



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERIA EN CIENCIAS  
APLICADAS**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
REDES DE COMUNICACIÓN**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN  
DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y  
REDES DE COMUNICACIÓN**

**TEMA: “SISTEMA DE MONITOREO DE CULTIVOS DE  
ARÁNDANOS BASADO EN Wi-SUN FAN MESH NETWORK”**

**Autora: JOHANNA ESTEFANÍA YÉPEZ VÁSQUEZ**

**Director: MSC. EDGAR ALBERTO MAYA OLALLA**

**IBARRA - ECUADOR**

**2023**



## UNIVERSIDAD TÉCNICA DEL NORTE

### BIBLIOTECA UNIVERSITARIA

#### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004499776		
APELLIDOS Y NOMBRES:	YÉPEZ VÁSQUEZ JOHANNA ESTEFANÍA		
DIRECCIÓN:	RIO VINCES 1-63		
EMAIL:	jeyopezv@utn.edu.ec		
TELÉFONO FIJO:	-	TELÉFONO MÓVIL:	0979120338

DATOS DE LA OBRA	
TÍTULO:	SISTEMA DE MONITOREO DE CULTIVOS DE ARÁNDANOS BASADO EN Wi-SUN FAN MESH NETWORK
AUTOR (ES):	YÉPEZ VÁSQUEZ JOHANNA ESTEFANÍA
FECHA: DD/MM/AAAA	17/02/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	SISTEMA DE MONITOREO DE CULTIVOS DE ARÁNDANOS BASADO EN Wi-SUN FAN MESH NETWORK
ASESOR /DIRECTOR:	MSC. EDGAR MAYA

### 2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 24 días del mes de febrero de 2023

EL AUTOR:

Yépez Vásquez Johanna Estefanía



## UNIVERSIDAD TÉCNICA DEL NORTE

### FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

#### CERTIFICACIÓN

MAGISTER EDGAR MAYA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente Trabajo de Titulación “SISTEMA DE MONITOREO DE CULTIVOS DE ARÁNDANOS BASADO EN Wi-SUN FAN MESH NETWORK”. Ha sido desarrollado por la señorita Yépez Vásquez Johanna Estefanía, bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

.....  
MSc. Edgar Maya  
DIRECTOR

**DEDICATORIA**

*A mi familia y amigos.*

## **AGRADECIMIENTOS**

*Primeramente, quiero agradecer a mis padres y hermanos por el todo el apoyo y por su paciencia en el tiempo que me tomó cursar mis estudios, agradecerles por su ejemplo y enseñanza que me lo supieron entregar a diario hasta culminar esta etapa.*

*Seguidamente agradecer a todos quienes también formar parte de esta aventura y me apoyaron como familia. A Adri, Gabriela, Mateo, Juan, Kenan y Francisco amigos con los que compartí y comparto a diario todos los buenos y malos momentos.*

*Finalmente, agradecer a mi tutor del proyecto, por su dedicación, guía y tiempo brindado durante el desarrollo de este trabajo de titulación.*

*Gracias a cada uno.*

## Contenido

Contenido.....	6
RESUMEN .....	12
ABSTRACT.....	13
Capítulo I .....	14
Antecedentes .....	14
1.1 Tema .....	14
1.2 Planteamiento del Problema .....	14
1.3 Objetivos de la Investigación .....	15
1.3.1 Objetivo General. ....	15
1.3.2 Objetivos Específicos .....	15
1.4 Alcance .....	15
1.5 Justificación .....	16
Capítulo 2.....	18
Fundamentación Teórica .....	18
2.1 Red de Sensores.....	18
2.1.1 Aplicaciones de WSN .....	18
2.1.2 Problemas de diseño de una red de sensores inalámbricos (WSN).....	19
2.2 Tipo de Topología .....	20
2.2.1 Red en estrella .....	20
2.2.2 Red de malla.....	21
2.2.3 Híbrida Estrella - Red de malla .....	22
2.3 Estructura de comunicación de una red de sensores inalámbricos.....	23
2.4 Estándar IEEE 802.15.4.....	24
2.4.1 Descripción General .....	25
2.4.2 Formato de la Trama .....	25
2.4.3 Campos de Aplicación .....	26
2.4.4 Componentes de IEEE 802.15.4 WPAN.....	26
2.4.5 Gestión de PHY múltiple (MPM) del SUN WPAN.....	26
2.4.6 Arquitectura.....	27
2.5 RIOT-OS .....	28
2.6 Agricultura de Precisión .....	30
2.6.1 Las ventajas de la agricultura de precisión.....	30

2.6.2	Herramientas de recopilación de datos.....	31
2.6.3	Análisis en agricultura de precisión .....	31
2.7	Cultivo de Arándanos .....	33
2.7.1	¿Cuándo plantar arándanos?.....	33
2.7.2	Elegir y preparar un lugar de plantación .....	33
2.7.3	¿Cómo Plantar Arándanos?.....	34
2.8	Cloud Computing .....	35
2.8.1	Brokers de MQTT y guía de alojamiento en la nube .....	35
2.8.2	Servidor propio instalado localmente.....	35
2.8.3	Instalar Broker en un servidor virtual .....	36
Capítulo 3	.....	37
	Diseño de Hardware y Software.....	37
3.1	Metodología.....	37
3.1.1	Estudio de viabilidad (Feasibility Study).....	37
3.1.2	Análisis y especificación de requisitos (Requirements analysis and specification).....	38
3.1.3	Diseño (Design).....	38
3.1.4	Codificación y pruebas unitarias (Coding and Unit testing).....	38
3.1.5	Integración y prueba del sistema (Integration and System testing) .....	38
3.1.6	Mantenimiento (Maintenance) .....	38
3.2	Análisis .....	38
3.2.1	Situación Actual del Invernadero .....	39
3.3	Métodos y Técnicas de Investigación.....	39
3.3.1	Método de la Investigación .....	40
3.3.2	Técnica de la Investigación .....	40
3.3.3	Resultados de la Entrevista .....	41
3.4	Descripción General del Sistema.....	42
3.5	Determinación de los Stakeholders .....	43
3.6	Requerimientos para el diseño.....	43
3.6.1	Nomenclatura de requerimientos .....	43
3.6.2	Stakeholders requeridos .....	44
3.6.3	Requerimientos del funcionamiento del sistema.....	44
3.6.4	Requerimientos de Arquitectura.....	45
3.7	Hardware y Software del sistema .....	46
3.7.1	Elección de Hardware .....	46

3.7.2	Elección de Sensores .....	52
3.7.3	Elección de Software.....	58
3.8	Ubicación de los Nodos .....	63
3.9	Diseño del Prototipo .....	66
3.10	Arquitectura General del Sistema.....	76
3.10.1	Arquitectura de Internet of Things .....	78
3.10.2	Capa de Red .....	83
3.10.3	Capa de Servicio / Procesamiento de Datos.....	85
3.10.4	Capa de Aplicaciones .....	88
Capítulo 4: Implementación y Pruebas .....		92
4.1	Implementación en el Invernadero .....	92
4.2	Pruebas de Sensores.....	95
4.2.1	Lectura de Sensores.....	95
4.2.2	Análisis de Paquetes de Wireshark .....	97
4.2.3	Pruebas de Distancias .....	100
4.2.4	Análisis con SDR .....	104
Conclusiones .....		107
Recomendaciones .....		109
Bibliografía .....		110
Anexos (Hojas de Datos/ Código/ Encuesta/ Entrevista) .....		113



## INDICE DE FIGURAS

Figura 1. Red de Estrella. Fuente: (Matin & Islam, 2012). .....	21
Figura 2. Red de Malla. Fuente: (Matin & Islam, 2012). .....	22
Figura 3. Red de Híbrida Estrella- Malla. Fuente: (Matin & Islam, 2012).....	23
Figura 4. Stack de Protocolos de WSN. Fuente: (Matin & Islam, 2012). .....	24
Figura 5. Formato de Estructura de Datos General. Fuente: (IEEE Computer Society, 2020). .....	26
Figura 6. Arquitectura del dispositivo LR-WPAN. Fuente: (IEEE Computer Society, 2020). .....	28
Figura 7. Modelo en Cascada. Fuente: (Adenowo & Adenowo, 2013).....	37
Figura 8. Extensión del Invernadero, visto desde Google Earth.....	39
Figura 9. Estructura de RIOT-OS. Fuente: (Baccelli, Hahm, Günes, Wählisch, & Schmidt, 2013). .....	59
Figura 10. Ubicación en el mapa. ....	63
Figura 11. Coordenadas de los sensores. ....	64
Figura 12. Ubicación de cada sensor. ....	64
Figura 13. Añadir tipo de Red.....	65
Figura 14. Línea de Vista y Zona de Fresnel Nodo 1. ....	66
Figura 15. Línea de Vista y Zona de Fresnel Nodo 2. ....	66
Figura 16. Diagrama de Bloques Nodos Sensores. ....	67
Figura 17. Diagrama de Bloques Borde Router.....	67
Figura 18. Diagrama de Flujo Nodo de Sensores 1. ....	68
Figura 19. Diagrama de Flujo Nodo Sensor 2. ....	69
Figura 20. Diagrama de Flujo de los Nodos Sensores en OpenMote-B. ....	70
Figura 21. Diagrama Circuitual Nodo Sensor 1.....	71
Figura 22. Diagrama Circuitual Nodo Sensor 2.....	71
Figura 23. Diagrama de Flujo Borde Router. ....	72
Figura 24. Diagrama de Bloques Broker MQTTSN y Puente MQTT.....	73
Figura 25. Diagrama de Flujo Broker MQTTSN y Puente MQTT. ....	74
Figura 26. Diagrama de bloques Google Cloud, Mongo DB. ....	75
Figura 27. Diagrama de Flujo de Publicación y Visualización de Datos. ....	76
Figura 28. Arquitectura del Sistema. ....	77
Figura 29. Nodo Sensor 1 ubicado en maceta.....	78
Figura 30. Nodo Sensor 2 ubicado en maceta.....	79
Figura 31. Código de Arduino Nodo Sensor 1 lectura de datos. ....	80
Figura 32. Código de Arduino Nodo Sensor 2 lectura de datos. ....	81
Figura 33. Código de la Recepción de datos por Comunicación Serial.....	81
Figura 34. Sentencias de Configuración de Modulación en Open Mote B. ....	83
Figura 35. Configuración del Enrutamiento RPL para los Nodos. ....	84
Figura 36. Código del enrutamiento hacia el Broker MQTT-SN. ....	84
Figura 37. Configuración del enrutamiento de la tap. ....	85
Figura 38. Código del proceso de publicación en el Broker MQTT SN. ....	85
Figura 39. Configuración de los puertos abiertos en el Broker. ....	86
Figura 40. Códigos para añadir credenciales para el Cloud.....	86
Figura 41. Código de espera de un Mensaje Recibido. ....	87
Figura 42. Log de Registro, Conexión y Publicación del Broker RSMB.....	87

Figura 43. Código para la conexión a la Base de Datos. ....	88
Figura 44. Configuración credenciales de suscripción a Google Cloud en Página Web.....	88
Figura 45. Código de la recepción y separación de etiquetas en la Página Web. ....	89
Figura 46. Código para la visualización de Datos con Handlebar. ....	89
Figura 47. Configuración de Web Socket para actualización de tiempos. ....	90
Figura 48. Ejemplo de la Interfaz de Usuario. ....	91
Figura 49. Ubicación de Sensores en la maceta. ....	92
Figura 50. Nodo de Sensor 1 ubicado en la maceta. ....	93
Figura 51. Nodo de Sensores 2 ubicado en la maceta.....	93
Figura 52. (a) Broker RSMB, (b) Puente Transparente, (c) Borde Router, (d) Enrutamiento tap5.....	94
Figura 53. Paquetes JSON recibidos en la Nube. ....	95
Figura 54. Base de Datos en MongoDB. ....	95
Figura 55. Visualización de datos de sensores dentro de los rangos adecuados.....	96
Figura 56. Visualización de datos de sensores fuera de rangos adecuados. ....	96
Figura 57. Log de paquetes de Conexión.....	97
Figura 58. Visualización de paquete de Conexión en Wireshark. ....	97
Figura 59. Log de paquetes de Registro.....	98
Figura 60. Visualización de captura de Registro en Wireshark.....	98
Figura 61. Log de paquetes de Publicación. ....	99
Figura 62. Visualización de paquete de Publicación en Wireshark.....	99
Figura 63. Log de paquetes de Desconexión. ....	100
Figura 64. Visualización de captura de Desconexión en Wireshark. ....	100
Figura 65. Captura de ping a los 10 metros. ....	101
Figura 66. Resumen de Estadísticas de envío de ping a los 10 metros.....	101
Figura 67. Captura de ping a los 50 metros. ....	102
Figura 68. Resumen de Estadísticas de envío de ping a los 50 metros.....	102
Figura 69. Captura de ping a los 100 metros. ....	103
Figura 70. Resumen de Estadísticas de envío de ping a los 100 metros.....	103
Figura 71. Captura de ping a los 150 metros. ....	104
Figura 72. Espectro de la Señal.....	105
Figura 73. Análisis del espectro de la señal por muestra. ....	105
Figura 74. Análisis de dos paquetes con su ACK. ....	106

## INDICE DE TABLAS

Tabla 1. Brokers más populares.....	35
Tabla 2. Lista de Stakeholders.....	43
Tabla 3. Nomenclatura de requerimientos.....	43
Tabla 4. Requerimientos de Stakeholders.....	44
Tabla 5. Requerimientos de Funcionamiento.....	45
Tabla 6. Requerimientos de Arquitectura.....	45
Tabla 7. Características Texas Instruments CC1352P.....	47
Tabla 8. Características Texas Instruments CC1352P.....	48
Tabla 9. Características Técnicas del Open Mote - B.....	48
Tabla 10. Selección de Nodo de Sensores.....	49
Tabla 11. Especificaciones Arduino UNO.....	50
Tabla 12. Especificaciones Arduino MEGA.....	51
Tabla 13. Especificaciones Arduino Leonardo.....	52
Tabla 14. Selección de Nodo de Sensores.....	52
Tabla 15. Características del Sensor de pH SEN0161.....	53
Tabla 16. Especificaciones Sensor SEN0169.....	54
Tabla 17. Selección de Sensor de PH.....	54
Tabla 18. Características del Sensor DHT22.....	55
Tabla 19. Características del Sensor DHT11.....	56
Tabla 20. Selección de Sensor de Temperatura.....	56
Tabla 21. Características del Sensor de Humedad de Suelo.....	57
Tabla 22. Características del Sensor de Humedad de Suelo.....	57
Tabla 23. Selección de Sensor de Humedad de Suelo.....	57
Tabla 24. Características de la placa compatible en RIOT.....	59
Tabla 25. Selección de Software.....	60
Tabla 26. Selección de Software.....	62
Tabla 27. Parámetros de Simulación Radio Mobile.....	65
Tabla 28. Tabla de Canales.....	82
Tabla 29. Subportadoras que se utilizan en un canal.....	82

## RESUMEN

En este trabajo se detalla el diseño de un prototipo para monitoreo de cultivos de arándanos basado en Wi-Sun Fan Mesh Network, con el propósito de verificar los parámetros de PH, Humedad Relativa del Suelo y Temperatura. El cultivo de Arándanos en Ibarra es una de las actividades económicas que más remuneración aporta; sin embargo, el proceso de cultivación de las plantas requiere mucho cuidado en un invernadero que no cuenta con un sistema de monitoreo. En la actualidad, se usan sistemas de riego automático para mejorar la calidad de la fruta, así como para ahorrar muchos recursos del cultivo.

Dados estos eventos los cultivos carecen de un entorno que sea adecuado tanto para su producción como para desarrollar, además el sistema conlleva demasiada intervención de los dueños del Invernadero que pueden resultar en error que arruinen la producción de las mismas.

El principal propósito para contribuir a la mejora de este cultivo se presenta este proyecto que consiste en que se apliquen los conocimientos adquiridos a lo largo del estudio de la carrera y además una investigación exhaustiva para los conceptos de un sistema de monitoreo de los parámetros ambientales que están involucrado en este cultivo de arándanos, con el motivo de lograr aprovechar la tecnología dentro de la agricultura para poder analizar cada uno de los parámetros que se han estudiado.

Una vez se analizan todos estos parámetros y datos recolectados se genera el prototipo, el cual se implementó en el Barrio San Pedro de la Esperanza, donde se analizó las condiciones del cultivo y se enviaron a una plataforma Web donde se podía visualizar las gráficas sobre el PH, Humedad Relativa del Suelo y la Temperatura del Ambiente en cada una de macetas de arándanos. Dando como resultado una aplicación que permitía al encargado del cultivo tomar las acciones requeridas según el estado del cultivo.

## **ABSTRACT**

This paper details the design of a prototype for monitoring blueberry crops based on Wi-Sun Fan Mesh Network, with the purpose of verifying the parameters of PH, Soil Relative Humidity and Temperature. The cultivation of blueberries in Ibarra is one of the economic activities that contributes the most remuneration; however, the process of cultivating the plants requires great care in a greenhouse that does not have a monitoring system. At present, automatic irrigation systems are used to improve the quality of the fruit, as well as to save many crop resources.

Given these events, the crops lack an environment that is suitable for both their production and development, in addition, the system involves too much intervention from the owners of the Greenhouse that can result in errors that ruin their production.

The main purpose to contribute to the improvement of this crop is presented in this project that consists of applying the knowledge acquired throughout the study of the career and also an exhaustive investigation for the concepts of a monitoring system of environmental parameters that are involved in this blueberry crop, with the aim of taking advantage of technology within agriculture to be able to analyze each of the parameters that have been studied.

Once all these parameters and collected data are analyzed, the prototype is generated, which was implemented in the San Pedro de la Esperanza neighborhood, where the crop conditions were analyzed and sent to a Web platform where the graphs on the PH, Relative Soil Humidity and Ambient Temperature in each of the blueberry pots. Resulting in an application that allowed the person in charge of the crop to take the actions required according to the state of the crop.

## Capítulo I

### Antecedentes

#### 1.1 Tema

SISTEMA DE MONITOREO DE CULTIVOS DE ARÁNDANOS BASADO EN Wi-SUN FAN MESH NETWORK

#### 1.2 Planteamiento del Problema

Se han realizado varios estudios dentro de la agricultura que han demostrado que a medida que el tiempo avanza se requiere más producción, la Organización de las Naciones Unidas para la Alimentación reveló que para el 2050 se necesitarán más del 50% de la producción agrícola actual. El clima en Ibarra, así como en diferentes ciudades del Ecuador, es constantemente cambiante e impredecible, motivo por el cual es ideal para la producción de Arándanos y de esa forma se produzcan adecuadamente y en su totalidad. La Parroquia San José de Chaltura posee una zona con condiciones agroecológicas, que cuentan con ciertas ventajas climáticas, geográficas a más de la disponibilidad de terrenos para la siembra y vías de acceso en buen estado que la convierte en una zona apta para la producción agrícola, especialmente de arándanos. Por otro lado, debido a las condiciones y propiedades de los arándanos, se requieren muchos cuidados y un estricto control del estado de la planta, para de esa forma obtener una fruta de calidad y sin desperdicio de productos en ella.

Los arándanos son relativamente fáciles de cuidar, siempre y cuando se mantengan en el ambiente adecuado, asegurándose de que el suelo tenga un pH constante de 5,5 o menos, para evitar que las plantas desarrollen clorosis inducida por la cal y la deficiencia de hierro asociada.

“En cuanto a la temperatura el arándano es un cultivo que requiere un determinado número de horas-frío (temperatura inferior a 7°C) para salir de la latencia, que depende de la especie. Para el desarrollo del cultivo del arándano, el rango óptimo de temperatura oscila entre 16-25°C., finalmente en el cultivo del arándano se requiere de humedad relativa alta” (Jing, y otros, 2017).

Es por eso por lo que con este proyecto se propone el monitoreo de las condiciones de dicho cultivo, censando variables como la temperatura del suelo, el PH del agua y la Humedad Relativa del suelo de dicho cultivo para lograr reducir el desperdicio de arándanos a menores riesgos y gastos, usando nuevas tecnologías como es Wi-SUN que

se basa en el estándar IEEE 802.15.4g para la comunicación en recepción y transmisión de los datos, entre los distintos nodos de sensores y los nodos de procesamiento donde se administrará, analizará y monitoreará las condiciones de los cultivos de Arándanos, implementando así una comunicación óptima entre todos los dispositivos y un menor consumo de energía.

### ***1.3 Objetivos de la Investigación***

#### **1.3.1 Objetivo General.**

Diseñar un sistema de monitoreo de las condiciones del cultivo usando redes de sensores (WSN) basados en Wi-SUN FAN Mesh para permitir el monitoreo de los agentes ambientales que se presentan en cada zona de la plantación de Arándanos y se puedan tomar decisiones según sea el caso.

#### **1.3.2 Objetivos Específicos**

- Definir y recopilar información sobre los agentes que intervienen en el sembrío y cosecha de los Arándanos y el entorno en cada bloque de plantación, pudiendo identificar la mejor ubicación para cada nodo de sensor y la cobertura de los mismos.
- Diseñar el esquema de las redes de sensores que contenga el hardware de monitoreo de todos los parámetros identificados, para que se permita la recolección de la información de cada sensor y activar un sistema de alerta.
- Proponer un software de monitoreo para temperatura del suelo, humedad relativa del suelo, y PH del cultivo de Arándanos en el invernadero utilizando herramientas web que posibilitan el almacenamiento de los datos en la Nube.
- Realizar todas las pruebas de funcionamiento y además analizar todos los indicadores de rendimiento de la arquitectura para demostrar la correcta transmisión y recepción de los datos entre la red de sensores.

### ***1.4 Alcance***

Para el diseño de este sistema de monitoreo se centrará en obtener los mejores resultados tanto para el cultivo como para los servicios de la red de sensores y lograr el mejor resultado posible en ambas partes. Para ello se utilizará la metodología de Kanban y puede ser aplicada para este caso ya que se presentan situaciones impredecibles en los cultivos de los Arándanos que van a necesitarse revisar y solucionar en ese momento

para que sea eficiente el sistema y se cumpla al 100% con los objetivos del mismo. Es por eso que se requieren diferentes fases a cumplir:

En primer lugar, se deberá comprender y analizar todos los factores que afectan a un cultivo de Arándanos y aquellos parámetros que no permiten la producción en su totalidad y de esa forma se pierden muchas plantaciones. Como segundo parámetro se requiere determinar todos los componentes y tecnologías que serán usadas dentro del diseño de la arquitectura, por ello se ha contemplado el uso de la tecnología Wi-SUN para la transmisión y recepción de los datos entre los nodos de comunicación. Se realiza la investigación de nodos compatibles con esta tecnología y que permitan la correcta adaptación al sistema de la manera óptima posible.

El siguiente parámetro es la alerta, una vez analizados los datos se procede a generar una alerta al agricultor o aplicar acciones según sea el caso. Este proceso se lo realiza en la plataforma AWS IoT en donde se procesan los datos y se envía al Servidor Web el cual será montado en la plataforma AWS Cloud generando así las alertas para el cultivo; con la ayuda del sistema el agricultor se recibirá notificaciones cualquier momento informando sobre el posible riesgo en el bloque de cultivo; es decir en la planta, y así se pueda tomar las acciones oportunas dentro del mismo así como sugerencias del mejor momento para fertilizar, además de obtener un registro completo de datos desde la siembra hasta la cosecha y optimizar el uso de fertilizante y recursos hídricos.

Por último, “se requerirá analizar todas las métricas de calidad de servicios que comprobarán el funcionamiento así como las KPIs o Indicadores de Clave de Rendimiento que son Throughput, Latencia, Jitter y PER, según la Normativa expuesta en el Estándar IEEE 1900.2 y 1900.1 las cuales se enfocan en bloques de construcción arquitectónicos que permiten la toma de decisiones distribuida por dispositivos de red para el uso optimizado de recursos de radio en redes de acceso inalámbrico heterogéneas” (IEEE, 2019). Además, las basándose en la RFC 2544 la cual expone la metodología de evaluación comparativa para dispositivos de interconexión de red. (Bradner, 1999)

### ***1.5 Justificación***

En la búsqueda de un sustento económico y familiar, los agricultores en Chaltura han optado por la producción de Arándanos para la exportación del mismo y obtener beneficios económicos de ello, sin embargo la siembra y cosecha del mismo puede



resultar en una inversión bastante alta teniendo en cuenta que requiere mucho cuidado para obtener el producto en su totalidad y así mismo requiere implementar mucho tiempo para la cosecha del producto, es por estas razones que la mayoría de agricultores tienden a no realizar las plantaciones de Arándanos.

