcci n IP: ");
206 Serial.println(WiFi.localIP());
207 }
208
209 void reconnect() {
210 // Intenta conectar al servidor MQTT
211 while (!client.connected()) {
```

```
212     Serial.print("Intentando conexión MQTT...");
213
214     // Genera un ID de cliente único
215     String clientId = "ESP32Client-";
216     clientId += String(random(0xffff), HEX);
217
218     // Intenta conectar con el ID de cliente generado
219     if (client.connect(clientId.c_str())) {
220         Serial.println("Conectado al servidor MQTT");
221     } else {
222         Serial.print("falló, rc=");
223         Serial.print(client.state());
224         Serial.println(" intentando de nuevo en 5 segundos");
225         delay(5000);
226     }
227 }
228 }
229
230 void callback(char* topic, byte* payload, unsigned int length) {
231     // Maneja los mensajes MQTT entrantes si es necesario
232     // Puedes agregar tu lógica aquí para manejar mensajes específicos
233 }
```

Listing 2: Código de programación del dispositivo receptor