Es por ello que realizar un monitoreo del ambiente en el que se encuentran las plantaciones permite que su crecimiento sea adecuado además de evitar la proliferación de plagas en el cultivo y evitar que las plantas desarrollen clorosis inducida por la cal y la deficiencia de hierro asociada. Para ello se requiere control de temperatura, humedad y PH del suelo teniendo en cuenta todos estos parámetros se necesita también un método eficiente de recolección de estos datos y la transferencia de estos hacia un dispositivo capaz de computar y alertas al agricultor sobre las acciones que debe realizar dentro del cultivo para que no pierda producción del mismo.

“Según el objetivo 3 del Plan Nacional del Buen Vivir el cual establece como prioridad Mejorar la calidad de vida de la población y con el desarrollo de proyectos que mejoren la calidad de los mismo, con el Sistema de Monitoreo y Control se espera mejorar la calidad de los Arándanos siendo cosechados de forma natural con condiciones ambientales adecuadas para dicho proceso estos a su vez mejorarán la calidad de vida de dichos beneficiarios” (República Del Ecuador Consejo Nacional De Planificación, 2013).

Por otro lado, se lleva a realizar este proyecto para el aporte al crecimiento de la economía del agricultor y del país, con la ayuda de tecnologías innovadoras aplicadas al sector agrícola implementando una red de sensores (WSN), motivando a dejar de lado las tecnología o metodologías obsoletas siendo las actuales más eficientes con respecto a la conservación de energía, protección del medio ambiente y optimización de recursos.

## Capítulo 2

### Fundamentación Teórica

En este capítulo se va a describir los conceptos, características, elementos y diferentes parámetros a tomar en cuenta para el desarrollo de la red de sensores, juntamente con las distintas topologías, en el cual se describirá el objetivo principal del uso del estándar de IEEE 802.15.4 que de la misma forma se presentarán todos sus componentes y características. Para posteriormente señalar todos los conceptos a usarse dentro de la agricultura de precisión, para optimizar la calidad de producción con la ayuda de dispositivos que nos permitan el cumplimiento de dicho objetivo.

#### 2.1 Red de Sensores

Las redes inalámbricas de sensores a lo largo de cada año han ido en incremento y han sido incluidas tanto en el entorno industrial como social; sin embargo, aún existen varios retos de desarrollo para poder garantizar el funcionamiento de ambientes críticos.

“Estas redes se pueden definir como autoconfiguradas y sin infraestructura para monitorear las condiciones físicas o ambientales; tales como, temperatura, sonido, vibración, presión movimiento, etc. Para posteriormente pasar los datos de manera cooperativa sus datos a través de la red a una ubicación principal donde se pueden analizar y visualizar dichos datos” (Lazarescu, 2013).

Comúnmente, una red de sensores inalámbrica contiene cientos de miles de nodos sensores, los cuales pueden comunicarse entre mediante el uso de señales de radio, y está equipado con dispositivos de detección y computación, transceptores de radio y componentes de energía

“Las redes de sensores inalámbricos (WSN) permiten nuevas aplicaciones y requieren paradigmas no convencionales para el diseño de protocolos debido a varias limitaciones. Debido al requisito de una baja complejidad del dispositivo junto con un bajo consumo de energía (es decir, una larga vida útil de la red), se debe encontrar un equilibrio adecuado entre las capacidades de comunicación y procesamiento de señales/datos. Esto motiva un gran esfuerzo en actividades de investigación, procesos de estandarización e inversiones industriales en este campo desde la última década” (Matin & Islam, 2012).

##### 2.1.1 Aplicaciones de WSN

Las redes de sensores inalámbricos han ganado una popularidad considerable debido a su flexibilidad para resolver problemas en diferentes dominios de aplicación y tienen el potencial de cambiar nuestras vidas de muchas formas diferentes. Las WSN se han aplicado con éxito en varios dominios de aplicación, tales como:

- *Aplicaciones Militares:* gran parte de los sistemas de comando, control, comunicación, computación, inteligencia, vigilancia en el campo de batalla, etc.
- *Áreas de monitoreo:* monitoreo de áreas de interés, pueden ser desplegadas sobre regiones donde se necesita el monitoreo de algún fenómeno, al detectar el evento se informa a una estación base.
- *Transporte:* La WSN puede recopilar información sobre el tráfico en tiempo real para alimentar los modelos de transporte y alertar a los conductores sobre la congestión de las vías y problemas de tráfico.
- *Aplicaciones en la Salud:* entre ellas interfaces de soporte para discapacitados, monitoreo integrado de pacientes, telemonitorización de datos fisiológicos humanos y seguimiento de médicos a pacientes dentro del hospital.
- *Detección Ambiental:* “para cubrir muchas investigaciones de la ciencia hacia la tierra, esto incluye la detección de volcanes, océanos, glaciares, entre otros” (Herrera, Dien Tran, Egas, Pereira, & Sá Silva, 2016).

### **2.1.2 Problemas de diseño de una red de sensores inalámbricos (WSN)**

Se plantean muchos desafíos El despliegue de redes de sensores plantea muchos desafíos, que son un superconjunto de los que se encuentran en las redes inalámbricas ad hoc. Los nodos de sensores se van a comunicar a través de líneas inalámbricas sin infraestructura, adicionalmente se relaciona el suministro de energía limitado (Matin & Islam, 2012). Para maximizar la vida útil de la red, los protocolos deben diseñarse desde el principio con el objetivo de una gestión eficiente de los recursos energéticos, entre los mayores problemas se encuentran:

- *Tolerancia a fallas:* los nodos son vulnerables y suelen ser implementados en entornos peligrosos, pueden fallar debido a problemas de hardware, daños físicos o el agotamiento de suministro de energía.
- *Escalabilidad:* la escala varía de varios nodos hasta potencialmente varios cientos de miles; además, la densidad de implementación también es variable. Los protocolos implementados en las redes de sensores deben ser escalables a estos niveles y poder mantener un rendimiento adecuado.

- *Costo de producción:* debido a que muchos modelos de implementación consideran que los nodos de sensores son dispositivos desechables, las redes de sensores pueden competir con los enfoques tradicionales de recopilación de información solo si los nodos de sensores individuales se pueden producir a muy bajo costo. El precio objetivo previsto para un nodo sensor idealmente debería ser inferior a 1 dólar.
- *Restricciones de Hardware:* Cada nodo sensor debe tener una unidad de detección, una unidad de procesamiento, una de transmisión y una fuente de alimentación. Opcionalmente los nodos pueden tener varios sensores integrados o dispositivos adicionales.
- *Topología de red de sensores:* “De estas limitaciones, el consumo de energía es de suma importancia, lo que se demuestra por la gran cantidad de algoritmos, técnicas y protocolos que se han desarrollado para ahorrar energía y, por lo tanto, extender la vida útil de la red. El mantenimiento de la topología es uno de los temas más importantes investigados para reducir el consumo de energía en las redes de sensores inalámbricos” (Herrera, Dien Tran, Egas, Pereira, & Sá Silva, 2016).
- *Medios de Transmisión:* entre nodos se suelen implementar normalmente mediante comunicación por radio a través de bandas ISM.
- *Consumo de Energía:* Muchos de los desafíos de las WSN gira en torno a los recursos de energía, el tamaño de los nodos limita el tamaño de la batería y debe ser cuidadosamente considerado el uso eficiente de la energía.

## **2.2 Tipo de Topología**

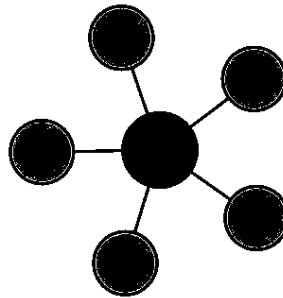
La estructura de una red de sensores inalámbricos incluye diferentes topologías para redes de comunicaciones por radio. A continuación, se describe una breve discusión de las topologías de red que se aplican a las redes de sensores inalámbricos:

### **2.2.1 Red en estrella**

Es una topología de comunicación en la que una sola estación base va a enviar y/o recibir mensajes de varios nodos remotos, los cuales no pueden enviarse mensajes entre ellos.

La principal ventaja de este tipo de red para WSN incluye la simplicidad, la capacidad de mantener al mínimo el consumo de energía del nodo remoto. También permite comunicaciones de baja latencia entre el nodo remoto y la estación base.

La principal desventaja es que la estación base deberá estar dentro del rango de transmisión de todos los nodos individuales y no es considerada tan robusta como otras redes debido a su dependencia de un solo nodo para la administración de la red.



*Figura 1. Red de Estrella. Fuente: (Matin & Islam, 2012).*

### **2.2.2 Red de malla**

Esta red permite la transmisión de datos de nodo a nodo de la red que se encuentre dentro de su rango de transmisión. Permitiéndole que se conozca en cada uno de los múltiples saltos, es decir, si un nodo envía un mensaje hacia otro nodo fuera del alcance de las comunicaciones, usará un nodo intermedio para reenviar el mensaje al nodo deseado.

Esta topología de red tiene la principal ventaja de la redundancia y la escalabilidad. Si un nodo falla, un nodo remoto que aún puede comunicarse con otro nodo en su rango podrá reenviar el mensaje a la ubicación deseada. Por otro lado, el rango de la red no está limitado por el rango entre los nodos individuales; sino que se podrá ampliar agregando más nodos al sistema.

La principal desventaja de este tipo de red es que el consumo de energía de los nodos que usan múltiples saltos es mayor que para los nodos que no tienen esta capacidad, lo que a menudo limita la vida útil de la batería. Además, conforme aumenta los saltos de comunicación a un destino, también aumenta el tiempo para que se entregue el mensaje, teniendo en cuenta si se requiere un funcionamiento de baja potencia de los nodos.

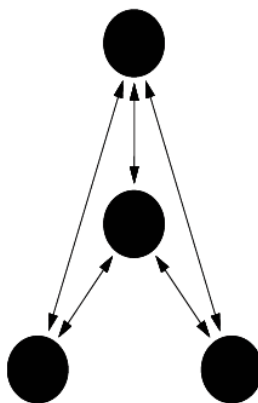


Figura 2. Red de Malla. Fuente: (Matin & Islam, 2012).

### 2.2.3 Híbrida Estrella - Red de malla

Un híbrido entre la red en estrella y en malla va a proporcionar una red de comunicaciones robusta y versátil, al tiempo que va a mantener la capacidad de consumo de energía al mínimo.

Los nodos de sensor con menor potencia no están habilitados con la capacidad de reenviar mensajes. Esto permite mantener un consumo de energía mínimo. Sin embargo, otros nodos de la red están habilitados con capacidad de múltiples saltos, lo que les permite reenviar mensajes desde los nodos de baja potencia a otros nodos de la red.

“Generalmente, los nodos con la capacidad de múltiples saltos son de mayor potencia y, si es posible, a menudo están conectados a la línea de suministro eléctrico. Esta es la topología implementada por el nuevo estándar de redes de malla conocido como ZigBee” (Matin & Islam, 2012).

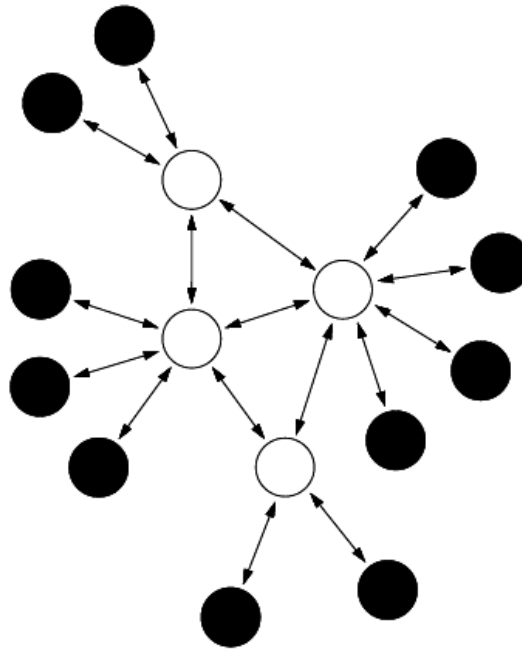


Figura 3. Red de Híbrida Estrella- Malla. Fuente: (Matin & Islam, 2012).

### 2.3 Estructura de comunicación de una red de sensores inalámbricos

Los nodos sensores generalmente están dispersos en un campo sensor. Cada uno de estos nodos sensores dispersos tiene la capacidad de recopilar datos y enrutarlos de regreso al nodo principal y a los usuarios finales. Los datos se enrutan de regreso al usuario final mediante una arquitectura sin infraestructura de múltiples saltos a través del nodo principal. El nodo principal puede comunicarse con el nodo del administrador de tareas a través de Internet o satélite.

Los protocolos usados por el receptor y los nodos de sensores se muestran a continuación en la Fig. 4, donde se combina la energía y conocimiento del enrutamiento, integra los datos con protocolos de red, comunica energía de manera eficiente a través del medio inalámbrico y promueve los esfuerzos cooperativos de los nodos sensores.

Consta de la capa de aplicación, la capa de transporte, la capa de red, la capa de enlace de datos, la capa física, en el plano de administración de energía, plano de administración de movilidad y el plano de administración de tareas.

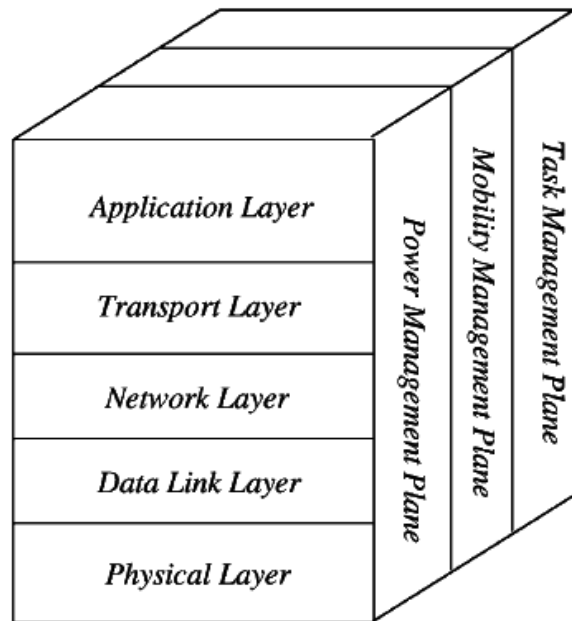


Figura 4. Stack de Protocolos de WSN. Fuente: (Matin & Islam, 2012).

La capa aplicación hace que el hardware y el software de la capa más baja sea transparente para el usuario final, la capa transporte ayuda a mantener el flujo de datos si se requiere; por otro lado, la capa de red se encargará de enrutar los datos suministrados por la capa de transporte, usando protocolos específicos entre los nodos. La capa enlace de datos es responsable de la multiplexación de los datos, detección de tramas, control de acceso a medios (MAC) y control de errores. La capa física aborda las necesidades de una modulación simple pero robusta, selección de frecuencia, encriptación de datos, transmisión y técnicas de recepción.

“Además, los planos de administración de energía, movilidad y administración de tareas van a monitorear la distribución de energía, movimiento y tareas entre cada uno de los nodos de sensores. Estos planos ayudan a los nodos sensores a coordinar la tarea de detección y reducir el consumo total de energía” (Seah, Eu, & Tan, 2009).

#### 2.4 Estándar IEEE 802.15.4

“Este estándar contiene las especificaciones de la capa física (PHY) y la subcapa de control de acceso al medio (MAC) usado para la conexión inalámbrica de velocidad de datos baja y con dispositivos fijos o móviles sin baterías o con requerimientos de consumo de batería limitada. Por otra parte, el estándar proporciona modos que permitirán precisión en donde los PHY son definidos



para los dispositivos que operan en varias regiones geográficas” (IEEE Computer Society, 2020).

Este estándar proporciona conexión inalámbrica ultra baja, con costos bajos, poco complejos y baja velocidad de datos entre dispositivos en especial aquellos de requisitos de comunicación en el Internet de las cosas. Además, algunos PHY alternativos brindarán capacidades de rango de precisión a un metro, admitiendo una variedad de bandas de frecuencia.

#### **2.4.1 Descripción General**

Una red de área personal inalámbrica de baja velocidad (LR-WPAN) es una red de comunicación simple y de bajo costo que permite la conectividad inalámbrica en aplicaciones con energía limitada y requisitos de rendimiento bajos. Los principales objetivos de un LR-WPAN son la facilidad de instalación, la transferencia de datos confiable, un costo extremadamente bajo y una duración razonable de la batería, mientras se mantiene un protocolo simple y flexible.

Este estándar define múltiples PHY que operan en una variedad de bandas de frecuencia.

Dos tipos de dispositivos diferentes pueden participar en una red IEEE 802.15.4: un dispositivo de función completa (FFD) y un dispositivo de función reducida (RFD). Un FFD es un dispositivo que puede servir como coordinador o coordinador (PAN). Un RFD es un dispositivo que no puede servir ni como coordinador PAN ni como coordinador. Un RFD está diseñado para aplicaciones que son extremadamente simples, como un interruptor de luz o un sensor de infrarrojos pasivo; no tiene la necesidad de enviar grandes cantidades de datos y solo se asocia con un solo FFD a la vez. En consecuencia, la RFD se puede implementar utilizando recursos y capacidad de memoria mínimos.

#### **2.4.2 Formato de la Trama**

El formato general de una estructura de datos se muestra en la Figura 5. Cada campo está representado por una columna en la figura que indica el tamaño del campo y el nombre y / o tipo del dato codificado en el campo. La convención de una estructura de datos es recursiva en el sentido de que un campo puede contener una estructura de datos que a su vez se compone de campos. Representa una estructura de datos con 3 campos de 2 octetos de longitud, 6 bits y 2 bits para un tamaño total de 3 octetos.

Octets: 2	Bits: 0–5	6–7	...
Field Name/Type (Number datum)	Field Name/Type (Bit string datum)	Field Name/Type (Bit string datum)	...

Figura 5. Formato de Estructura de Datos General. Fuente: (IEEE Computer Society, 2020).

### 2.4.3 Campos de Aplicación

- *Red de servicios inteligentes (SUN)* Los SUN permiten que múltiples aplicaciones operen sobre recursos de red compartidos, proporcionando monitoreo y control de un sistema de servicios públicos. Los dispositivos SUN están diseñados para funcionar en aplicaciones inalámbricas de baja potencia y a gran escala y, a menudo, requieren el uso de la potencia de transmisión máxima disponible según las regulaciones aplicables, a fin de proporcionar conexiones punto a punto de largo alcance.
- *Comunicaciones y control ferroviario (RCC)* RCC se refiere a un intercambio de información inalámbrico y comunicaciones de sensor o control implementadas en aplicaciones como las siguientes
- *Espacio en blanco de la televisión (TVWS)* La operación de TVWS tiene el requisito de determinar qué asignaciones de frecuencia de TVWS están disponibles para su uso en un momento y ubicación geográfica determinados.
- *Identificación por radiofrecuencia (RFID)* Los dispositivos RFID activos se utilizan para identificar y, a menudo, localizar personas u objetos en entornos industriales o comerciales.

### 2.4.4 Componentes de IEEE 802.15.4 WPAN

Un sistema conforme a esta norma consta de varios componentes. El más básico es el dispositivo. Un dispositivo tiene una única interfaz de radio que implementa un IEEE Std 802.15.4 MAC y PHY. Dos o más dispositivos que se comunican en el mismo canal físico constituyen una red de área personal inalámbrica (WPAN). Un WPAN incluye al menos un FFD, que opera como coordinador del PAN.

### 2.4.5 Gestión de PHY múltiple (MPM) del SUN WPAN

Varias SUN PHY diferentes pueden operar en la misma ubicación y dentro de la misma banda de frecuencia. Con el fin de mitigar la interferencia, se especifica un

esquema MPM para SUN para facilitar la coexistencia entre PHY. Para este propósito, el esquema MPM facilita la interoperabilidad y la negociación entre coordinadores potenciales con diferentes PHY al permitir que un coordinador potencial detecte una red operativa durante su fase de descubrimiento utilizando el modo de señalización común (CSM) apropiado para la banda que se está utilizando. El procedimiento MPM se puede utilizar junto con el mecanismo de evaluación de canal claro (CCA) para proporcionar coexistencia.

#### **2.4.6 Arquitectura**

La arquitectura IEEE 802.15.4 se define en términos de varios bloques para simplificar el estándar. Cada capa es responsable de una parte del estándar y ofrece servicios a las capas superiores. Las interfaces entre las capas sirven para definir los enlaces lógicos que se describen en este estándar.

Un dispositivo LR-WPAN contiene al menos un PHY, que contiene el transceptor de radiofrecuencia (RF) junto con su mecanismo de control de bajo nivel, y una subcapa MAC que proporciona acceso al canal físico para todos los tipos de transferencia. La Figura 6 muestra estos bloques en una representación gráfica.

Las siguientes capas superiores consisten en una capa de red, que proporciona configuración de red, manipulación y enrutamiento de mensajes, y una capa de aplicación, que proporciona la función prevista del dispositivo. La definición de estas capas está fuera del alcance de esta norma.

**2.4.6.1 PHY** Las características del PHY son la activación y desactivación del transceptor de radio, detección de energía (ED), indicación de calidad de enlace (LQI), selección de canal, CCA, rango y transmisión, así como recepción de paquetes a través del medio físico. La PHY de banda ultra ancha (UWB) de frecuencia de repetición de pulso de alta frecuencia (HRP) también tiene la característica de rango de precisión.

**2.4.6.2 Subcapa MAC** La subcapa MAC proporciona dos servicios: el servicio de datos MAC y el servicio de gestión MAC que se interconecta con el punto de acceso al servicio (SAP) de la entidad de gestión de la subcapa MAC (MLME) (conocido como MLME-SAP). El servicio de datos MAC permite la transmisión y recepción de unidades de datos de protocolo MAC (MPDU) a través del servicio de datos PHY.

Las características de la subcapa MAC son gestión de balizas, acceso al canal, gestión de intervalo de tiempo garantizado (GTS), validación de tramas, entrega de tramas reconocidas, asociación y disociación. Además, la subcapa MAC proporciona enlaces para implementar mecanismos de seguridad apropiados para la aplicación.

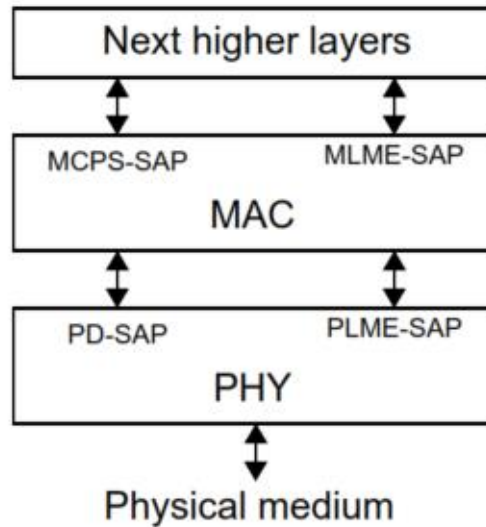


Figura 6. Arquitectura del dispositivo LR-WPAN. Fuente: (IEEE Computer Society, 2020).

## 2.5 RIOT-OS

RIOT OS tiene como objetivo cerrar la brecha que observamos entre el sistema operativo para WSN y el sistema operativo tradicional completo que se ejecuta actualmente en hosts de Internet. Se basa en objetivos de diseño que incluyen eficiencia energética, pequeña huella de memoria, modularidad y acceso API uniforme, independientemente del hardware subyacente.

RIOT implementa una arquitectura de micronúcleo heredada de FireKernel, por lo que admite múltiples subprocesos con API estándar. Además de las características originales de FireKernel, RIOT agrega soporte para C ++, lo que permite bibliotecas poderosas como el marco del algoritmo Wiselib, y proporciona una pila de red TCP / IP.

Las ventajas de la arquitectura RIOT incluyen:

- Alta confiabilidad
- Una API amigable para el desarrollador.

La estructura modular de microkernel de RIOT lo hace robusto contra errores en componentes individuales. Las fallas en el controlador del dispositivo o el sistema de archivos, por ejemplo, no dañarán todo el sistema. “RIOT permite a los desarrolladores crear tantos subprocesos como sea necesario y los sistemas distribuidos se pueden implementar fácilmente mediante el uso de la API de mensajes del kernel. La cantidad de subprocesos solo está limitada por la memoria disponible y el tamaño de la pila para cada subproceso, mientras que

la sobrecarga computacional y de memoria es mínima” (Baccelli, Hahm, Günes, Wählich, & Schmidt, 2013).

Para cumplir con los estrictos requisitos de tiempo real, RIOT impone períodos constantes para las tareas del kernel (por ejemplo, ejecución del programador, comunicación entre procesos, operaciones del temporizador). Un requisito previo importante para los tiempos de ejecución garantizados de O es el uso exclusivo de la asignación de memoria estática en el kernel. Sin embargo, se proporciona administración de memoria dinámica para las aplicaciones.

“Logramos un tiempo de ejecución constante del programador mediante el uso de una lista de subprocesos enlazados circular de tamaño fijo. El tiempo de ejecución constante de las operaciones del temporizador se obtiene aprovechando el hecho de que las MCU suelen proporcionar múltiples registros de comparación” (Trickey, 2020).

Es obligatorio maximizar la duración de los modos de sueño profundo, para implementar la eficiencia energética también para dispositivos IoT más potentes, por lo que RIOT presenta un programador que funciona sin eventos periódicos. Siempre que no haya tareas pendientes, RIOT cambiará al subproceso inactivo, que determina el modo de suspensión más profundo posible según los dispositivos periféricos en uso. Solo las interrupciones (externas o generadas por el kernel) despiertan el sistema desde el estado inactivo.

La baja complejidad de las funciones del kernel es un factor principal para la eficiencia energética de un sistema operativo. Por lo tanto, la duración y la ocurrencia del cambio de contexto deben minimizarse. En RIOT, el cambio de contexto se realiza en dos casos:

- Se llama a una operación del núcleo correspondiente, por ejemplo, un bloqueo mutex o la creación de un nuevo hilo
- Una interrupción provoca un cambio de hilo.

El primer caso ocurrirá raramente. Por ejemplo, cada hilo se crea normalmente una vez. Por lo tanto, es importante reducir el tiempo de procesamiento en caso de un cambio de hilo. “Por lo tanto, el kernel de RIOT proporciona un programador minimizado, cuando se llama fuera de una rutina de servicio de interrupción. En ese caso, no es necesario guardar el contexto del hilo anterior y, por lo tanto, se puede realizar un cambio de tarea en muy pocos ciclos de reloj” (Trickey, 2020).

## **2.6 Agricultura de Precisión**

La agricultura de precisión es un enfoque de gestión que se centra en la observación (casi en tiempo real), la medición y las respuestas a la variabilidad en cultivos, campos y animales. Puede ayudar a aumentar el rendimiento de los cultivos y el rendimiento de los animales, reducir los costos, incluidos los costos de mano de obra, y optimizar los insumos del proceso. “Todos estos pueden ayudar a aumentar la rentabilidad. Al mismo tiempo, la agricultura de precisión puede aumentar la seguridad laboral y reducir los impactos ambientales de la agricultura y las prácticas agrícolas, contribuyendo así a la sostenibilidad de la producción agrícola” (Wiangtong & Sirisuk, 2018).

El término agricultura de precisión significa una serie de estrategias y herramientas que permiten a los agricultores optimizar y aumentar la calidad y productividad del suelo poniendo en marcha una serie de intervenciones clave específicas, un resultado que se puede lograr gracias a la introducción de tecnologías cada vez más avanzadas.

“Se denomina “precisión” porque gracias a las herramientas de última generación utilizadas, es posible realizar la intervención adecuada, en el lugar adecuado, en el momento adecuado, respondiendo a las demandas específicas de los cultivos y áreas individuales de terreno con niveles superiores de precisión” (Dolci, 2017).

Las tecnologías se utilizan, en primer lugar, para recopilar los datos y la información necesarios para tomar decisiones sobre cómo impulsar la producción y, en segundo lugar, para poner en marcha las acciones correctivas necesarias para lograr este objetivo. “Hoy en día se habla cada vez más de Agricultura 4.0, que es la evolución del concepto de agricultura de precisión: este término se refiere a todas las herramientas y estrategias que utilizan tecnologías de punta de manera interconectada comenzando por el uso de datos para mejorar y optimizar la producción” (Ivanov, Bhargava, & Donnelly, 2015).

### **2.6.1 Las ventajas de la agricultura de precisión**

Producir más con menos recursos manteniendo estándares de calidad superiores: este es el objetivo final de la agricultura de precisión.

Este tipo de enfoque genera una doble ventaja:

- Para las empresas agrícolas, que pueden optimizar esfuerzos y recursos, reducir el consumo y el desperdicio e impulsar la productividad de la tierra. El trabajo también se vuelve más rentable para los agricultores y contratistas, ya que los procesos se gestionan de forma más rápida y eficaz, lo que lleva a una reducción de los costes por hora. Por último, pero no menos importante, aumentan el rendimiento al tiempo que reducen la fatiga de los recursos humanos.
- “Para el medio ambiente, dado que se reduce el desperdicio de fertilizantes y herbicidas, las emisiones y la compactación del suelo gracias a un uso más racional de los recursos” (Ivanov, Bhargava, & Donnelly, 2015).

## **2.6.2 Herramientas de recopilación de datos**

El seguimiento del estado sanitario de los cultivos requiere un gran esfuerzo, especialmente cuando se extienden por áreas muy extensas. La producción en los campos casi nunca es homogénea, algunas áreas son mucho más productivas que otras.

- El monitoreo de cultivos por satélite es una herramienta que permite a los agricultores monitorear constantemente la salud de sus campos gracias a un análisis de imágenes multiespectrales de imágenes de satélite de alta resolución y también activar rápidamente cualquier alarma.
- Una solución alternativa, que no implica el uso de satélites, son los drones agrícolas. Hoy en día se utilizan principalmente para cartografía terrestre, pero las versiones más avanzadas adoptan sensores infrarrojos y sistemas de imágenes para detectar problemas que no pueden ser detectados a simple vista, tal como lo hacen los satélites.
- “Otro instrumento importante para la agricultura de precisión, que no implica ninguna intervención desde arriba, son los sensores ambientales instalados en los campos, capaces de registrar datos climáticos e información sobre las necesidades hídricas del suelo” (Wiangtong & Sirisuk, 2018).

## **2.6.3 Análisis en agricultura de precisión**

Ya existen algunos sistemas de sensores agrícolas. En este artículo, describimos cómo los diseños y los niveles de inteligencia de estos sistemas varían según el área de aplicación.

### **2.6.3.1 Vigilancia**

Kerry Taylor y sus colegas describen un sistema WSN implementado en Kirby Farm cerca de Armidale, Nueva Gales del Sur.<sup>1</sup> La red incorpora varios sensores diferentes para la humedad del suelo, temperatura del aire, humedad y presión, lluvia y granizo. Los datos de monitoreo, proporcionados por los sensores, se envían a una entidad centralizada, donde los datos se enriquecen y analizan. Los resultados del análisis se comunican al personal de la granja a través de sus dispositivos móviles personales. De esta manera, el sistema presenta una solución ejemplar de apoyo a la toma de decisiones para el manejo de pastos.

Al mismo tiempo, C.J. Rutten y sus colegas han realizado una encuesta integral que muestra la falta de análisis e inteligencia en los sistemas de sensores para el manejo de la salud animal en una granja lechera. “La encuesta presenta una clasificación de cuatro niveles de las soluciones de sensores existentes. El primer nivel está destinado a sistemas que miden aspectos específicos de la salud animal. Las lecturas obtenidas por dichos sistemas (por ejemplo, la composición de la leche) se presentan tal cual, al personal de la granja, quien saca sus propias conclusiones” (Ivanov, Bhargava, & Donnelly, 2015).

Los sistemas de segundo nivel interpretan las lecturas. Por ejemplo, la conductividad eléctrica de la leche puede usarse para identificar mastitis (inflamación del tejido de la ubre y la glándula mamaria, que generalmente ocurre debido a una infección bacteriana), y la aceleración 3D se puede usar para detectar problemas de locomoción. Los sistemas de tercer nivel integran información procedente de diversas fuentes, incluidos otros sistemas de sensores y datos que no son de sensores. Un sistema de este tipo ayudaría al personal agrícola a comprender cantidades potencialmente grandes de datos, lo que podría resultar abrumador. Rutten y sus colegas han identificado una seria falta de tales sistemas. El cuarto nivel es para los sistemas de apoyo a la toma de decisiones.

“Aunque algunos de los sistemas de sensores se han identificado como de nivel cuatro, su funcionalidad se basa en alertar sobre eventos de segundo nivel, como mastitis o estro (un período de fertilidad y receptividad sexual en las hembras de mamíferos)” (Ivanov, Bhargava, & Donnelly, 2015).

### **2.6.3.2 Automatización**

El contexto y la conciencia del usuario son tendencias comunes en los sistemas modernos de automatización de procesos agrícolas. Por ejemplo, Fernando Auat



Cheein y Ricardo Carelli encuestaron sistemas robóticos no tripulados propuestos para aplicaciones agrícolas. “Este sistema típicamente intenta reemplazar el trabajo manual en tareas, como monitorear el entorno de la granja, mapear los resultados del monitoreo (como el mapeo de rendimiento) y tomar acciones particulares (por ejemplo, raleo de flores). El sistema puede incluir varios sensores diferentes (por ejemplo, para polen o nivel de CO<sub>2</sub>) y opera” (Ivanov, Bhargava, & Donnelly, 2015).

## **2.7 Cultivo de Arándanos**

Los arándanos gruesos y jugosos no solo son deliciosos. Tienen un alto contenido de nutrientes, antioxidantes, fibra y vitaminas. Afortunadamente, con muy poco esfuerzo, los arándanos son fáciles de cultivar, si tiene las condiciones de suelo adecuadas. Antes de la década de 1900, la única forma de disfrutar de estos nativos norteamericanos era encontrarlos en la naturaleza. Luego, los científicos comenzaron a descubrir los secretos del cultivo de arándanos.

“Un pariente del rododendro y la azalea, los arbustos de arándanos no son solo una gran planta fructífera, sino también una adición atractiva a su paisaje en general, ya que ofrecen follaje otoñal escarlata y flores primaverales de color blanco cremoso en forma de campana. Además, los arándanos son uno de los superalimentos de la naturaleza, repletos de nutrientes esenciales, minerales y polifenoles que mejoran la salud” (Xie & Wu, 2009).

### **2.7.1 ¿Cuándo plantar arándanos?**

- Los arándanos se pueden plantar en primavera o también a fines del otoño en todas las regiones excepto en las más frías. En las Zonas 5 e inferiores, es mejor esperar hasta principios o mediados de la primavera para plantar.
- “Si están disponibles, las plantas de 1 a 3 años son una buena opción. Estos se pueden comprar en contenedores o desnudos. En cualquier caso, asegúrese de comprar en un vivero o sitio web de buena reputación” (Ochmian, Błaszak, Lachowicz, & Piwowarczyk, 2020).

### **2.7.2 Elegir y preparar un lugar de plantación**

- Seleccione un lugar soleado y protegido. Si bien los arándanos son tolerantes a la sombra, se obtienen mejores cosechas al sol. Al mismo tiempo, no deben exponerse a vientos fuertes y secos.

- No plante los arándanos demasiado cerca de los árboles, ya que los árboles no solo bloquearán la luz solar, sino que también absorberán la humedad del suelo.
- Si planta varios arbustos, es mejor plantarlos en un parche, en lugar de esparcirlos por todo el jardín. Esto reforzará la producción y la calidad de las berries.
- El arándano es una planta de raíces poco profundas. Por lo tanto, requiere un suelo que retenga la humedad, pero que también drene bien y no permanezca mojado. No plante arándanos en sitios que tengan suelos arcillosos pesados que permanezcan húmedos.
- Los arándanos prosperan en suelos ácidos. El pH del suelo idealmente debería estar entre 4.0 y 5.0; el suelo que no es lo suficientemente ácido atrofiará el crecimiento. “El suelo se puede acidificar mezclando una pequeña cantidad de azufre granulado en el suelo varios meses antes de plantar. La turba, así como la corteza de pino o las agujas, también son buenas adiciones que ayudarán a acidificar el suelo” (Ochmian, Grajkowski, Mikiciuk, Ostrowska, & Chelpinski, 2009).
- Mezcle materia orgánica en el suelo antes de colocar sus arbustos de arándanos.

### **2.7.3 ¿Cómo Plantar Arándanos?**

- Consejo: tenga cuidado de no plantarlos demasiado profundamente. El cepellón debe estar justo debajo de la superficie (de un cuarto a media pulgada).
- Cava hoyos de aproximadamente 20 pulgadas de profundidad y 18 pulgadas de ancho (o aproximadamente el doble de ancho y el doble de profundo que las raíces de la planta).
- Separe los arbustos de 4 a 5 pies de distancia en una fila, con al menos 8 pies entre filas. Prepare una mezcla de siembra de 2 partes de marga y una parte de moho de hoja de roble, turba, aserrín envejecido o compost, y coloque una capa de esta mezcla en el fondo del hoyo.
- Coloque el arbusto en el agujero con el cepellón justo debajo de la superficie y sus raíces se extienden. Rellena el agujero con tierra.
- “Aplique fertilizante un mes después de plantar, no al momento de plantar. Aplique ½ onza de fertilizante 10-10-10 en una banda alrededor de la planta de 6 a 12 pulgadas de la corona” (Fang, Nunez, da Silva, Phillips, & Munoz, 2020).

## 2.8 Cloud Computing

La computación en la nube se refiere a cualquier tipo de servicio alojado entregado a través de Internet. Estos servicios a menudo incluyen servidores, bases de datos, software, redes, análisis y otras funciones informáticas que se pueden operar a través de la nube.

Los usuarios del servicio pueden acceder a los archivos y programas almacenados en la nube desde cualquier lugar, lo que elimina la necesidad de estar siempre cerca del hardware físico. En el pasado, por ejemplo, los documentos y hojas de cálculo creados por el usuario tenían que guardarse en un disco duro físico, unidad USB o disco. Sin algún tipo de componente de hardware, los archivos eran completamente inaccesibles fuera de la computadora en la que se originaron. Gracias a la nube, pocas personas ya se preocupan por los discos duros dañados o las unidades USB perdidas o dañadas. La computación en la nube hace que los documentos estén disponibles en todas partes porque los datos en realidad viven en una red de servidores alojados que transmiten datos a través de Internet.

### 2.8.1 Brokers de MQTT y guía de alojamiento en la nube

Cuando se trata de elegir un bróker MQTT, tiene tres opciones principales:

- Utilice su propio broker / servidor instalado localmente
- Utilice un servidor virtual o un servidor basado en la nube
- Útilice una aplicación de servidor compartido

### 2.8.2 Servidor propio instalado localmente

Puede instalar un bróker MQTT en su propio hardware de servidor, incluida una Raspberry Pi. Hay muchos corredores / servidores para elegir, y la mayoría son gratuitos y de código abierto. Cada uno de estos corredores tendrá sus propios requisitos, por ejemplo, Aedes (formalmente Mosca) está basado en Node.js y, por lo tanto, requiere la instalación de node.js.

La siguiente tabla enumera los corredores más populares

Tabla 1. Brokers más populares.

Broker	Descripción
Mosquitto	- Broker ligero de código abierto escrito en C. Probablemente el broker MQTT más popular. Admite MQTT v3.1.1 y v5. Es

	probable que este sea el intermediario predeterminado para las redes de borde.
<b>Mosca- Now Aedes</b>	<ul style="list-style-type: none"> <li>- Mosca ahora Aedes se basa en Node.js y, por lo tanto, requiere la instalación de node.js. También se puede instalar como nodo en node-red.</li> <li>- Aedes reemplaza a Mosca y Mosca ya no se está desarrollando.</li> <li>- No tiene muchas funciones en comparación con mosquitto, pero es ideal para proyectos de automatización del hogar.</li> <li>- Actualmente es compatible con v 3.1.1</li> <li>- Se ejecuta en Windows y Linux</li> </ul>
<b>emqtt</b>	<ul style="list-style-type: none"> <li>- Escrito en Erlang es de código abierto y se describe como enormemente escalable.</li> <li>- EMQ Implementa las especificaciones de protocolo MQTT V3.1 y V3.1.1 y es compatible con MQTT-SN, CoAP, WebSocket, STOMP.</li> </ul>

*Fuente: (Steve, 2021)*

### 2.8.3 Instalar Broker en un servidor virtual

Están disponibles en muchos proveedores y actualmente se utilizan principalmente para el alojamiento de sitios web. Sin embargo, se pueden utilizar para alojar cualquier aplicación web y serán una opción popular para alojar Node.js y MQTT en la nube.

Con este tipo de alojamiento, usted no es responsable del hardware, pero la instalación y administración del software es su responsabilidad. Es probable que se utilicen servidores / corredores MQTT en línea o basados en la nube para conectar diferentes ubicaciones geográficas físicas. Google, Amazon, Microsoft, IBM y muchos otros ofrecen alojamiento basado en la nube.

## Capítulo 3

### Diseño de Hardware y Software

En este capítulo se detallará el proceso metodológico que se usó para el diseño y desarrollo de la propuesta de investigación. Inicialmente se presenta los requerimientos de Usuario o Stakeholders, luego se realiza un estudio de Benchmark para la elección adecuada de los equipos a usarse. A continuación, en la elaboración del prototipo se define el software que incluye la programación del sistema y finalmente se verificará cada fase usando como referencia el modelo de cascada.

#### 3.1 Metodología

Este modelo es muy básico para el desarrollo de software. Es muy simple pero idealista. El modelo de cascada clásico divide el ciclo de vida en un conjunto de fases. Este modelo considera que una fase puede iniciarse después de la finalización de la fase anterior. Esa es la salida de una fase que será la entrada a la siguiente fase. Por lo tanto, el proceso de desarrollo puede considerarse como un flujo secuencial en la cascada. Aquí las fases no se superponen entre sí. Las diferentes fases secuenciales del modelo de cascada clásico se muestran en la siguiente figura:

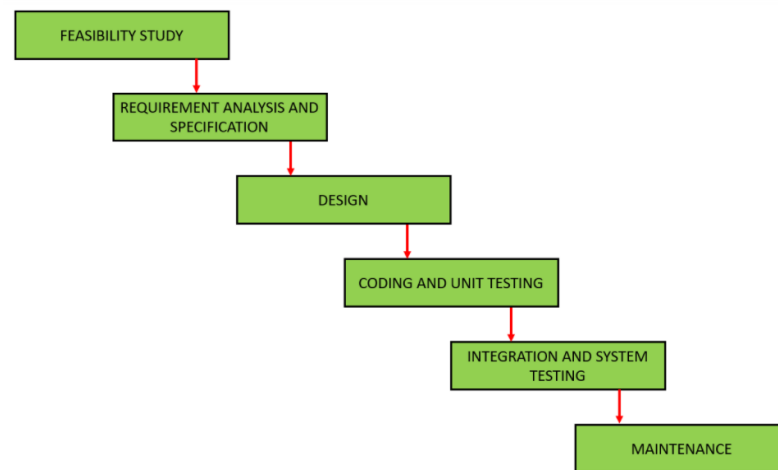


Figura 7. Modelo en Cascada. Fuente: (Adenowo & Adenowo, 2013)

Las fases se describen a continuación:

##### 3.1.1 Estudio de viabilidad (Feasibility Study)

El objetivo principal de esta fase es determinar si sería viable desde el punto de vista financiero y técnico desarrollar el software. El estudio de viabilidad implica comprender el problema y luego determinar las diversas estrategias posibles para resolver el problema. “Estas diferentes soluciones identificadas se analizan en función

de sus beneficios e inconvenientes. Se elige la mejor solución y todas las demás fases se llevan a cabo según esta estrategia de solución” (Adenowo & Adenowo, 2013) .

### **3.1.2 Análisis y especificación de requisitos (Requirements analysis and specification)**

El objetivo de la fase de análisis y especificación de requisitos es comprender los requisitos exactos del cliente y documentarlos adecuadamente. Esta fase consta de dos actividades diferentes: Recopilación y análisis de requisitos, Especificación de requerimiento.

### **3.1.3 Diseño (Design)**

El objetivo de la fase de diseño es transformar los requisitos especificados en el documento SRS en una estructura que sea adecuada para su implementación en algún lenguaje de programación.

### **3.1.4 Codificación y pruebas unitarias (Coding and Unit testing)**

En la fase de codificación, el diseño del software se traduce al código fuente utilizando cualquier lenguaje de programación adecuado. “Por tanto, cada módulo diseñado está codificado. El objetivo de la fase de prueba unitaria es comprobar si cada módulo funciona correctamente o no” (Adenowo & Adenowo, 2013).

### **3.1.5 Integración y prueba del sistema (Integration and System testing)**

La integración de los diferentes módulos se lleva a cabo poco después de que se hayan codificado y probado en la unidad. La integración de varios módulos se lleva a cabo de forma incremental en una serie de pasos.

“Durante cada paso de integración, los módulos previamente planificados se agregan al sistema parcialmente integrado y el sistema resultante se prueba. Finalmente, después de que todos los módulos se hayan integrado y probado con éxito, se obtiene el sistema de trabajo completo y se lleva a cabo la prueba del sistema” (Adenowo & Adenowo, 2013).

### **3.1.6 Mantenimiento (Maintenance)**

El mantenimiento es la fase más importante del ciclo de vida de un software. El esfuerzo dedicado al mantenimiento es el 60% del esfuerzo total dedicado a desarrollar un software completo. Básicamente, existen tres tipos de mantenimiento: Mantenimiento correctivo, Mantenimiento perfecto, Mantenimiento adaptativo.

## **3.2 Análisis**

Para el Análisis se tomará en cuenta el estado actual del vivero, la ubicación y las características necesarias para la producción.

### 3.2.1 Situación Actual del Invernadero

El Invernadero de los Arándanos se encuentra ubicado en el Barrio San Pedro de la Esperanza en la Calle Galo Plaza Lasso. Para la demostración del proyecto, se tomará en cuenta una pequeña extensión ubicada dentro del Invernadero en el cual están ubicadas 1700 plantas de Arándanos y 500 de ellas se encuentran actualmente en producción, tiene una extensión de aprox.  $50 m^2$  dedicados solamente al cultivo de Arándanos, se puede observar en la figura 8.



Figura 8. Extensión del Invernadero, visto desde Google Earth.

Dentro del cultivo no existe ningún método de medición automático ni de monitoreo de parámetros para tomar decisiones correctas con respecto a la producción de los Arándanos, usa un sistema de riego por goteo con el fin de suministrar agua a las plantas, así mismo cuenta con un sistema de cámaras de vigilancia para proteger la extensión de terreno.

### 3.3 Métodos y Técnicas de Investigación

Para completar la investigación hará falta tener una metodología, así como técnicas que nos ayudarán a descubrir todos los puntos necesarios y los puntos críticos que formarán parte del prototipo.

### **3.3.1 Método de la Investigación**

A continuación, se definen los métodos de investigación usados para el desarrollo del prototipo

**3.3.1.1 Método Inductivo deductivo.** En este método se estudió los beneficios de la implementación de los dispositivos tecnológicos que van a censar detenidamente los cultivos obteniendo los datos de la maceta.

**3.3.1.2 Método Sistemático.** Este método define todos los pasos a seguir para la implementación de un dispositivo electrónico de una manera lógica y sistemática determinando así el beneficio que puede otorgar.

**3.3.1.3 Método Descriptivo.** Este método va a implicar la recopilación de información y datos mediante una verificación bibliográfica y una presentación ordenada de dichos datos recolectados y así poder tener una idea de la situación actual en la que se encuentra, detallando además todos los requerimientos a ser abordados en el diseño

### **3.3.2 Técnica de la Investigación**

A continuación, se definen las técnicas de recopilación de datos para la veracidad y definición del Diseño del Proyecto:

#### **3.3.2.1 Observación Directa.**

Se realiza una observación directa en el Sembrío evidenciando que existen pérdida de productos debido a la irregularidad del clima conociendo además las posibles causas que afecta a la calidad de los arándanos y su producción.

#### **3.3.2.2 Entrevista**

Las entrevistas generalmente se llevan a cabo en persona, es decir, cara a cara, pero también se pueden administrar por teléfono o utilizando tecnología informática más avanzada, como Skype, Zoom, Teams. A veces se llevan a cabo en la casa del entrevistado, a veces en un lugar más neutral. Es importante que los entrevistados decidan si se sienten cómodos al invitar al investigador a su hogar y si tienen una habitación o un área donde puedan hablar libremente sin molestar a otros miembros del hogar.

Esta entrevista es realizada al dueño del Terreno donde se encuentran los cultivos de Arándanos y en el cual se aplica el prototipo, gracias a la información presentada se puede conocer la situación actual de los cultivos y los principales problemas que se



presentan en ellos, además se conoce el interés por parte del dueño recalcando que sería algo muy útil e innovador que facilitaría y beneficiaría a la producción de los Arándanos.

### **3.3.3 Resultados de la Entrevista**

A continuación, se va a detallar las respuestas generalizadas de lo que fue la entrevista hacia el encargado del Cultivo, datos necesarios para determinar la situación actual del cultivo y como se encuentra estructurado.

#### ***3.3.3.1 Datos Informativos.***

Se realizó una entrevista dirigida hacia uno de los encargados principales del cuidado de los cultivos su nombre es Víctor Rosero el cual se encuentra trabajando en el terreno ubicado en el Barrio San Pedro de la Esperanza calle Galo Plaza Lasso.

#### ***3.3.3.2 Datos Actuales de los Cultivos.***

Las Dimensiones del vivero son de alrededor 200 metros cuadrados de los cuales están ocupados únicamente 50 metros cuadrados por aproximadamente 1700 plantas de Arándanos, del total de plantas 500 de ellas producen arándanos por un total de 15 Kilogramos a la semana. Sin embargo, se podrían producir muchos más ya que tienen pérdidas en la producción y su principal motivo es por el contagio de plagas y la mala utilización de fertilizantes, así como el cambio del clima y la variación de las características necesarias para una buena producción.

#### ***3.3.3.3 Datos Técnicos de los Cultivos.***

Este último problema tiene que ver con varias características que deben ser monitoreadas constantemente estos datos deben ser el PH, la Humedad del Suelo y la temperatura del vivero. Para el primero de ellos; el PH, los rangos apropiados para la producción son de 4,5 a 5,5, para la humedad del suelo el rango va entre 80% al 100%; por otro lado, para la temperatura dentro del vivero varia en la noche alrededor de 10° y durante el día alrededor de 30°.

#### ***3.3.3.4 Tecnología dentro del Cultivo***

Actualmente para el monitoreo de datos se usan medidores manuales para el PH, conductividad y temperatura que toman los datos del agua de riego para el primer y

segundo caso; por otro lado, para la temperatura se usa el medidor dentro del cultivo y posteriormente se ingresan los datos en una hoja de Excel en Google Drive. También cuentan con Acceso a Internet dentro del cultivo el cual se encuentra conectado al sistema de cámaras de vigilancia, para el Invernadero se encuentran destinados 2 equipos adicionales con acceso a internet una Tablet y un Smartphone. Dado el sistema actual a los propietarios les gustaría implementar un Sistema que les ayude a monitorearlo de forma remota y con acceso a través de una página Web.

### ***3.4 Descripción General del Sistema***

El proyecto reúne 3 bloques indispensables para el funcionamiento del sistema inteligente, que permitirá a los cultivos producir Arándanos de buena calidad, implementando sensores que permitan detectar las condiciones del cultivo, un centro de administración donde los encargados podrán analizar las situaciones que se presenten, un sistema de Alerta y Control que se encargue de notificar a los granjeros para que puedan tomar acciones sobre los cultivos. Se detallan más a fondo a continuación:

#### *- Red de Sensores*

Donde se adquieren los datos, se envía a un nodo receptor especial, se recopilan y se envía al Gateway, se tendrán dos nodos de procesamiento que constarán de un microcontrolador OpenMote-B el cual es compatible con tecnología 802.15.4g. Y a este se conectarán los sensores de Temperatura, Ph, Y Humedad, estos serán ubicados dentro de la plantación de Arándanos donde se recolectarán los datos.

#### *- Centro de Administración*

Donde se recopilan los datos y se analizan; al igual que los sensores, pueden obtener energía almacenada en una batería. Este Centro de Administración consta de un nodo Central representado por OpenMote-B Gateway el cual recibe los datos los envía hacia un Base de Datos Ubicada en la Nube donde se procesarán o computarán estos datos.

#### *- Alerta y Control*

Una vez analizados los datos se procede a generar una alerta al agricultor o aplicar acciones según sea el caso. Este proceso se lo realiza en la plataforma AWS IoT en donde se procesan los datos y se envía al Servidor Web el cual será montado en la plataforma AWS IAM Service Web perteneciente a la compañía de Amazon generando así las alertas para el control del cultivo.

### 3.5 Determinación de los Stakeholders

Los principales Stakeholders involucrados en el desarrollo electrónico, son aquellos que tienen interés en el diseño e implementación de este proyecto. Se definen los requerimientos específicos de los Stakeholders para poder realizar el diseño e implementación con el objetivo de beneficiar a la producción del Arándano y contribuir al monitoreo del cultivo. En la Tabla 2 se detallan los Stakeholders.

Tabla 2. Lista de Stakeholders.

REQUERIMIENTOS DE STAKEHOLDERS	
Universidad Técnica del Norte	Entidad de respaldo
Msc. Edgar Maya	Docente Tutor
Msc. Fabián Cuzme	Asesor 1
Srta. Johanna Yépez	Desarrolladora del Proyecto
Daniel Rosero	Encargado del vivero

Fuente: Criterios de diseño del proyecto.

### 3.6 Requerimientos para el diseño

Se detallarán los requerimientos del diseño siendo los siguientes: Requerimientos de Stakeholders, requerimientos del funcionamiento del sistema y Requerimientos de Arquitectura. Se seleccionarán para que el dispositivo cumpla con el objetivo planteado así poder satisfacer las necesidades de los usuarios ya que ayudarán a tomar los datos de sensores como temperatura, humedad y PH en los cultivos.

#### 3.6.1 Nomenclatura de requerimientos

Se definirán los requerimientos de forma abreviada para una mejor comprensión y síntesis y de esa forma poder entender de manera adecuada lo que se realizará, véase tabla 3.

Tabla 3. Nomenclatura de requerimientos.

NOMENCLATURA	
Requerimiento	Nomenclatura
Stakeholders	RSSt

Funcionamiento del Sistema

RSFs

Arquitectura

RARa

*Fuente:* Criterios de diseño del proyecto.

### 3.6.2 Stakeholders requeridos

A continuación, se plantean los Stakeholders para que el sistema cumpla con la necesidad del usuario involucrado y se detallará en la tabla 4, para ser evaluado una vez se presente la implementación.

*Tabla 4. Requerimientos de Stakeholders.*

REQUERIMIENTOS DE STAKEHOLDERS (RSSt)			
N°	Requerimientos	Prioridad	
		Alta	Media
	Requerimientos del Sistema		
RSSt 1	Sistema de bajo consume de energía	X	
RSSt 2	Deberá implementarse en el Invernadero de Arándanos	X	
RSSt 3	Adquirir datos para ser almacenados	X	
RSSt 4	Garantizar la conexión con la Nube	X	
	Requerimientos de Usuario		
RSSt 5	Sistema pequeño, que invada poco al cultivo		X
RSSt 6	Sistema Fácil de Usar		X
RSSt 7	Inmune a la Humedad	X	
RSSt 8	Mostrar los datos en una Interfaz Amigable	X	

*Fuente:* Criterios de diseño del proyecto.

### 3.6.3 Requerimientos del funcionamiento del sistema

En los requisitos de funcionamiento se enumera las funciones específicas de este sistema, definiendo el comportamiento del mismo, la forma de uso, las conexiones

inalámbricas e internas y finalmente los requerimientos físicos y se pueden observar en la tabla 5.

Tabla 5. *Requerimientos de Funcionamiento.*

REQUERIMIENTOS DE FUNCIONAMIENTO (RSFs)			
N°	Requerimientos	Prioridad	
	Requerimientos de Uso	Alta	Media
RSFs 1	Equipos Pequeños que no interfieran en el crecimiento del cultivo	X	
RSFs 2	Dispositivos alimentados por baterías	X	
RSFs 3	Equipo completamente sellado para evitar daños por humedad		X
RSFs 4	Interfaz de Usuario clara. Ordenada y comprensible.		X
Requerimientos de Conexión			
RSFs 5	Conexión Inalámbrica usando tecnología 802.15.4g	X	
RSFs 6	Envío y Recepción de datos entre los nodos en todo momento	X	
RSFs 7	Adaptable a la disposición del Usuario		X
RSFs 8	Conexión con la Base de Datos en la Nube	X	

*Fuente:* Criterios de diseño del proyecto.

### 3.6.4 Requerimientos de Arquitectura

Se detallan los requerimientos y los componentes necesarios ya sea en software como en hardware, el análisis se presenta en la tabla 6.

Tabla 6. *Requerimientos de Arquitectura.*

REQUERIMIENTOS DE ARQUITECTURA (RARa)		
N°	Requerimientos	Prioridad

Requerimiento Lógico		Alta	Media
RARa 1	Entradas y Salidas Analógicas y Digitales	X	
RARa 2	Comunicación y Conexión en todo el sistema	X	
RARa 3	Calibración de Sensores	X	
Requerimientos de Conexión			
RSFs 5	Conexión Inalámbrica usando tecnología 802.15.4g	X	
RSFs 6	Envío y Recepción de datos entre los nodos en todo momento	X	
RSFs 7	Adaptable a la disposición del Usuario		X
RSFs 8	Conexión con la Base de Datos en la Nube	X	

*Fuente:* Criterios de diseño del proyecto.

### 3.7 Hardware y Software del sistema

Se hará la comparación de las opciones que cumplen los requisitos para poder desarrollar el prototipo que cumpla con cada uno de los requisitos planteados anteriormente y escoger así la mejor opción que logre sobrellevar los requisitos prioritarios.

#### 3.7.1 Elección de Hardware

Para cada una de las etapas se requerirá un nodo de recepción de datos el cual deberá recolectarlos y posteriormente enviarlos al borde router que se encargará de enviarlos a la Nube para ser procesados.

##### 3.7.1.1 Nodos Sensores

Se realizará el análisis para las diferentes placas que pueden cumplir esta función y posteriormente se hará una comparación para escoger el mejor dispositivo que se adapte a los requisitos planteados.

##### - *Texas Instruments CC1352P*

El dispositivo SimpleLink™CC1352P es un microcontrolador inalámbrico (MCU) sub-1 GHz y 2,4 GHz multiprotocolo y multibanda compatible con Thread, Zigbee,

Bluetooth 5.2 Low Energy, IEEE 802.15.4g, objetos inteligentes habilitados para IPv6 (6LoWPAN), MIOTY, Wi-SUN, sistemas propietarios, incluido TI 15.4-Stack (Sub-1 GHz y 2,4 GHz), y multiprotocolo concurrente a través de un controlador Dynamic Multiprotocol Manager (DMM).

Tabla 7. Características Texas Instruments CC1352P.

<b>Micro-Controller (Texas Instruments, CC1352P)</b>	
Protocolos	IEEE 802.15.4, Wireless M-Bus (T, S, C, N mode), 6LoWPAN, Wi-SUN NWP, Proprietary 2.4 GHz, Amazon Sidewalk, Bluetooth 5.2 Low Energy, Zigbee, Thread, MIOTY
RAM (KB)	80
Tipo	Wireless MCU
Características	Malla Bluetooth, enrutador de borde, radiogoniometría, LE 1M PHY, LE 2M PHY, LE codificado PHY (largo alcance), NCP, OAD, dispositivo final Sleepy, coordinador Zigbee
GPIO	26
Periféricos	2 SPI, 2 UART, 2 comparadores, 4 temporizadores, DAC de 8 bits, I2S, controlador de sensor
Seguridad	Identidad del dispositivo, seguridad de depuración, arranque seguro, actualización de firmware segura, aceleración criptográfica (RNG, AES, SHA, ECC)
Sensibilidad (dBm)	-121
Rango de temperatura de funcionamiento (C)	-40 a 85

Fuente: Datasheet de la placa. (INDUSTRIAL SHIELDS, 2019)

#### - **Texas Instruments CC1352R**

El dispositivo SimpleLink™CC1352R es un microcontrolador inalámbrico (MCU) sub-1 GHz y 2,4 GHz multiprotocolo y multibanda. Las características destacadas de este dispositivo incluyen:

- Dispositivo multibanda compatible con multiprotocolo simultáneo para Sub-1 GHz y 2,4 GHz a través de un controlador DMM.
- Amplia flexibilidad de compatibilidad con la pila de protocolos en el kit de desarrollo de software (SDK) SimpleLink™ CC13x2 y CC26x2.
- Potencia máxima de transmisión de +14 dBm a Sub-1 GHz con 24,9 mA y +5 dBm a 2,4 GHz con un consumo de corriente de 9,6 mA.

- Aplicaciones inalámbricas de mayor duración de la batería con baja corriente de espera de  $0,85 \mu\text{A}$  y retención completa de RAM.

Tabla 8. Características Texas Instruments CC1352P.

<b>Micro-Controller (Texas Instruments, CC1352R)</b>	
Microcontroller	Powerful 48-MHz Arm® Cortex®-M4F processor, EEMBC CoreMark, 352KB of in-system programmable flash, 256KB of ROM for protocols and library functions
RAM (KB)	80
Tipo	Wireless MCU
Características	TI-RTOS, controladores, gestor de arranque, controlador de baja energía Bluetooth® 5.2 e IEEE 802.15.4 MAC en ROM para optimizar el tamaño de la aplicación
Periféricos	Los periféricos digitales se pueden enrutar a cualquier GPIO, temporizadores de uso general de $4 \times 32$ bits u $8 \times 16$ bits, ADC de 12 bits, 200 k muestras/s, 8 canales
Sección de radio	Transceptor RF multibanda sub-1 GHz y 2,4 GHz compatible con Bluetooth 5.2 Low Energy y especificaciones LE anteriores, y IEEE 802.15.4 PHY y MAC, mecanismos de coexistencia PTA de 3 hilos, 2 hilos y 1 hilo
Sensibilidad (dBm)	-121
Rango de temperatura de funcionamiento (C)	-40 a 105

Fuente: Datasheet de la placa. (INDUSTRIAL SHIELDS, 2019)

#### - **Placa Open Mote - B**

El OpenMote B es la plataforma de creación de prototipos y desarrollo de hardware definitiva para el Industrial Internet of Things (IoT), específicamente para investigadores y desarrolladores que trabajan hacia redes inalámbricas de área de campo de baja potencia y largo alcance de última generación basadas en la pila IPv6.

Está construido alrededor del bien soportado Texas Instruments CC2538 ARMCortex-M3 microcontrolador, y cuenta con operación multibanda simultánea en el 2.4 GHz y bandas ISM de 868/915 MHz con soporte completo para el último IEEE 802.15.4 estándares, incluidas las modulaciones MR-OFDM de IEEE 802.15.4g-2012 (INDUSTRIAL SHIELDS, 2019).

Tabla 9. Características Técnicas del Open Mote - B

<b>Micro-Controller (Texas Instruments, CC2538)</b>	
---	--



ARM Cortex-M3 with code pre-fetch	Funcionamiento a 16 MHz o 32 MHz 32 Kbytes RAM 512 Kbytes FLASH
Periféricos en chip	4x propósito general, 1x temporizador de suspensión 1x ADC de 12 bits con 8 canales 2x SPI, 2x UART, 1x I2C
Aceleración de hardware de seguridad	Cifrado AES-128/256/SHA2 Intercambio seguro de claves ECC-128/256
Operación de baja potencia	Modo activo: 7/13mA (16/32 MHz) LPM1: 600uA (retención completa, activación 4us) LPM2: 1,3 uA (retención de RAM de 16 Kbyte, activación de 128 us, activación desde RTC) LPM3: 0,4 uA (retención de RAM de 16 Kbyte, activación de 128us, activación desde GPIO)
<b>Transceiver 1 (Texas Instruments, CC2538)</b>	
Opera en la banda ISM de 2,4 GHz con soporte para IEEE 802.15.4-2006	Modulación: OQPSK con DSSS Velocidad de datos: 250 kbps Sensibilidad del receptor: -97 dBm Potencia de transmisión: 7 dBm Corriente de transmisión: 24 mA a 0 dBm Recibir corriente: 20 mA
<b>Transceiver 2 (ATMEL, AT86RF215)</b>	
Opera en las bandas ISM de 868/915 MHz y 2,4 GHz con soporte para IEEE 802.15.4g-2012	Modulación: MR-FSK/OFDM/O-QPSK Velocidad de datos: 6,25 kbps a 2400 kbps Sensibilidad del receptor: -123 dBm Potencia de transmisión: 14.5dBm Corriente de transmisión: 62 mA a 14 dBm Corriente de recepción: 28 mA

Fuente: Datasheet de la placa. (INDUSTRIAL SHIELDS, 2019)

Una vez se presentaron cada uno de los posibles candidatos para la realización del prototipo se procede a escoger la mejor opción de acuerdo con los requisitos necesarios para la aplicación dentro del Invernadero.

Tabla 10. Selección de Nodo de Sensores.

Selección de Nodo de Sensores				
Hardware	RSSt	RSFs	RARa	Total
<i>CC1352P</i>	0	0	1	1
<i>CC1352R</i>	1	1	0	2
<i>Open Mote - B</i>	1	1	1	3

---

**Selección: Open Mote B – Cumple con todos los requisitos del prototipo.**

---

### 3.7.1.2 Recolector de Datos

Se requerirá un microcontrolador que permita obtener y recolectar todos los datos dentro de un archivo JSON que nos permitirá posteriormente enviarlo al Nodo de Sensores, el cual realizará el enrutamiento al Router Borde.

- **Arduino UNO**

Arduino UNO es una placa de microcontrolador basada en el ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden usarse como salidas PWM), 6 entradas analógicas, un resonador de cerámica de 16 MHz, una conexión USB, un conector de alimentación, un cabezal ICSP y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o enciéndalo con un adaptador de CA a CC o una batería para comenzar. Puede jugar con su UNO sin preocuparse demasiado por hacer algo mal, en el peor de los casos, puede reemplazar el chip por unos pocos dólares y comenzar de nuevo.

Tabla 11. Especificaciones Arduino UNO.

Board	Arduino UNO R3
Microcontrolador	ATmega328P
USB connector	USB-B
Pins	Built-in LED Pin: 13
	Digital I/O Pins: 14
	Analog input pins: 6
	PWM pins: 6
	UART: Sí
Communication	I2C: Sí
	SPI: Sí
	I/O Voltage: 5 V.
Power	Input voltage (nominal): 7 – 12 V.
	DC Current per I/O Pin: 20 mA
	Power Supply Connector: Barrel Plug
Clock speed	Main Processor: ATmega328P 16 MHz

---

	USB-Serial Processor: ATmega16U2 16 MHz
Memory	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensiones	Peso: 25 g                      Ancho: 53.4 mm Longitud: 68.6 mm

*Fuente:* Datasheet del Arduino UNO. (Arduino, 2022).

#### - **Arduino MEGA**

El Arduino Mega 2560 “es una placa de microcontrolador basada en el ATmega2560. Tiene 54 pines de entrada/salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos seriales de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un cabezal ICSP, y un botón de reinicio” (ARDUINO, 2022). Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o enciéndalo con un adaptador de CA a CC o una batería para comenzar. La placa Mega 2560 es compatible con la mayoría de los escudos diseñados para Uno y las placas anteriores Duemilanove o Diecimila.

*Tabla 12. Especificaciones Arduino MEGA.*

<b>Board</b>	<b>Arduino MEGA</b>
Microcontrolador	ATmega2560
Voltaje de Operación	5V
Voltaje de Entrada	7-12V
Digital I/O Pins	54
SRAM	8 KB
Dimensiones	101.52 mm x 53.3 mm
EEPROM	4 KB
Entradas Análogas	16

*Fuente:* Datasheet del Arduino MEGA. (Arduino, 2022).

#### - **Arduino Leonardo**

“La Leonardo se diferencia de todas las placas anteriores en que la ATmega32u4 tiene comunicación USB incorporada, lo que elimina la necesidad de un procesador secundario” (ARDUINO, 2022). Esto permite que el Leonardo aparezca en una computadora conectada como un ratón y un teclado, además de un puerto serial / COM virtual (CDC).

Tabla 13. Especificaciones Arduino Leonardo.

<b>Board</b>	<b>Arduino Leonardo</b>
Microcontrolador	ATmega32u4
Voltaje de Operación	5V
Voltaje de Entrada	7-12V
Digital I/O Pins	20
SRAM	8 KB
Dimensiones	68.6 mm x 53.3 mm
EEPROM	1 KB
Entradas Análogas	16

*Fuente:* Datasheet del Arduino Leonardo. (Arduino, 2022).

Tomando en cuenta los 3 Microcontrolador escogeremos el que más se adapte a los requerimientos como se muestra en la tabla a continuación donde haremos la comparación.

Tabla 14. Selección de Nodo de Sensores.

<b>Selección de Recolector de Datos</b>				
Hardware	RSSt	RSFs	RARa	Total
<i>Arduino UNO</i>	1	1	1	3
<i>Arduino MEGA</i>	1	0	1	2
<i>Arduino Leonardo</i>	1	0	0	1
<b>Selección: Arduino UNO – Cumple con todos los requisitos del prototipo.</b>				

### 3.7.2 Elección de Sensores

Los Sensores por usar son una parte esencial para el prototipo, ya que serán los que reciban los datos a para comprobar el estado del cultivo, por ello es importante escoger a los que más se adapten a los requisitos.

#### 3.7.2.1 Sensor PH

El Sensor de PH será ubicado en el agua de riego que se dirige hacia la planta el cual censará el valor de PH que tiene el agua, este dato debe ser idealmente de 4,5 a 5,5. Por ello se requiere el sensor más exacto.

- *SEN0161*

El medidor de pH analógico V2 está diseñado específicamente para medir el pH de la solución y reflejar la acidez o alcalinidad. Se usa comúnmente en diversas aplicaciones, como acuaponía, acuicultura y pruebas ambientales de agua.

Como una versión mejorada del medidor de pH V1, este producto mejora en gran medida la precisión y la experiencia del usuario. “El chip regulador de voltaje incorporado admite un amplio suministro de voltaje de 3,3 ~ 5,5 V, que es compatible con la placa de control principal de 5 V y 3,3 V. La señal de salida filtrada por hardware tiene un jitter bajo. La biblioteca de software adopta el método de calibración de dos puntos y puede identificar automáticamente dos soluciones tampón estándar (4.0 y 7.0), de manera simple y conveniente” (DFRobot Gravity, 2020).

Tabla 15. Características del Sensor de pH SEN0161.

<b>Placa de conversión de señal (transmisor) V2</b>	
Voltaje de suministro	3.3 ~ 5.5 V
Voltaje de Salida	0 ~ 3.0V
Conector de sonda	BNC
Conector de señal	PH2.0-3 P
Precisión de la medición	± 0.1 @ 25 °C
Dimensión	42mm*32mm/1.66*1.26in
<b>Sonda de pH</b>	
Tipo de sonda	Grado de laboratorio
Rango de detección	0 ~ 14
Rango de temperatura	5 ~ 60 °C
Punto cero	7 ± 0.5
Tiempo de respuesta	< 2 min
Resistencia interna	< 250 MΩ
Vida útil de la sonda	>0.5 año (La vida útil está relacionada con la frecuencia de uso y la escena)
Longitud del cable	100 cm

Fuente: Datasheet del Sensor SEN0161. (DFRobot Gravity, 2020).

#### - **SEN0169**

Es un medidor de pH analógico, especialmente diseñado para controladores Arduino y tiene una conexión y características simples, convenientes y prácticas incorporadas. Tiene un LED que funciona como indicador de encendido, un conector BNC y una interfaz de sensor PH2.0. Para usarlo, simplemente conecte el sensor de pH

con el conector BNC y conecte la interfaz PH2.0 al puerto de entrada analógica de cualquier controlador Arduino.

Tabla 16. Especificaciones Sensor SEN0169.

Board	Sensor SEN0169
Potencia del módulo	5V
Rango de medición	0 – 14 PH
Rango de Temperatura	0 – 60 °C
Precisión	± 0.1 PH
SRAM	8 KB
Dimensiones	43 mm x 32 mm
Tiempo de respuesta	≤ 1 min
Potenciómetro de ajuste de ganancia	Si

Fuente: Datasheet del Sensor SEN0169. (Arduino, 2022).

Tomando en cuenta ambos sensores de PH se verificará aquel que cumpla con los requisitos necesarios para el diseño.

Tabla 17. Selección de Sensor de PH.

Selección de Sensor de PH				
Hardware	RSSt	RSFs	RARa	Total
SEN0169	0	1	0	1
SEN0161	1	1	0	2
<b>Selección: SEN0161– Cumple con la mayoría de requisitos del prototipo.</b>				

### 3.7.2.2 Sensor Temperatura

Al igual que el Sensor de Temperatura se requerirá un sensor preciso que no interfiera el entorno de la maceta por ello se escogerá entre varios modelos, los rangos ideales para el cultivo son: dentro del vivero varia en la noche alrededor de 10° y durante el día alrededor de 30°.

- **DHT22**

El sensor de temperatura utiliza tecnología exclusiva de recolección de señales digitales y tecnología de detección de humedad, lo que garantiza su confiabilidad y estabilidad.

Sus elementos de detección están conectados con una computadora de un solo chip de 8 bits. “Cada sensor de este modelo está compensado por temperatura y calibrado en una cámara de calibración precisa y la calibración- El coeficiente se guarda en el tipo de programa en la memoria OTP, cuando el sensor está detectando, citará el coeficiente de la memoria” (Aosong Electronics Co, 2020).

El tamaño pequeño, el bajo consumo y la larga distancia de transmisión (20 m) permiten que DHT22 se adapte a todo tipo de ocasiones de aplicación difíciles. Empaquetado de una sola fila con cuatro pines, lo que hace que la conexión sea muy conveniente.

Tabla 18. Características del Sensor DHT22.

<b>Modelo</b>	<b>DHT22</b>	
Fuente de Alimentación	3.3-6V DC	
Señal de salida	Señal digital a través de un solo bus	
Elemento de detección	Condensador de polímero	
Rango de operación	Humedad 0-100% RH	Temperatura -40 ~ 80 Celsius
Precisión	Humedad +-2% RH (Max +-5% RH)	Temperatura <+-0.5 Celsius
Resolución o sensibilidad	Humedad 0.1% RH	Temperatura 0.1 Celsius
Repetibilidad	Humedad +-1% RH	Temperatura +-0.2 Celsius
Histéresis de humedad	+-0.3% RH	
Estabilidad a largo plazo	+-0.5% RH/año	
Período de detección	Promedio: 2s	
Intercambiabilidad	Totalmente Intercambiable	
Dimensiones	Talla pequeña 14*18*5.5mm	Tamaño Grande 22*28*5mm

Fuente: Datasheet del Sensor. (Aosong Electronics Co, 2020).

#### - **DHT11**

El DHT11 es un sensor digital de temperatura y humedad relativa de bajo costo y fácil uso. “Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica)” (Aosong Electronics Co, 2020). Utilizado en aplicaciones

académicas relacionadas al control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.

Tabla 19. Características del Sensor DHT11.

Modelo	DHT22	
Fuente de Alimentación	3-5V DC	
Señal de salida	Señal digital a través de un solo bus	
Rango de operación	Humedad 20-90%	Temperatura 0 ~ 50 Celsius
Precisión	Humedad 5%RH	Temperatura $\pm 2.0$ Celsius
Resolución o sensibilidad	Humedad 1%RH	Temperatura 0.1Celsius
Repetibilidad	Humedad $\pm 1$ %RH	Temperatura $\pm 0.2$ Celsius
Período de detección	Promedio: 1s	
Dimensiones	Talla pequeña 14*12*5mm	Tamaño Grande 16*12*5mm

Fuente: Datasheet del Sensor. (Aosong Electronics Co, 2020).

Ahora que se han analizados ambos sensores, se procede con la comparativa para cada uno de los requisitos de diseño y se escogerá la mejor opción, para este caso se tomará el DHT22.

Tabla 20. Selección de Sensor de Temperatura.

Selección de Sensor de Temperatura				
Hardware	RSSt	RSFs	RARa	Total
<i>DHT22</i>	1	1	1	3
<i>DHT11</i>	1	1	0	2
<b>Selección: DHT22– Cumple con todos los requisitos del prototipo.</b>				

### 3.7.2.3 Sensor Humedad del Suelo

Finalmente se usará un sensor de humedad del suelo que se aplicará en la maceta, el mismo no deberá ser muy invasivo con la maceta y ubicarse a una distancia prudente del suelo como se han mencionado en los requisitos del diseño. El rango ideal para este parámetro es de 80% al 100% de humedad.

#### - *SEN0193*

Este es un sensor de humedad del suelo capacitivo analógico que mide los niveles de humedad del suelo mediante detección capacitiva, es decir, la capacitancia varía



según el contenido de agua presente en el suelo. “La capacitancia se convierte en nivel de voltaje básicamente de 1,2 V a 3,0 V como máximo. La ventaja del sensor de humedad del suelo capacitivo es que están hechos de un material resistente a la corrosión que le da una larga vida útil” (DFRobot, 2020).

Tabla 21. Características del Sensor de Humedad de Suelo.

<b>Sensor capacitivo de humedad del suelo: SEN0193</b>	
Voltaje de funcionamiento	3.3 ~ 5.5 VDC
Voltaje de salida	0 ~ 3.0 VDC
Corriente de funcionamiento	5mA
Interfaz	PH2.0-3P
Dimensiones	3.86 x 0.905 inches (L x W)
Peso	15g

Fuente: Datasheet del Sensor. (DFRobot, 2020).

#### - **FC-28**

El sensor es ideal para monitorear el nivel de humedad de tus plantas y así recordar cuando necesitan ser regadas o incluso para realizar un sistema totalmente automatizado de riego añadiendo una válvula o una bomba de agua.

Tabla 22. Características del Sensor de Humedad de Suelo.

<b>Sensor capacitivo de humedad del suelo: FC-28</b>	
Voltaje de funcionamiento	3.3 ~ 5 VDC
Voltaje de salida	0 ~ 3.0 VDC
Corriente de funcionamiento	35mA
Superficie de electrodo	Estaño
Dimensiones	30 x 16 mm (L x W)
Peso	15g

Fuente: Datasheet del Sensor. (DFRobot, 2020).

Al igual que todos los casos anteriores se va a escoger la mejor de las opciones para el diseño, dado cada una de las características y se decide por el sensor de Humedad SEN0193 y se puede observar en la tabla 23.

Tabla 23. Selección de Sensor de Humedad de Suelo.

<b>Selección de Sensor de Humedad de Suelo</b>				
Hardware	RSSt	RSFs	RARa	Total

<i>FC-28</i>	0	1	0	1
<i>SEN0193</i>	1	1	1	3
<b>Selección: SEN0193– Cumple con todos los requisitos del prototipo.</b>				

### 3.7.3 Elección de Software

Escogidos el Hardware que va a intervenir dentro del prototipo se va a escoger el software necesario para la programación de los equipos y de la misma forma el software que se implementará dentro de la Nube y Base de Datos necesarios para el diseño del prototipo.

#### 3.7.3.1 Software para programación de Nodos

Tanto los Nodos de Sensores como los Recolectores de Datos son microcontroladores que requieren un Sistema Operativo para funcionar a continuación se detallan datos importantes sobre varios de ellos y finalmente la opción a escoger para el diseño.

##### - RIOT-OS

RIOT es un sistema operativo de código abierto basado en microkernel, diseñado para cumplir con los requisitos de los dispositivos de Internet de las cosas (IoT) y otros dispositivos integrados. “Estos requisitos incluyen una huella de memoria muy baja (kilobytes), alta eficiencia energética, capacidades en tiempo real, soporte para una amplia gama de hardware de bajo consumo, pilas de comunicación para redes inalámbricas y pilas de comunicación para redes cableadas” (Bacelli, Hahm, Günes, Wählich, & Schmidt, 2013).

El código base de RIOT está estructurado en cinco grupos.

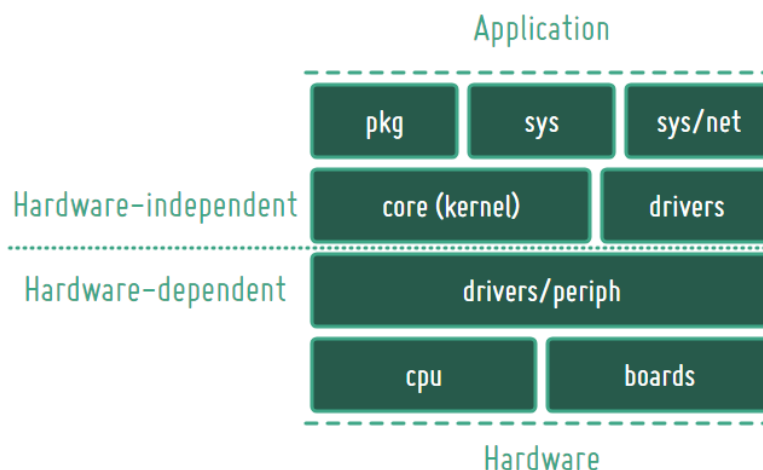


Figura 9. Estructura de RIOT-OS. Fuente: (Baccelli, Hahm, Günes, Wählisch, & Schmidt, 2013).

- El núcleo (core)
- Código específico de la plataforma (cpu; boards)
- Controladores de dispositivos (drivers)
- Bibliotecas y código de red (sys; pkg)
- Aplicaciones para demostrar características y para probar (examples; test)

Además, RIOT incluye una colección de scripts para varias tareas (dist), así como un entorno predefinido para generar esta documentación (doc).

Este Software fue seleccionado principalmente por su diseño hacia las aplicaciones IoT así como su compatibilidad y soporte para placas OpenMote-b.

OpenMote es una placa delgada que viene con un SoC de TI que combina un microcontrolador ARM Cortex-M3 con una radio IEEE802.15.4. Actualmente, RIOT admite la actualización del OpenMote mediante un adaptador Segger JLink JTAG o mediante USB, mediante el cargador de arranque de la placa (Baccelli, Hahm, Günes, Wählisch, & Schmidt, 2013).

Tabla 24. Características de la placa compatible en RIOT.

MCU	CC2538SF53
Familia	ARM Cortex-M3
Vendedor	Texas Instruments
RAM	32Kb
Flash	512Kb

Frecuencia	32MHz
FPU	no
Timers	4
ADCs	1x 12-bit (8 channels)
UARTs	2
SPIs	2
I2Cs	1
Vcc	2V - 3.6V

---

*Fuente:* Documentación RIOT-OS. (Baccelli, Hahm, Günes, Wählich, & Schmidt, 2013).

- ***Contiki***

Contiki-NG es un sistema operativo para dispositivos con recursos limitados en el Internet de las cosas. Contiki-NG contiene una pila de comunicación IPv6 de bajo consumo compatible con RFC, que permite la conectividad a Internet. El sistema se ejecuta en una variedad de plataformas basadas en arquitecturas de eficiencia energética como ARM Cortex-M3/M4 y Texas Instruments MSP430.

El repositorio Contiki-NG está estructurado de la siguiente manera:

- *os*: contiene el código Contiki-NG real. Incluye las primitivas de los sistemas, como procesos y temporizadores, la pila de redes y todas las bibliotecas y servicios.
- *arch*: Contiene todo el código dependiente del hardware. Esto incluye controladores de CPU, dispositivos y plataformas.
- *ejemplos*: contiene proyectos de ejemplo listos para usar. Muestra cómo usar redes, bibliotecas y servicios de almacenamiento.
- *herramientas*: contiene herramientas, que no deben incluirse en un firmware de Contiki-NG, sino que están destinadas a ejecutarse en una computadora.
- *pruebas*: contiene todas las pruebas de integración continua que se ejecutan para cada solicitud de extracción y combinación, para garantizar que no haya regresión.

A continuación, se realizará la tabla comparativa para escoger uno de los Sistemas Operativos que estará incluido en la programación del prototipo.

*Tabla 25. Selección de Software.*

<b>Selección de Software</b>				
Hardware	RSSt	RSFs	RARa	Total

<i>RIOT - OS</i>	0	1	1	2
<i>Contiki-NG</i>	0	0	1	1
<b>Selección: RIOT - OS – Cumple con la mayoría de los requisitos del prototipo.</b>				

### 3.7.3.2 Infraestructura como servicio (IaaS)

También conocida como Infrastructure as a Service (IaaS) sirve para almacenar todos los datos recibidos dentro del prototipo se van a guardar en una Base de Datos en la nube para posteriormente ser publicados y recibidos en una página web, para ello necesitamos una Infraestructura que trabaje como servicio otorgándonos estos beneficios.

#### - **Google Cloud**

“La infraestructura como servicio (también conocida como IaaS) es la disponibilidad bajo demanda de recursos de computación muy escalables como servicios a través de Internet. De este modo, se elimina la necesidad de aprovisionar, configurar o gestionar los recursos y solo se paga por el uso que se hace de estos recursos” (Google Cloud, 2022).

Las principales Ventajas son:

**Económico:** “Como los recursos de IaaS se usan bajo demanda y las empresas solo tienen que pagar por los recursos de computación, almacenamiento y redes que se usan” (Google Cloud, 2022).

**Eficaz:** Los recursos de IaaS suelen estar disponibles para las empresas cuando los necesitan.

**Aumenta la productividad:** Como el proveedor de nube es responsable de la configuración y el mantenimiento de la infraestructura física subyacente, los departamentos de TI de las empresas pueden ahorrar tiempo y dinero.

**Fiabilidad:** “El modelo de IaaS no tiene ningún punto único de fallo. Si un componente de los recursos de hardware falla, en la mayor parte de los casos el servicio seguirá estando disponible” (Google Cloud, 2022).

#### - **Amazon Web Service (AWS)**

AWS es una plataforma ubicada en la nube que contiene más de 200 servicios de análisis de datos, con millones de usuarios y empresas grandes que están usando sus

servicios y así reducir los gastos en implementos de hardware en cada empresa, a su vez para aumentar la agilidad e innovar en los servicios que prestan.

Las ventajas principales son:

**Mayor Funcionalidad:** cuenta con una insuperable cantidad de servicios y características dentro de sus opciones de proveedores de la nube, al igual que ofrece.

**Muchos Usuarios:** Actualmente AWS cuenta con una gran comunidad de usuarios activos y además una gran cantidad de socios, en todos los sectores que adaptan su tecnología con AWS.

**Más Seguro:** se diseñó para ser un entorno flexible y seguro, la infraestructura principal está creada para cumplir los requisitos de seguridad para el ejército, bancos y otras organizaciones.

- **Microsoft Azure**

Microsoft Azure IoT Central es una superficie de UX y API preparada para conectar y administrar dispositivos a escala, que ofrece datos confiables para obtener información empresarial. Preensambla ofertas de plataforma como servicio (PaaS), reuniendo cada servicio debajo de él para una oferta de IoT fácil de configurar, completa y segura.

Características principales:

- Conectividad rápida entre dispositivos IoT y la nube
- Administración centralizada para reconfigurar y actualizar dispositivos fácilmente
- Visualizaciones y análisis para dar sentido a los datos de IoT, desde el panorama general hasta los pequeños detalles
- Extensibilidad para unir las aplicaciones comerciales y los datos de IoT, convirtiendo los conocimientos en acción

Finalmente se optará escoger la Infraestructura como servicio de acuerdo con los requisitos de diseño, optando por el más confiable y que mejor características nos ofrezca, en este caso se escogerá Google Cloud como IaaS.

Tabla 26. Selección de Software.

Selección de Software				
Hardware	RSSt	RSFs	RARa	Total

<i>Microsoft Azure</i>	0	0	1	1
<i>Amazon Web Service</i>	1	0	1	2
<i>Google Cloud</i>	1	1	1	3
<b>Selección: Google Cloud – Cumple con la mayoría de requisitos del prototipo.</b>				

### 3.8 Ubicación de los Nodos

El Invernadero tiene una extensión de 200 metros cuadrados de los cuales únicamente se usan 50 metros cuadrados para la plantación de los Arándanos, ya que la topología a ser usadas es una Red Mesh, la ubicación de cada uno de los elementos es primordial para que cubra toda la extensión y poder evitar la pérdida de los paquetes ya que esta topología permite redundancia para poder llegar al nodo central. Se simulará la existencia de línea de vista y una zona de fresnel entre estos nodos en la cual no existirá ningún obstáculo, como requerimiento es imprescindible ubicar los nodos dentro de las macetas del cultivo, en la siguiente figura se observa la ubicación de cada uno de los Nodos y el Nodo Central.



Figura 10. Ubicación en el mapa.

Para simular si existe línea de vista sin obstrucciones ni zona de fresnel se obtuvo ayuda del simulador Radio Mobile, y para garantizar que no existan pérdidas de paquetes, se van a establecer las posiciones de cada sensor en el invernadero, como se observa en la figura 11.

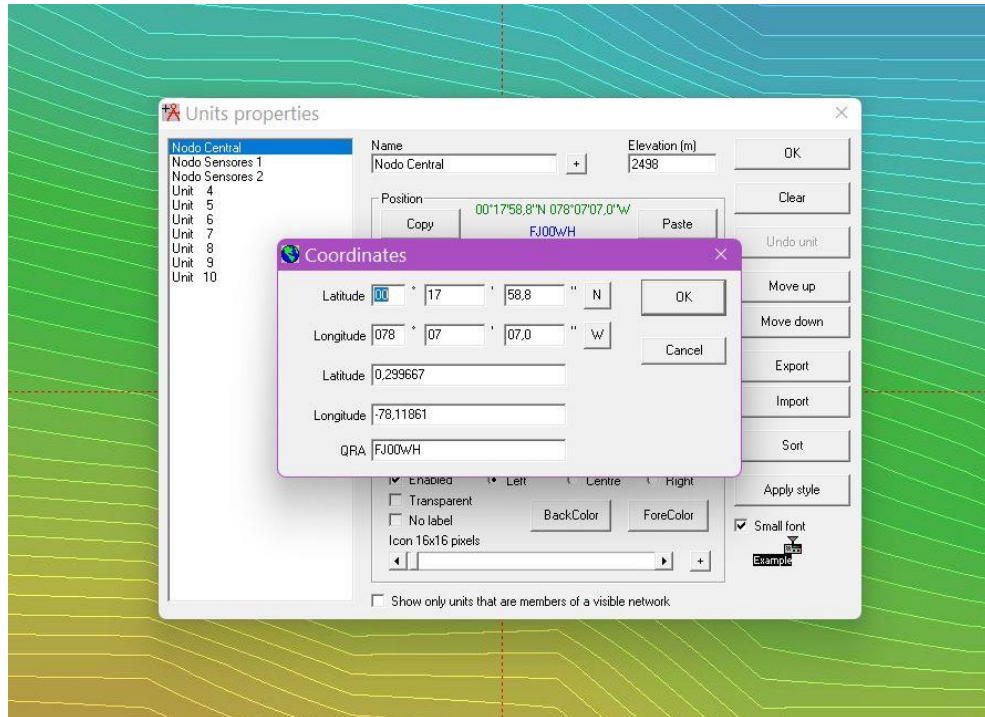


Figura 11. Coordenadas de los sensores.

A continuación, se crean cada uno de los nodos de sensores que van a formar parte del prototipo.

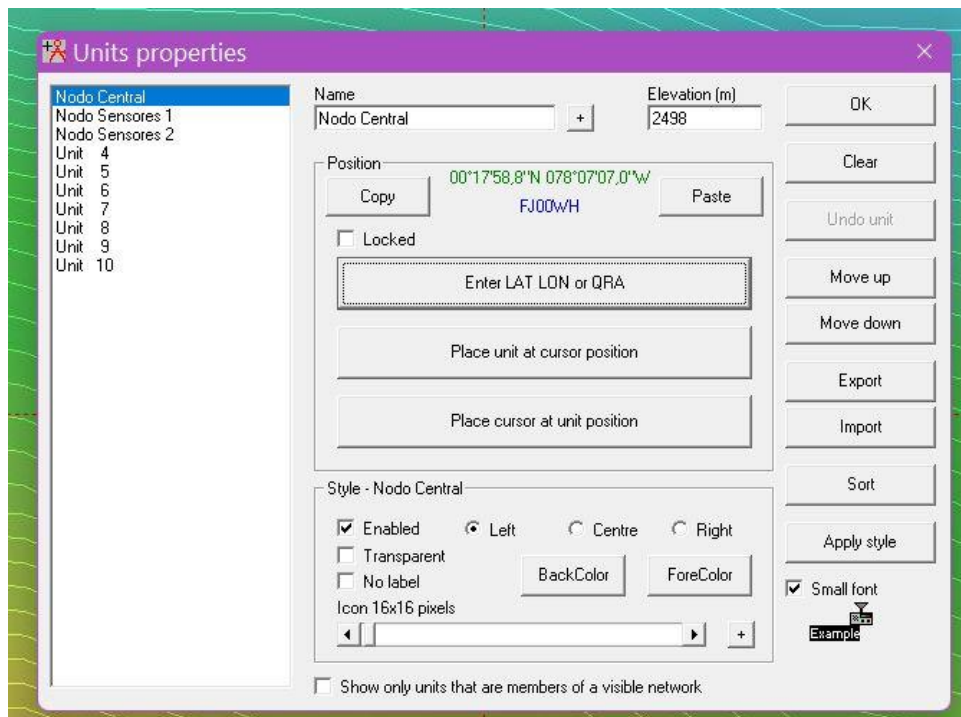


Figura 12. Ubicación de cada sensor.

Para la simulación se tomará en cuenta los siguientes valores a configurar en cada sensor junto a sus valores por defecto, se detallan en la siguiente tabla.



Tabla 27. Parámetros de Simulación Radio Mobile.

Parámetros	Nodo Central	Nodos Sensor 1	Nodo Sensor 2
Ubicación	Latitud: 0.299667 Longitud: -78.11861	Latitud: 0.29994 Longitud: -78.11855	Latitud: 0.30025 Longitud: -78.11892
Elevación	2498 [m]	2495 [m]	2492 [m]
Frecuencia	Min: 863 MHz Max: 869 MHz	Min: 863 MHz Max: 869 MHz	Min: 863 MHz Max: 869 MHz
Polarización	Vertical	Vertical	Vertical
Tx Power	3 [dBm]	3 [dBm]	3 [dBm]
Umbral de Recepción	-107 [dBm]	-107 [dBm]	-107 [dBm]
Ganancia de Antena	2,15 [dBi]	2,15 [dBi]	2,15 [dBi]
Tipo de Antena	Dipolo	Dipolo	Dipolo

Al ser creados los nodos, se va a crear la red tipo Mesh y garantizar el envío de los paquetes desde cada sensor hacia el nodo central.

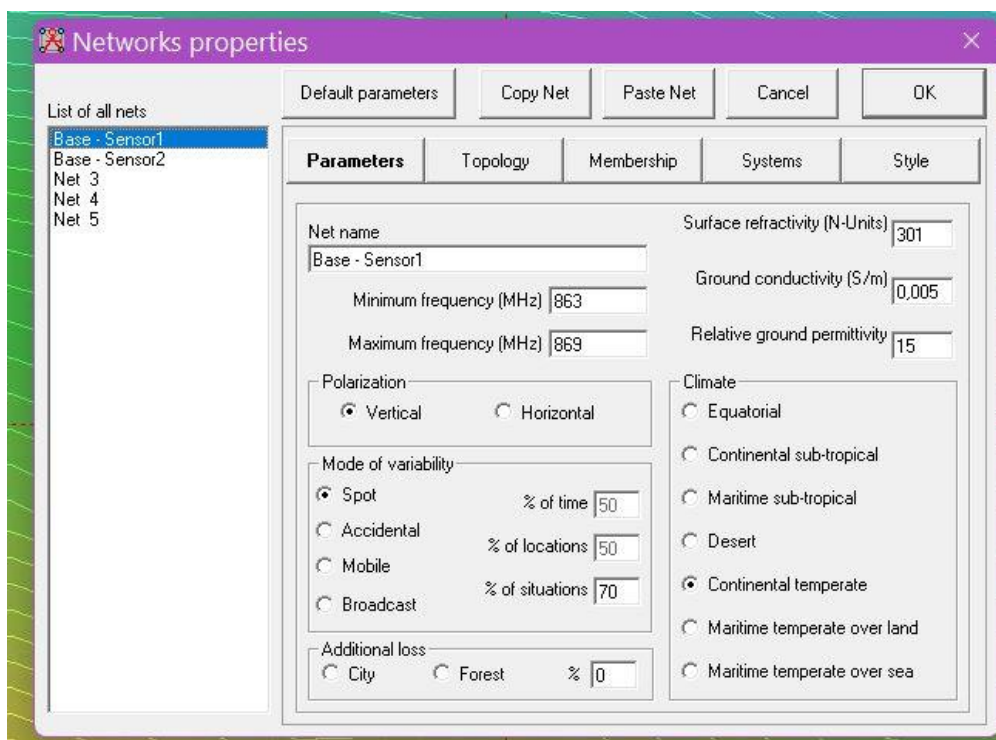


Figura 13. Añadir tipo de Red.

El resultado se puede observar en la figura 14, en donde se muestra la red tipo Mesh, y lo que implica la comunicación entre nodos será óptima, esta herramienta nos permite observar la línea de vista y zona de fresnel entre estos nodos y con el fin de corregir

errores se pueden variar los parámetros de altura y así obtener el nivel de recepción aceptable en este caso 45.3 [dB] en la figura 15 se verá también para la comunicación entre el Nodo Central y Nodo Sensor 2, cuya recepción es de 34.6 [dB].

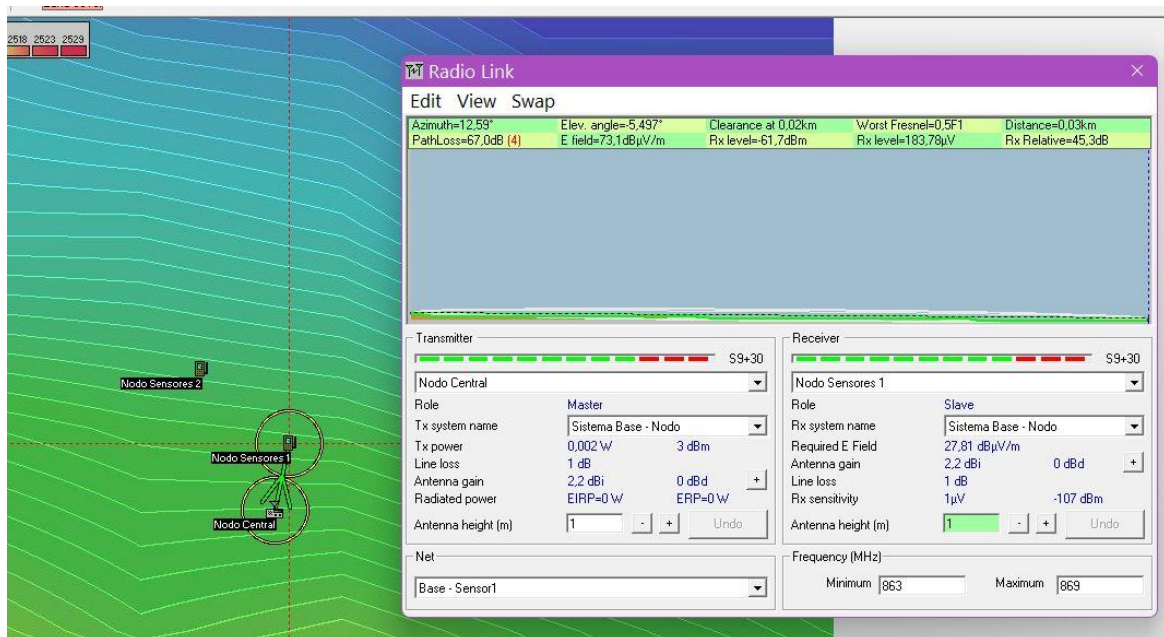


Figura 14. Línea de Vista y Zona de Fresnel Nodo 1.

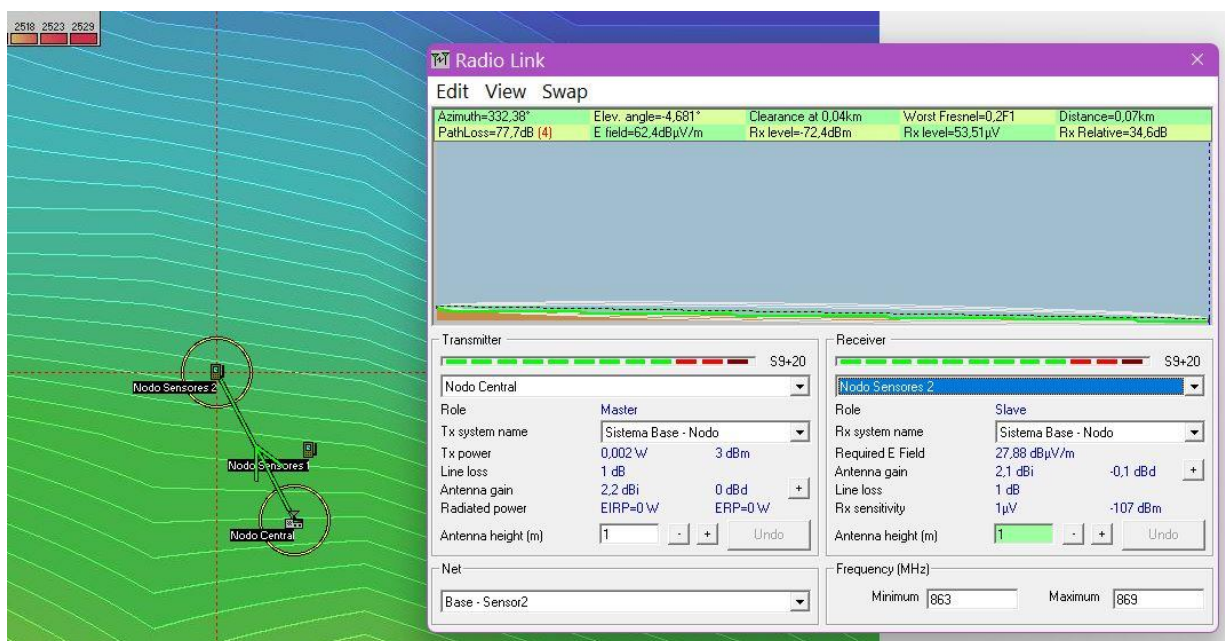


Figura 15. Línea de Vista y Zona de Fresnel Nodo 2.

### 3.9 Diseño del Prototipo

Ya que se han identificado los parámetros a medir en la planta y los equipos necesarios para el mismo se procederá a realizar el diseño de los Nodos de Sensores que recolectarán todos los datos del cultivo y esos a su vez serán enviados a través de

comunicación Serial hacia el equipo OpenMote-b que enviará los datos usando el protocolo MQTT-SN hacia el Gateway en el Borde Route donde se recopilarán todos los datos de cada Estación. Véase la figura 16 en el siguiente apartado donde se identificará este proceso en un diagrama de bloques.

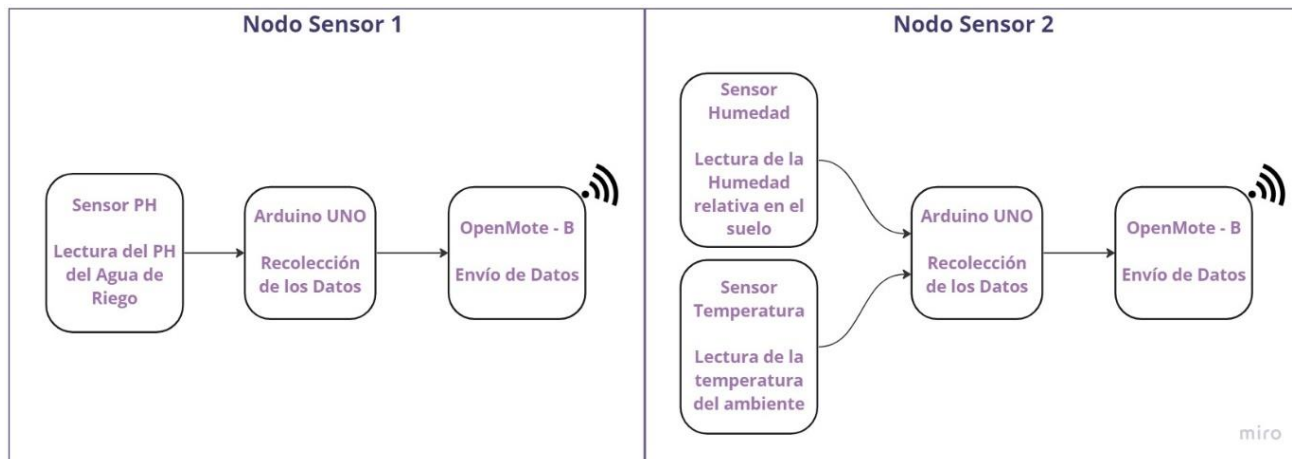


Figura 16. Diagrama de Bloques Nodos Sensores.

Por otro lado, el Borde Router será el encargado de la recepción de estos datos y el enrutamiento de los mismos hacia el Cloud para su posterior visualización en la siguiente figura 17 se puede observar el diagrama de bloques del funcionamiento del Router de Borde.

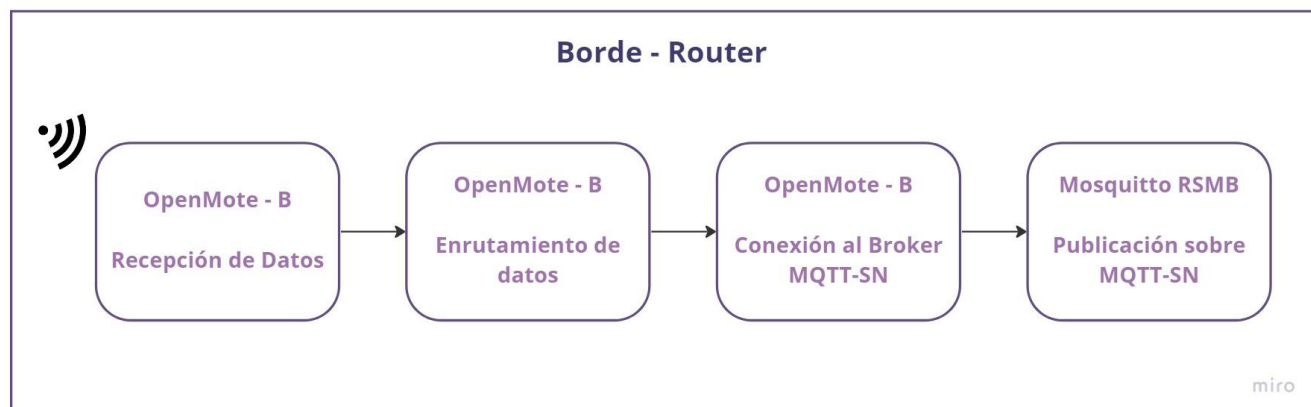


Figura 17. Diagrama de Bloques Borde Router.

Una vez se conoce el funcionamiento interno del sistema se puede proceder con el diseño de este, para dicho caso se presentan el diagrama de flujo para poder describir el proceso de cada etapa de la programación la cual se efectúa con un Arduino UNO. Primeramente, se inicia con los Nodos Sensores; quienes, como se mencionó anteriormente, recopilarán toda la información. Véase figura 18 y 19.

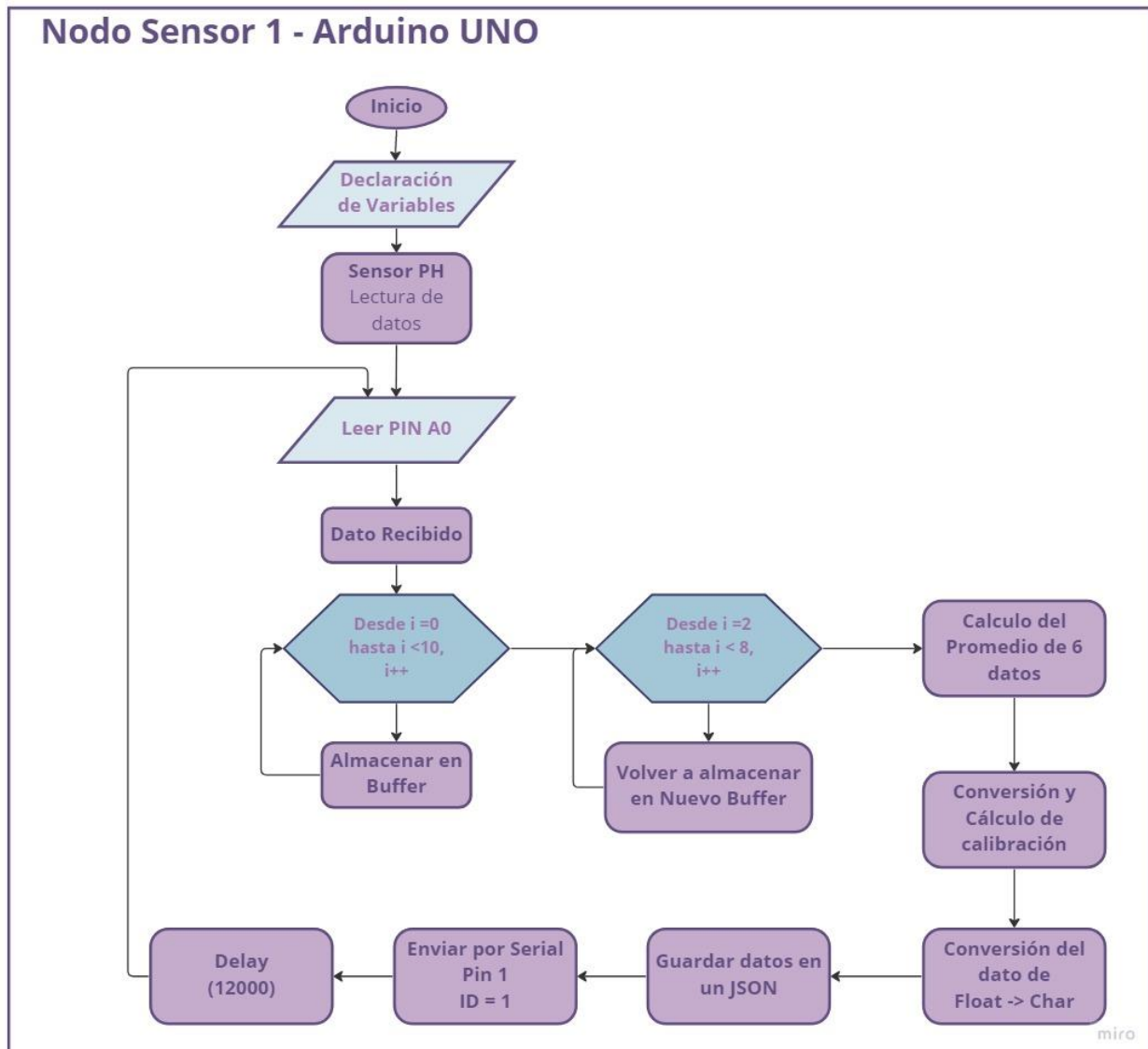


Figura 18. Diagrama de Flujo Nodo de Sensores 1.

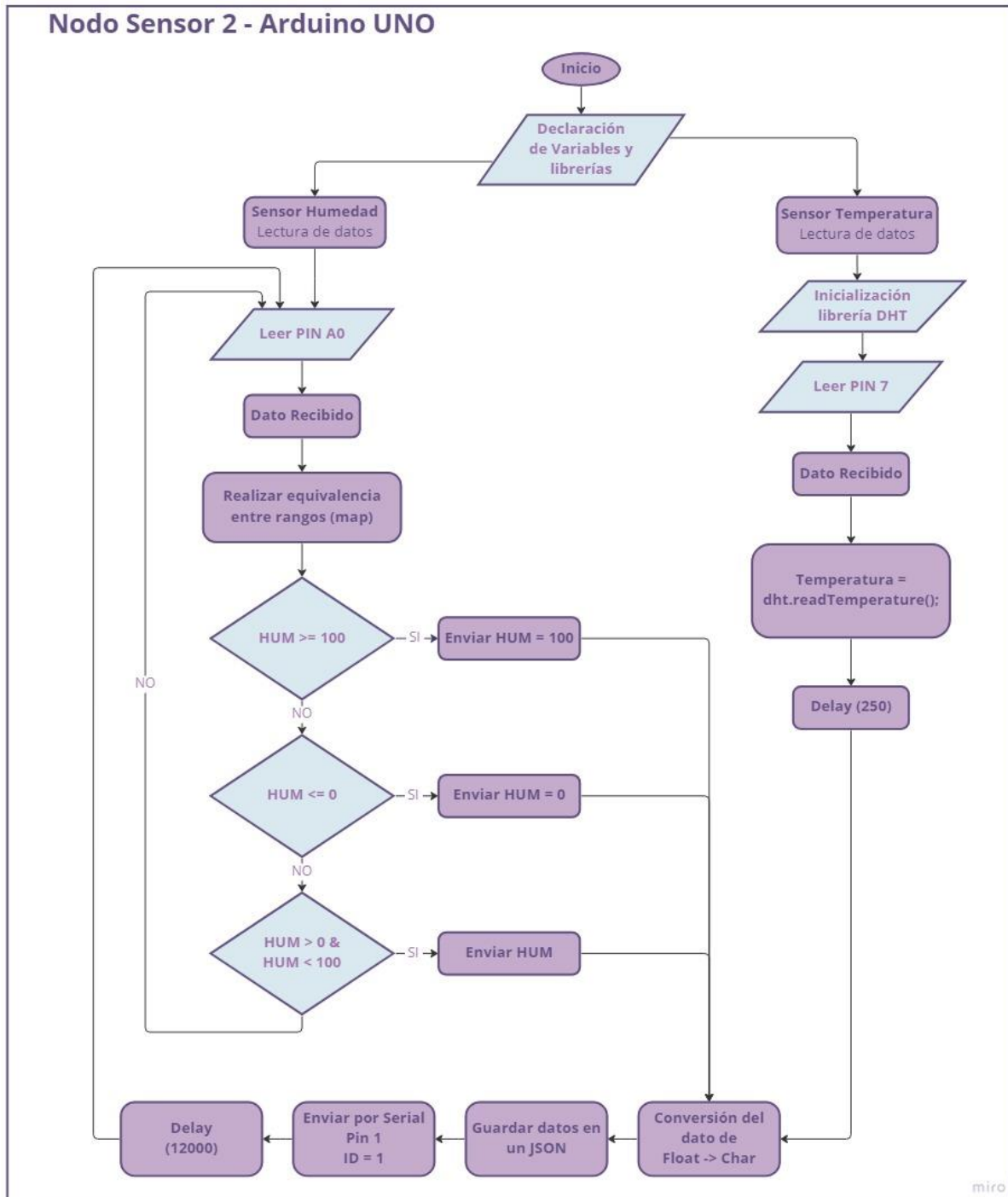


Figura 19. Diagrama de Flujo Nodo Sensor 2.

Siguiendo el diagrama de bloques visto en la figura 20 ahora que recolectamos los datos de los sensores en el Arduino UNO se enviarán por Comunicación Serial hacia la placa OpenMote-b para el envío posterior de los datos, y se describe este proceso en la figura 14 mostrada a continuación.

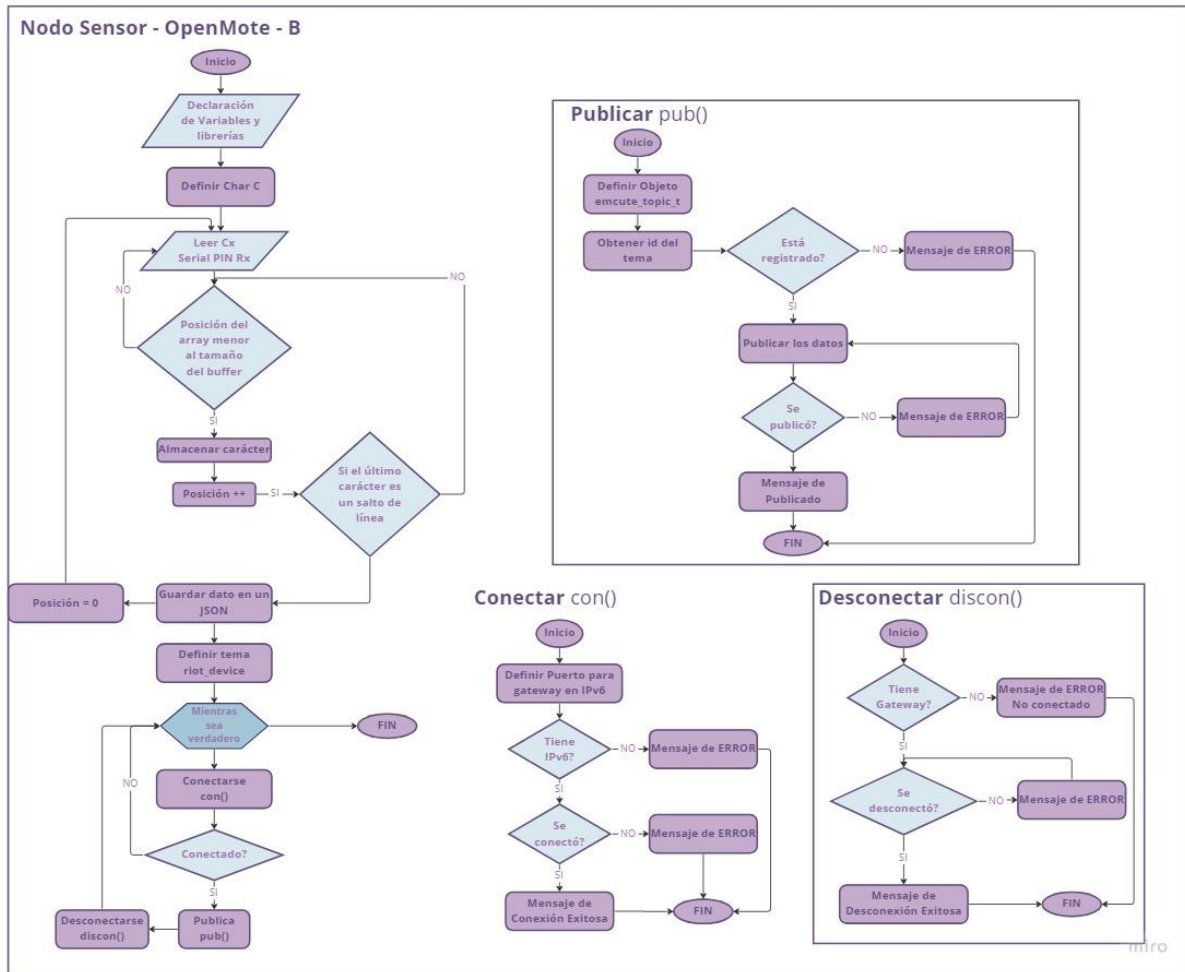


Figura 20. Diagrama de Flujo de los Nodos Sensores en OpenMote-B.

Podremos observar el diagrama de conexiones para cada uno de los Sensores y se observa en las figuras 21 y 22.

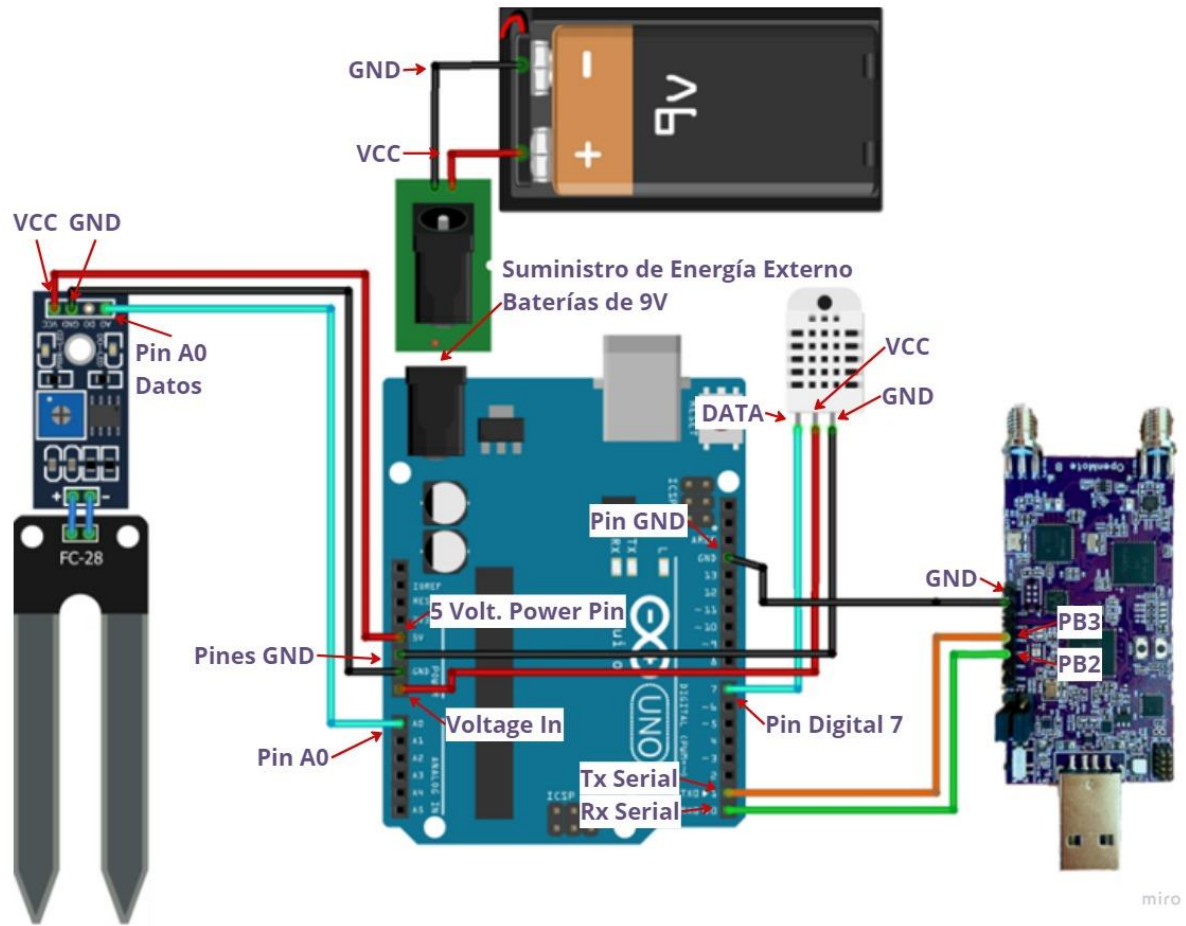


Figura 21. Diagrama Circuitual Nodo Sensor 1.

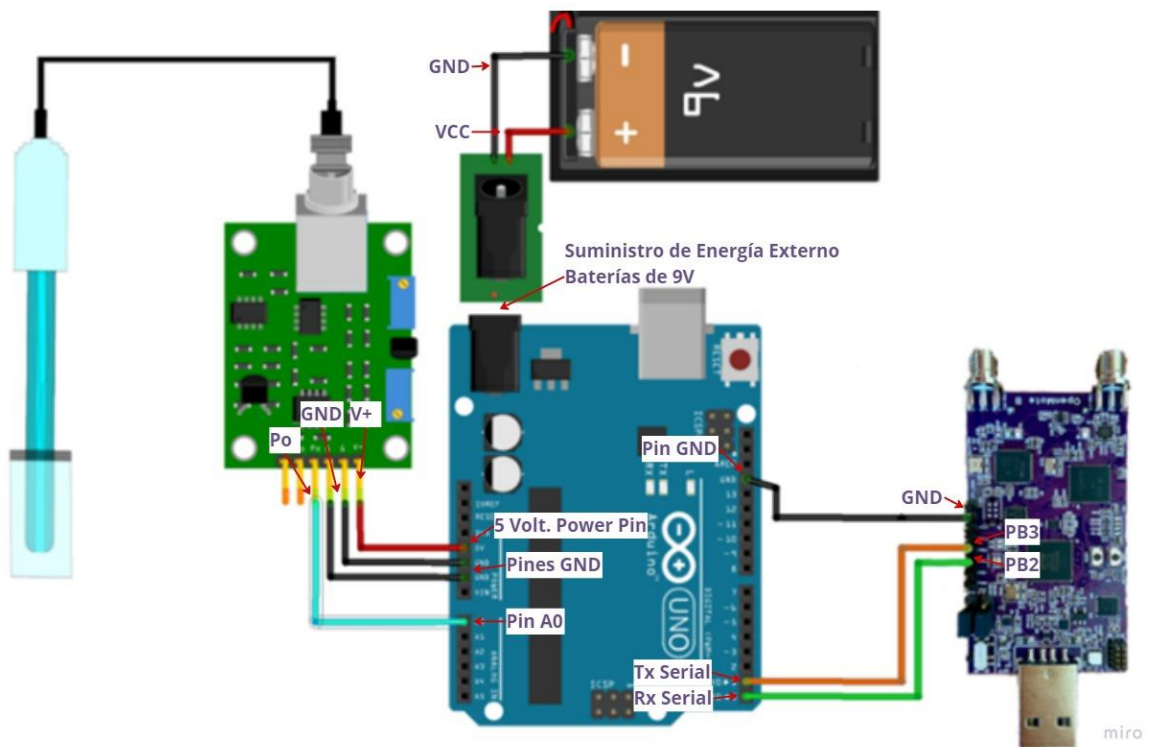


Figura 22. Diagrama Circuitual Nodo Sensor 2.

A continuación, se recibirán estos datos en el Borde Router que será el encargado de enviarlos al Broker que es efectuado en Mosquitto RSMB, los valores que serán publicados sobre MQTT-SN, y luego necesitan llegar a la nube vía MQTT, a través de un puente entre ambos protocolos, este proceso podemos observarlo en la figura 23.

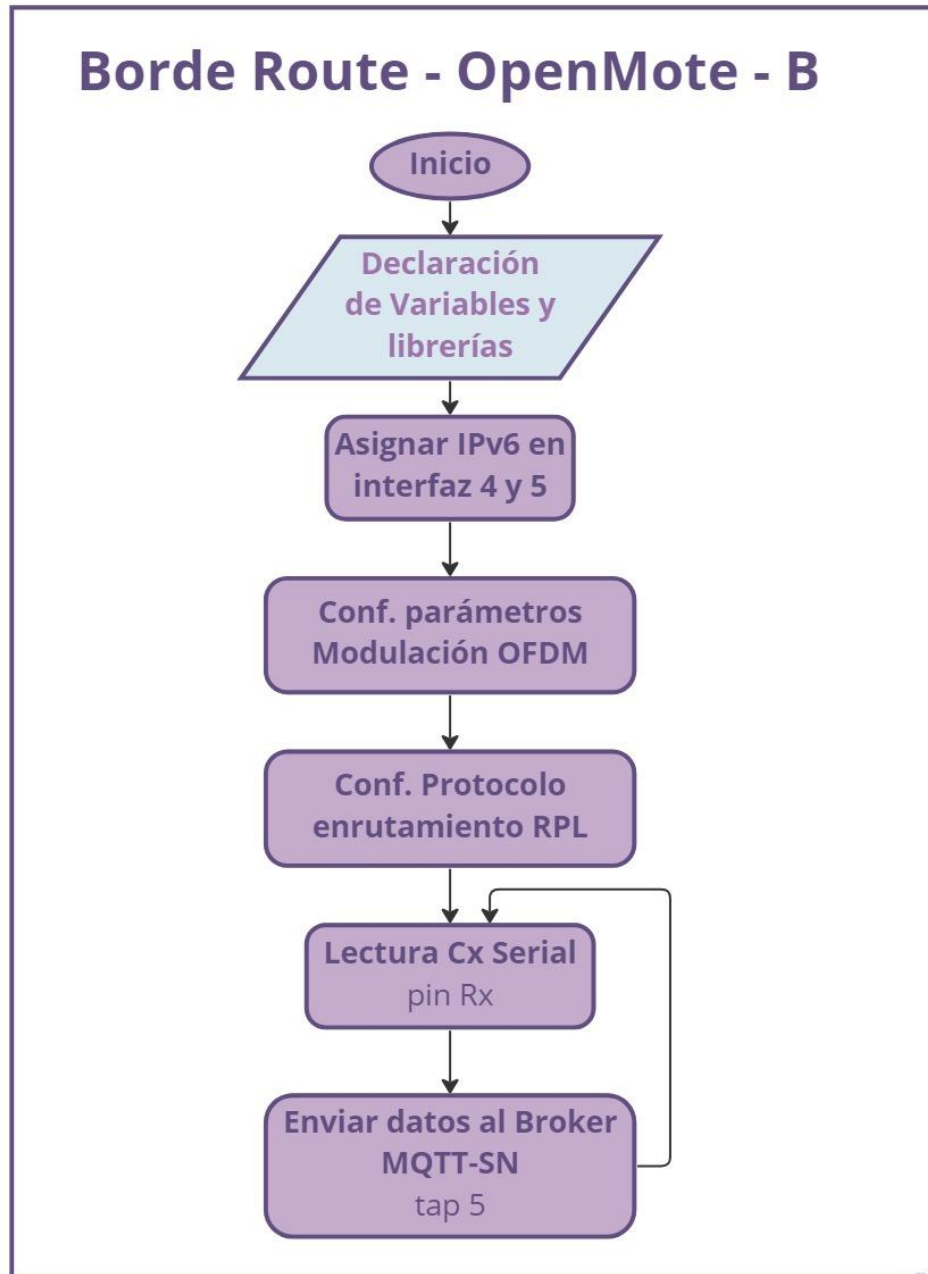


Figura 23. Diagrama de Flujo Borde Router.

Con los datos listos para ser analizados lo primero que podemos hacer es configurar el intermediario MQTT-SN. Se ha utilizado Mosquitto Really Small Message Broker (RSMB) como Broker en el lado de los sensores y a su vez la conexión de un Puente



Transparente que realice el cambio de MQTT-SN hacia MQTT, como lo explica el diagrama de bloques en la figura 24.

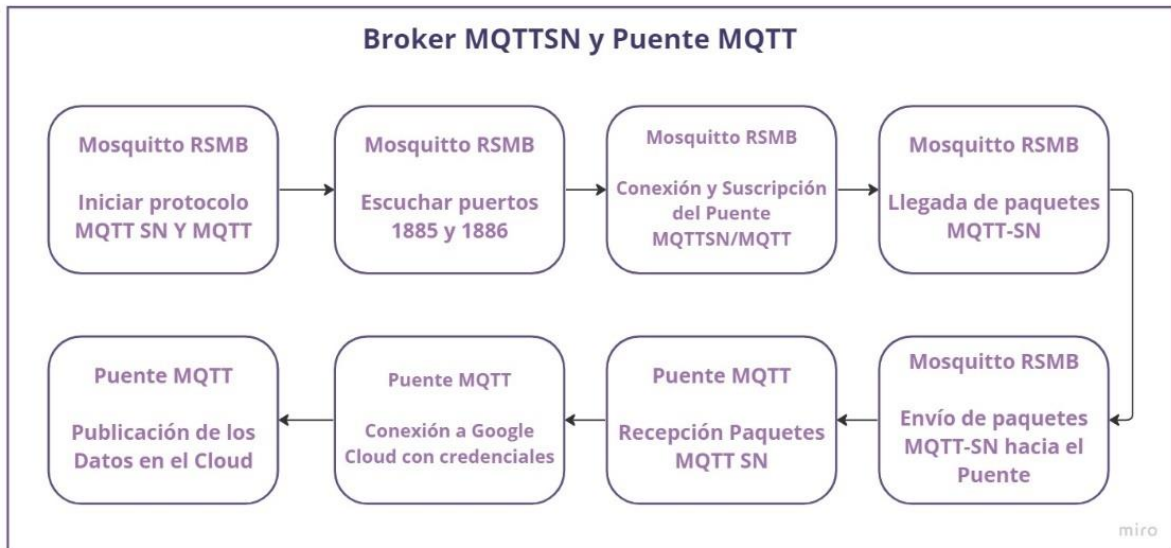


Figura 24. Diagrama de Bloques Broker MQTTSN y Puente MQTT.

Como se mencionó anteriormente dado que no es posible el puente directo entre el RSMB y el servicio en la nube se creará un puente que cumplirá el objetivo de recibir los mensajes con la conexión a temas deseados y enviarlos a Google Cloud Service y posteriormente ingresarlos en una base de Datos, en la figura 25 se puede observar la lógica de programación representada en un Diagrama de Flujo.

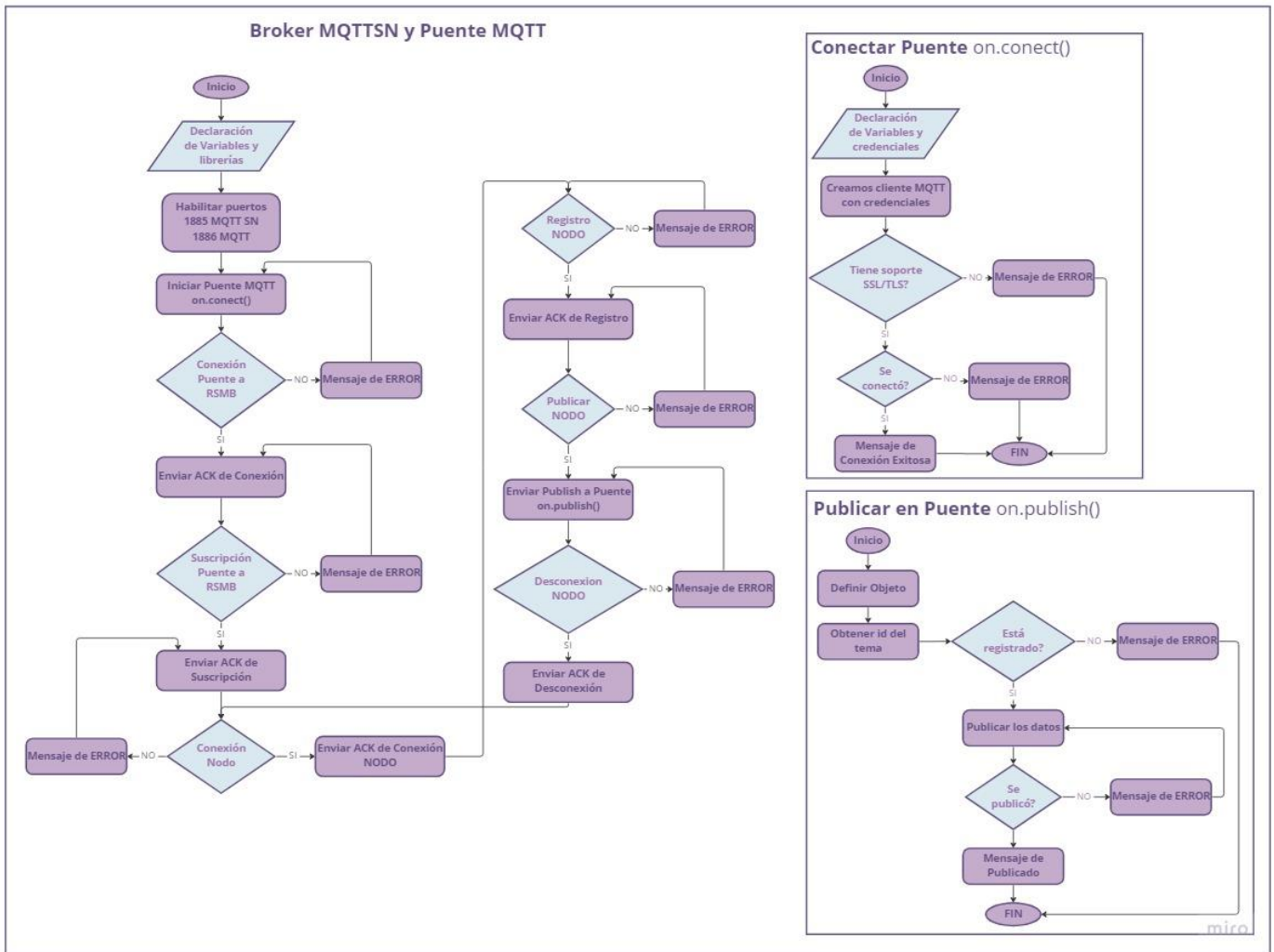


Figura 25. Diagrama de Flujo Broker MQTTSN y Puente MQTT.

Una vez los datos han sido publicados en la IaaS, se va a procederá a almacenar los datos en una base de datos la cual a su vez realizará la publicación en una página web que tendrá por objetivo visualizar todos los datos de cada sensor en una tabla y mostrar los últimos 10 valores igualmente, se puede observar en diagrama de bloques de la figura 26.

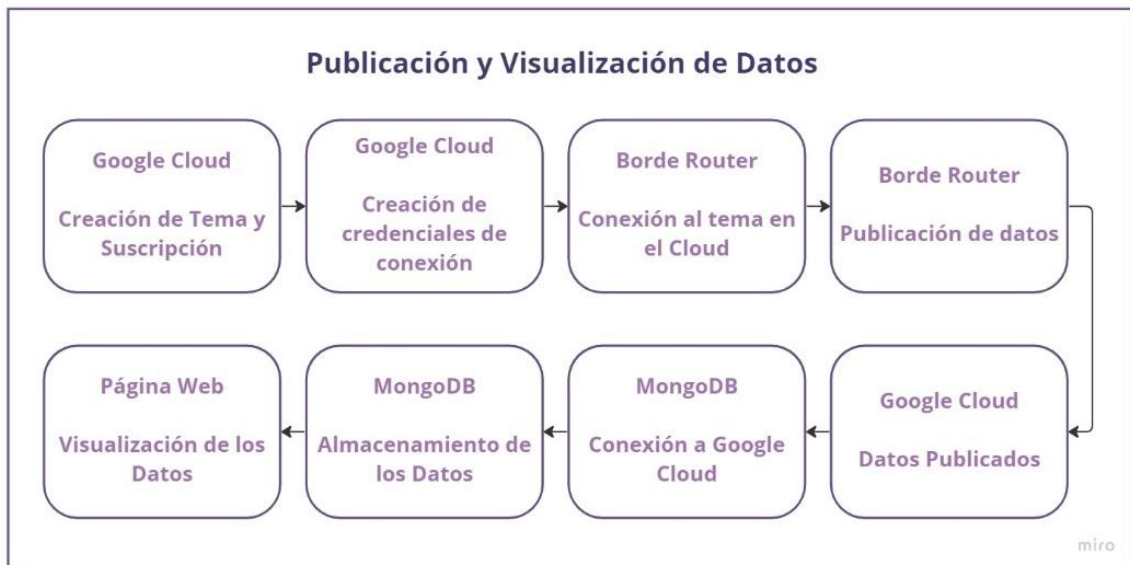


Figura 26. Diagrama de bloques Google Cloud, Mongo DB.

Entendido el funcionamiento de cada uno de los pasos a realizar se procede a diseñar un diagrama de flujo con la lógica de programación que constará en el desarrollo del prototipo.

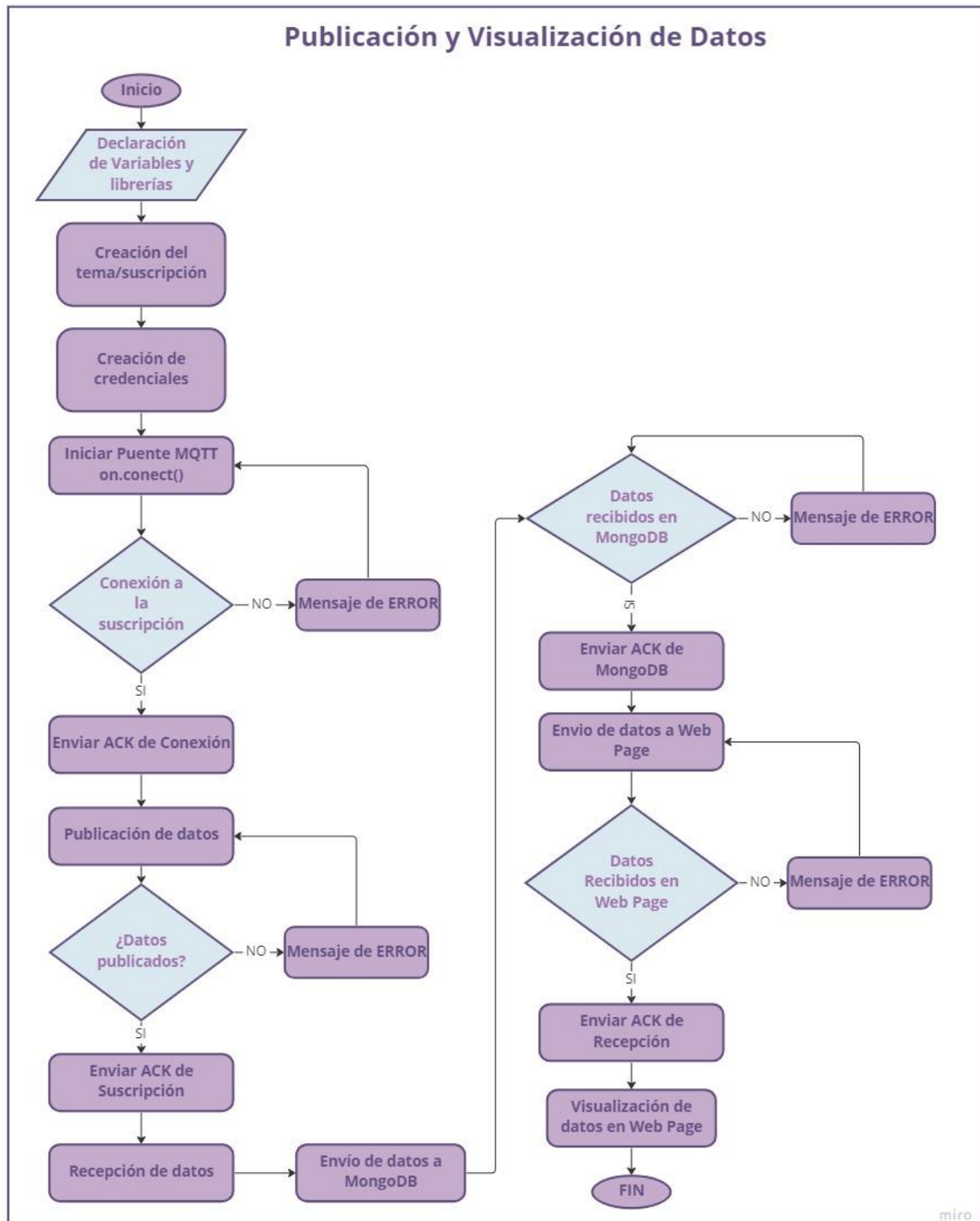


Figura 27. Diagrama de Flujo de Publicación y Visualización de Datos.

### 3.10 Arquitectura General del Sistema

La WSN está constituida por nodos sensores y un nodo central o Gateway el cual actuará como medio de comunicación entre los nodos de sensores y la estación base, y que de esa forma los datos extraídos sean transmitidos a la nube y analizados, como se

detalla en la Figura 28. Iniciando desde cada uno de los sensores se puede observar que cada uno se comunicará con el nodo que recolectará todos los datos de su sección de sensores esta comunicación se dará a través de Comunicación Serial conectando los sensores en los pines respectivos del Microcontrolador el cual envía los datos a través de conexión inalámbrica; en este caso usando el estándar Wi-SUN, sobre el protocolo MQTT-SN (MQTT for Sensor Networks). MQTT es un protocolo de mensajería estándar de OASIS para Internet de las cosas (IoT). Está diseñado como un transporte de mensajería de publicación / suscripción extremadamente liviana que es ideal para conectar dispositivos remotos con una huella de código pequeña y un ancho de banda de red mínimo. Una vez recibe el Gateway OpenMote-B el cual contiene valores que serán publicados sobre MQTT-SN, llegando así a MQTT-SN Broker que será montado sobre la plataforma Mosquitto; es un gestor de datos en la nube. Recibe todos los datos que están siendo publicados, y decide qué suscriptores deben recibirlos, según el interés de cada uno de ellos. A continuación, necesita llegar a la nube a través de MQTT, por ello necesitamos un puente entre el protocolo MQTT-SN y el MQTT. El propósito es recibir mensajes que solo se conecten a los Clientes deseados y enviarlos a Google Cloud a través de MQTT; además, para el correcto funcionamiento de la Aplicación Web (en la que se mostrarán los valores de los sensores) montada en la plataforma Node JS., el puente inserta todos los datos en una tabla en el servicio MongoDB.

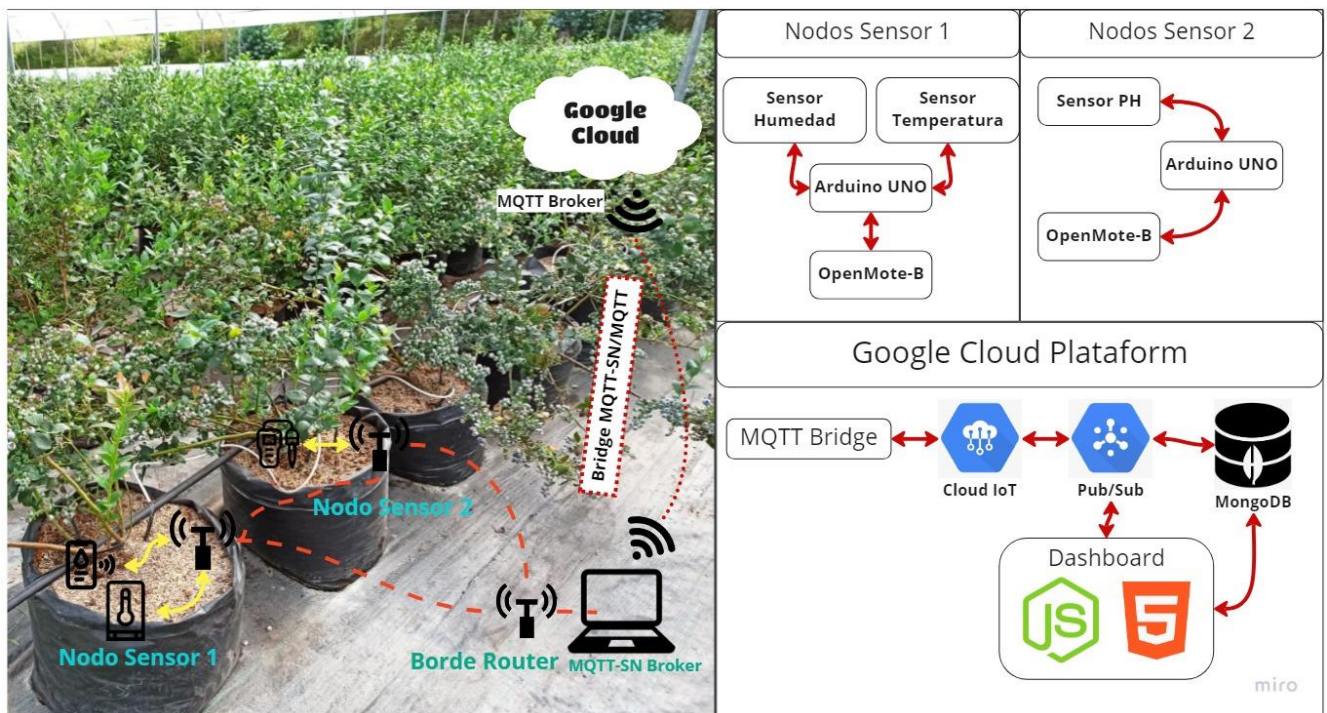


Figura 28. Arquitectura del Sistema.

### 3.10.1 Arquitectura de Internet of Things

Los administradores usan la arquitectura IoT para administrar y admitir dispositivos IoT. Los dispositivos IoT pueden ser cualquier cosa, desde una bombilla conectada a Internet hasta sensores de seguridad de presión en una planta química.

Estos dispositivos utilizan pequeños sensores para recopilar datos sobre su entorno y enviar esos datos a un servidor para su procesamiento. Los servidores procesan estos datos para crear información y conocimientos para las empresas. Muchas veces, esta información se usa para automatizar tareas que mejoran el tiempo de actividad y la eficiencia en múltiples sistemas comerciales.

#### 3.10.1.1 *Capa de Dispositivo*

En esta capa se conecta todo con el Internet y es la infraestructura principal para IoT, va a consistir en la tecnología inalámbrica y la interfaz de los sensores que recolectarán y transmitirán las señales al controlador principal, el cual se basará en IEEE 802.15.4. Por eso se presenta aquí los sensores y el punto donde se recolectará toda esa información, armando el circuito como vemos la figura 23 - 24 siguiendo el diagrama de conexión de la figura 29 – 30.



*Figura 29. Nodo Sensor 1 ubicado en maceta.*



*Figura 30. Nodo Sensor 2 ubicado en maceta.*

En este caso fue diseñado con un Arduino UNO, para el primer NODO tenemos la conexión del Sensor de PH, el código se presenta en la figura 31. En la primera parte está la calibración del Sensor donde se hace un promedio de 10 datos recibidos y se resta el voltaje de entrada, finalmente se arma un archivo JSON que se envía por comunicación serial hacia el Nodo Sensor.

```

Serial.begin(115200);
}

float lecturaPH() {
  for (int i = 0; i < 10; i++) {
    buf[i] = analogRead(analogInPin);
    delay(30);
  }
  for (int i = 0; i < 9; i++) {
    for (int j = i + 1; j < 10; j++) {
      if (buf[i] > buf[j]) {
        temp = buf[i];
        buf[i] = buf[j];
        buf[j] = temp;
      }
    }
  }
  avgValue = 0;
  for (int i = 2; i < 8; i++) {
    avgValue += buf[i];
  }
  float pHVol = (float)avgValue * 5.0 / 1024 / 6;
  float pHValue = -5.70 * pHVol + calibration;
  delay(200);
  return (pHValue);
}

void loop() {
  //Pasar de float a string es medio raro pero este combo es bueno
  char phChar[32]; //Almacena la lectura en un char array para conversion
  float phFloat = lecturaPH(); //Almacena en un float el return de la funcion
  dtostrf(phFloat, 4, 2, phChar); //Convierte un float(phFloat) de minimo 4 de longitud con 2 dec
  String phString(phChar); //Se crea una variable String con el contenido del char(phChar)
  sprintf(nodo2, "{\"id\":\"2\",\"datetime\":\"2022-06-13 22:10:10\",\"ph\":\"%s\"}\n", phChar);
  Serial.write(nodo2);
  delay(12000);
}

```

Figura 31. Código de Arduino Nodo Sensor 1 lectura de datos.

El mismo procedimiento se hace con el Nodo Sensor 2, en este caso son dos tipos de sensores: humedad y temperatura, inicialmente se ve la calibración para el sensor de humedad haciendo un mapeo comparando lo datos de humedad de aire y agua, la segunda parte es de la temperatura en el cual usamos la librería de Arduino mismo. Finalmente enviamos por un archivo JSON hacia el Nodo de Sensores de la misma forma que en el caso anterior. Véase la figura 32.



```

float lecturaHUMEDAD() {
  soilMoistureValue = analogRead(A0); //put Sensor insert into soil
  Serial.println(soilMoistureValue);
  soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
  if (soilmoisturepercent >= 100) {
    soilmoisturepercent = 100;
    return (soilmoisturepercent);
  } else if (soilmoisturepercent <= 0) {
    soilmoisturepercent = 0;
    return (soilmoisturepercent);
  } else if (soilmoisturepercent > 0 && soilmoisturepercent < 100) {
    return (soilmoisturepercent);
  }
  delay(250);
}

float lecturaTEMPERATURA(){
  temp= dht.readTemperature();
  return (temp);
  delay(250); //Delay
}

void loop() {
  char humChar[32]; //Almacena la lectura en un char array para conversion
  char tempChar[32]; //Almacena la lectura en un char array para conversion
  float humFloat = lecturaHUMEDAD(); //Almacena en un float el return de la funcion
  float tempFloat = lecturaTEMPERATURA(); //Almacena en un float el return de la funcion
  dtostrf(humFloat, 4, 2, humChar); //Convierte un float(phFloat) de minimo 4 de longitud con 2 decimales hacia un char(phChar)
  dtostrf(tempFloat, 4, 2, tempChar); //Convierte un float(phFloat) de minimo 4 de longitud con 2 decimales hacia un char(phChar)
  sprintf(nodo2, "{\"id\":\"2\",\"datetime\":\"2022-06-13 22:10:10\",\"humedad\":\"%s\",\"temperatura\":\"%s\"}\n", humChar, tempChar);
  Serial.write(nodo2);
  delay(12000);
}

```

Figura 32. Código de Arduino Nodo Sensor 2 lectura de datos.

En el siguiente paso se pasarían los datos hacia la placa Open Mote B la cual recolectará esos datos y realizará el enrutamiento hacia el Borde Router, inicialmente vemos la recepción de los datos a través de Comunicación Serial, véase figura 33.

```

//el thread del emcute esta arriba incluso, aqui va bien este
static void *readSerial_thread(void *arg)
{
  (void)arg;
  //SERIAL RX
  int received = 0;

  while (1) {
    char c;
    stdio_read(&c, 1);

    test_buffer[received] = c;

    if (received < TEST_BUF_SIZE - 1) {
      received++;
    }

    if (c == '\n' || c == '\r') {
      strncpy(json_global, test_buffer, received);
      json_global[received] = '\0';

      //Reenvio a arduino
      stdio_write(test_buffer, received); //SI DA PROBLEMAS CON ARDUINO BORRAR
      stdio_write("\n", 1); //SI DA PROBLEMAS CON ARDUINO BORRAR

      LED2_TOGGLE;
      received = 0;
    }
  }

  return NULL; /* nunca debe ser alcanzado */
}

```

Figura 33. Código de la Recepción de datos por Comunicación Serial.

En este mismo apartado se procederá a configurar la capa física para la modulación de las señales como se mencionó el apartado del Marco Teórico IEEE 802.15.4 trabaja en diferentes modulaciones en este caso se usará la modulación de OFDM, podemos ver la tabla a continuación basándonos en la frecuencia central y los canales, véase tabla 27.

Tabla 28. Tabla de Canales.

<b>OFDM</b>				
<b>Canales</b>	<b>OFDM Option 1</b>	<b>OFDM Option 2</b>	<b>OFDM Option 3</b>	<b>OFDM Option 4</b>
0	863,625 MHz	863,425 MHz	863,225 MHz	863,125 MHz
1	864,825 MHz	864,225 MHz	863,625 MHz	863,325 MHz
2	866,025 MHz	865,025 MHz	864,025 MHz	863,525 MHz
3	867,225 MHz	865,825 MHz	864,425 MHz	863,725 MHz
4	868,425 MHz	866,625 MHz	864,825 MHz	863,925 MHz
5	869,625 MHz	867,425 MHz	865,225 MHz	864,125 MHz
6		868,225 MHz	865,625 MHz	864,325 MHz
7		869,025 MHz	866,025 MHz	864,525 MHz
8		869,825 MHz	866,425 MHz	864,725 MHz
9			866,825 MHz	864,925 MHz
10			867,225 MHz	865,125 MHz
11			867,625 MHz	865,325 MHz
12			868,025 MHz	865,525 MHz
13			868,425 MHz	865,725 MHz
14			868,825 MHz	865,925 MHz
15			869,225 MHz	866,125 MHz
16			869,625 MHz	866,325 MHz

Fuente: Autora basándose en (IEEE, 2019).

Un símbolo OFDM es la combinación de las subportadoras, cada una de las cuales transporta una parte de la información que se va a transmitir. La combinación de las subportadoras equivale a tener una modulación de señal de alto orden, lo que hace posible lograr altas velocidades de datos. El estándar describe 4 formas de agrupar subportadoras para formar un símbolo OFDM; se denominan “opciones”, y están numeradas del 1 al 4. La Tabla 29 muestra el número de subportadoras para cada opción, así como el ancho de banda que ocupa.

Tabla 29. Subportadoras que se utilizan en un canal.

<b>OFDM</b>				
	<b>OFDM Option 1</b>	<b>OFDM Option 2</b>	<b>OFDM Option 3</b>	<b>OFDM Option 4</b>
# Sub-carriers	104	52	26	14
(data/pilot)	96/8	48/4	24/2	12/2
Ancho de Canal (kHz)	1094	552	281	156

Espacio entre canales (kHz)	1200	800	400	200
# Canales	5	8	17	34

*Fuente:* Autora basándose en (IEEE, 2019).

Debido a que la modulación de la subportadora se especifica por el Esquema de modulación y codificación (MCS) la tasa de datos será dado por esa combinación, en el Esquema MCS0 se tiene una velocidad de datos de 50 [kbps] en la opción 2; la misma que tiene 52 subportadoras que hacen posible una comunicación a altas velocidades de datos, lo cual es suficiente para la comunicación en IoT. Finalmente se escoge un canal de forma aleatoria para realizar las pruebas del prototipo, en dicho caso se ha escogido el canal número 1 en el cual la frecuencia central será 864.225 MHz, dentro del código de la placa Open Mote B se configurará todas las opciones de modulación para MR-OFDM en el esquema 0, con el canal 1 en la opción 2. Véase la figura 34.

```
//Configuraciones automaticas
puts("Configurando Nodo\n");
sh("ifconfig");
sh("ifconfig 6 add 2001:db8::4/48");
sh("ifconfig 6 set phy mr-ofdm");
sh("ifconfig 6 set power 13");
sh("ifconfig 6 set channel 1");
sh("ifconfig 6 set option 2");
sh("ifconfig 6 set scheme 0");
sh("rpl init 6");
sh("ifconfig");
puts("Configuraciones terminadas\n");
puts("Empezando envio de datos...\n");
```

*Figura 34. Sentencias de Configuración de Modulación en Open Mote B.*

### 3.10.2 Capa de Red

En esta infraestructura de redes inalámbricas, se tiene dos funciones: Función de Red y Transporte; la primera de ellas proporciona funciones de control, como es el acceso y el control de recursos, para la de Transporte se tiene el responsable de proporcionar la conectividad para el transporte del servicio IoT y gestión de control. En este apartado podemos ver el protocolo de RPL por vector de distancia para LLN que usa IPv6, de tal manera que no hay ciclos presentes. En los Nodos Sensores, se enrutará hacia el Borde Router y posteriormente se publicarán en el Broker MQTT-SN.

```

//Configuraciones automaticas
puts("Configurando Nodo\n");
sh("ifconfig");
sh("ifconfig 6 add 2001:db8::4/48");
sh("ifconfig 6 set phy mr-ofdm");
sh("ifconfig 6 set power 13");
sh("ifconfig 6 set channel 1");
sh("ifconfig 6 set option 2");
sh("ifconfig 6 set scheme 0");
sh("rpl init 6");
sh("ifconfig");
puts("Configuraciones terminadas\n");
puts("Empezando envio de datos...\n");

```

Figura 35. Configuración del Enrutamiento RPL para los Nodos.

```

//MQTT THREAD
static void *sendMqtt_thread(void *arg)
{
    (void)arg;
    puts("Inicio cmd_auto");
    char topic[32];
    //sprintf(topic, "sensor/station%d", 2);
    sprintf(topic, "riot_device");
    while(1){ //Me da a que da problemas un loop junto con el serial
        // intenta conectarse a la puerta de enlace
        if (con("2001:db8::1", 1885)) {
            LED0_TOGGLE;
            continue;
        }
        pub(topic, json_global, 0);
        discon();
        LED1_TOGGLE;

        xtimer_sleep(15);
    }

    return NULL; /* nunca debe ser alcanzado */
}

```

Figura 36. Código del enrutamiento hacia el Broker MQTT-SN.

Dentro de la máquina intermediaria que participará en el envío de los datos hacia el Cloud se necesitará realizar el enrutamiento de la interfaz tap(virtual) creada para Borde Router; inicialmente se añade la IPv6 a la interfaz virtual, lo siguiente es borrar todas las rutas para la red 2001:db8::/64, añadimos una nueva ruta la cual nos dirige hacia la interfaz SubGhz del Open Mote B a través de la tap5 y borramos cualquier ruta por defecto que se haya creado y enrutamos la red 2001:db8::/64 a través de 2001:db8::2 por la tap5. Fig 37.

```

new release 22.01.1-B13 available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Nov 28 13:22:15 2022 from 192.168.0.102
riot@riot:~$ sudo ip a a 2001:db8::1/64 dev tap5
[sudo] contraseña para riot:
riot@riot:~$ sudo ip r d 2001:db8::/64 dev tap5
riot@riot:~$ sudo ip r a 2001:db8::2 dev tap5
riot@riot:~$ sudo ip r a 2001:db8::/64 via 2001:db8::2 dev tap5
RTNETLINK answers: File exists
riot@riot:~$ sudo ip r d 2001:db8::/64 dev tap5
riot@riot:~$ sudo ip r a 2001:db8::/64 via 2001:db8::2 dev tap5
riot@riot:~$ ping 2001:db8::3
PING 2001:db8::3(2001:db8::3) 56 data bytes
64 bytes from 2001:db8::3: icmp_seq=1 ttl=64 time=64.7 ms
64 bytes from 2001:db8::3: icmp_seq=2 ttl=64 time=38.4 ms
^C

```

Figura 37. Configuración del enrutamiento de la tap.

### 3.10.3 Capa de Servicio / Procesamiento de Datos

Se le denomina computación en la nube, tiene dos grupos: función de soporte general; que son las funciones de procesamiento de datos, y la función de soporte específico; que son las funciones que se adaptan a los requisitos de las aplicaciones con el fin de proporcionar diferentes funciones de apoyo. En este apartado podemos identificar la publicación hacia el Broker MQTT que será el que enviará hacia el Cloud para su procesamiento y almacenamiento.

```

static int pub(char* topic, char* data, int qos){
    emcute_topic_t t;
    unsigned flags = EMCUTE_QOS_0;
    switch (qos) {
        case 1:
            flags |= EMCUTE_QOS_1;
            break;
        case 2:
            flags |= EMCUTE_QOS_2;
            break;
        default:
            flags |= EMCUTE_QOS_0;
            break;
    }
    t.name = topic;

    if (emcute_reg(&t) != EMCUTE_OK) {
        puts("error: unable to obtain topic ID");
        return 1;
    }
    if (emcute_pub(&t, data, strlen(data), flags) != EMCUTE_OK) {
        printf("error: unable to publish data to topic '%s [%i]'\n",
            t.name, (int)t.id);
        return 1;
    }
    printf("Publicado %s en el topic %s\n", data, topic);
    LED3_TOGGLE;
    return 0;
}

```

Figura 38. Código del proceso de publicación en el Broker MQTT SN.

Al momento de recibir el borde router los datos enviará un ACK hacia los nodos y seguido de esto se enviará al Broker MQTT SN; el cual está montado en Mosquitto RSMB. Broker irá recibiendo cada uno de los paquetes y enviándolos al Puente Transparente, para ello se requiere abrir los puertos para los protocolos, figura 39.

```
# add some debug output
trace_output protocol
# listen for MQTT-SN traffic on UDP port 1885
listener 1885 INADDR_ANY mqtt
ipv6 true

# listen to MQTT connections on tcp port 1886
listener 1886 INADDR_ANY
ipv6 true
```

Figura 39. Configuración de los puertos abiertos en el Broker.

Al mismo tiempo el puente transparente; quien tendrá las credenciales de acceso a la Nube, se registrará en el Broker MQTT-SN para recibir el JSON que será publicado en Google Cloud, fig. 40.

```
# Parameters to authenticate the device on google
project_id = 'trabajodegrado-369023'
cloud_region = 'us-central1'
registry_id = 'my-registry'
device_id = 'my-device'
private_key_file = '/home/riot/rsa_private.pem'
algorithm = 'RS256'
ca_certs = '/home/riot/roots.pem'
mqtt_bridge_hostname = 'mqtt.googleapis.com'
mqtt_bridge_port = '8883'
message_type = 'event'

sub_topic = 'events' if message_type == 'event' else 'state'

mqtt_topic = '/devices/{}/{}'.format(device_id, sub_topic)
print ("mqtt_topic: ", mqtt_topic)

client = GoogleMQTTClient.get_client(
    project_id, cloud_region, registry_id,
    device_id, private_key_file, algorithm,
    ca_certs, mqtt_bridge_hostname, mqtt_bridge_port)
```

Figura 40. Códigos para añadir credenciales para el Cloud.

Ahora que hemos configurado nuestro cliente, solo tenemos que escuchar a nuestro broker RSMB y asegurarnos de que cada mensaje que llegue se reenvíe a Google.

```

def messageArrived(self, topicName, payload, qos, retained, msgid):
    client.publish(mqtt_topic, payload, qos=0)
    client.loop()

    print("default publishArrived", topicName, payload, qos, retained, msgid)
    return True

if __name__ == "__main__":

    aclient = Client("GoogleBridge", port=1885)
    aclient.registerCallback(Callback())
    aclient.connect()
    rc, topic1 = aclient.subscribe("riot_device")
    while(True):{    #listening for messages

```

Figura 41. Código de espera de un Mensaje Recibido.

Finalmente se recibirá el ACK respectivo por la llegada de cada uno de los paquetes tanto de registro como conexión y publicación, paquetes que se especificarán más adelante, véase figura 42.

```

20221128 133345.043 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133345.043 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133345.043 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133345.107 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4681 topicid: 0 topicname: riot_devic
20221128 133345.107 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4681 topicid: 1 returncode: 0 (0)
20221128 133345.187 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133345.187 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133345.203 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133345.203 CWNAN0038I Disconnection request received from client NODO2
20221128 133345.203 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)
20221128 133400.258 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133400.258 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133400.258 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133400.322 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4682 topicid: 0 topicname: riot_devic
20221128 133400.322 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4682 topicid: 1 returncode: 0 (0)
20221128 133400.402 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133400.402 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133400.418 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133400.418 CWNAN0038I Disconnection request received from client NODO2
20221128 133400.418 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)

```

Figura 42. Log de Registro, Conexión y Publicación del Broker RSMB.

A continuación, al llegar los datos a la Nube para ser procesados se necesitará el análisis de cada uno de ellos en una Web Page para ello se deberá almacenar en una DaaS (Database as a Service), en dicho caso se ha escogido MongoDB por su conexión directa hacia Google Cloud y la Página Web, en la cual se deberá conectar como se observa en la figura 43.

```

// Map global promise - get rid of warning
mongoose.Promise = global.Promise;
//connect to mongoose
const uri = "mongodb+srv://jeyepz: [REDACTED]@tdg.yfkyoeg.mongodb.net/TdG";
mongoose.connect(uri, {
  useNewUrlParser: true
})
.then(() => {
  console.log('MongoDb Connected..');
})
.catch(err => console.log(err));

```

Figura 43. Código para la conexión a la Base de Datos.

### 3.10.4 Capa de Aplicaciones

En la capa de Aplicación se pondrá en usa la gran cantidad de información adquirida y creada a partir de las capas anteriores y es donde se verá reflejada la mayor cantidad de programación vista previamente, para ello se necesitan las credenciales de Google Cloud para que se conecte con la IaaS, seguido de la conexión a la Base de Datos.

```

}
// Imports the Google Cloud client library
io.on('connection', function(socket){
  var values = Values.find(
    { date: { $gt: parseInt(Date.now()/1000) - 3600 } }
  ).sort({date:-1}).then(values => {
    io.emit("lastvalues", values);
  });
  const subscriptionName1 = 'projects/trabajodegrado-369023/subscriptions/my_subscription';
  // Creates a client; cache this for further use
  const pubSubClient = new PubSub();
  function listenForMessages() {
    // References an existing subscription
    const subscription1 = pubSubClient.subscription(subscriptionName1);
    // Create an event handler to handle messages

```

Figura 44. Configuración credenciales de suscripción a Google Cloud en Página Web.

Una vez se conecta adecuadamente al Google Cloud se leerán los últimos mensajes que han llegado al Cloud y se separan por etiqueta adjuntando la hora actual para posteriormente visualizarles con handlebars; véase figura 45, donde de la misma forma se realizará un cálculo de promedio de los últimos 10 valores y se ubicarán en una línea de tiempo que se irá dibujando según la hora, véase figura 46.



```

let messageCount = 0;
const messageHandler = message => {
  console.log('Received message ${message.id}:');
  console.log('\tData: ' + message.data);
  console.log('\tAttributes: ${message.attributes}');
  messageCount += 1;
  var payload = JSON.parse(message.data);
  const newValue = {
    id: payload.id,
    datetime: payload.datetime,
    temperatura: payload.temperatura,
    humedad: payload.humedad,
    ph: payload.ph
  };
  if (payload.ph == undefined && payload.temperatura != undefined){
    new Values(newValue).save();
    console.log(payload.id);
    console.log(payload.datetime);
    console.log(payload.temperatura);
    console.log(payload.humedad);
    // TEMPERATURE
    io.emit("temperatura", (payload.id + ";" + payload.temperatura + ";" + payload.datetime).toString());
    // HUMIDITY
    io.emit("humedad", (payload.id + ";" + payload.humedad + ";" + payload.datetime).toString());
    console.log("HolaNodo2");
  } else {
    /*const newValue = {
      id: payload.id,
      datetime: payload.datetime,
      ph: payload.ph
    };*/
    new Values(newValue).save();
    console.log(payload.id);
    console.log(payload.datetime);
    console.log(payload.ph);
    io.emit("ph", (payload.id + ";" + payload.ph + ";" + payload.datetime).toString());
    console.log("HolaNodo1");
  }
  // "Ack" (acknowledge receipt of) the message
  message.ack();
};
// Listen for new messages until timeout is hit
subscription1.on('message', messageHandler);

```

Figura 45. Código de la recepción y separación de etiquetas en la Página Web.

```

/// Devices Management
/***** ANALYSIS PLOT *****/
var ctx = document.getElementById('temp-chart').getContext('2d');
var charttemp = new Chart(ctx, {
  // The type of chart we want to create
  type: 'line',
  // The data for our dataset
  data: {
    labels: [],
    datasets: [{fill:false, label: 'Temperatura', backgroundColor: '#58ACFA', borderColor: '#58ACFA',
      data: []
    }],
    sum: 0,
  },
  // Configuration options go here
  options: {
    responsive: true,
    title: {
      display: true,
      text: 'Temperatura (°C)'
    },
    scales: {
      yAxes: [{
        ticks: { suggestedMin: -50, suggestedMax: 50 }
      }]
    }
  }
});

```

Figura 46. Código para la visualización de Datos con Handlebar.

Para el cálculo del último valor recibido se usará Web Socket el cual nos permitirá realizar la línea de tiempo y el promedio de los últimos valores.

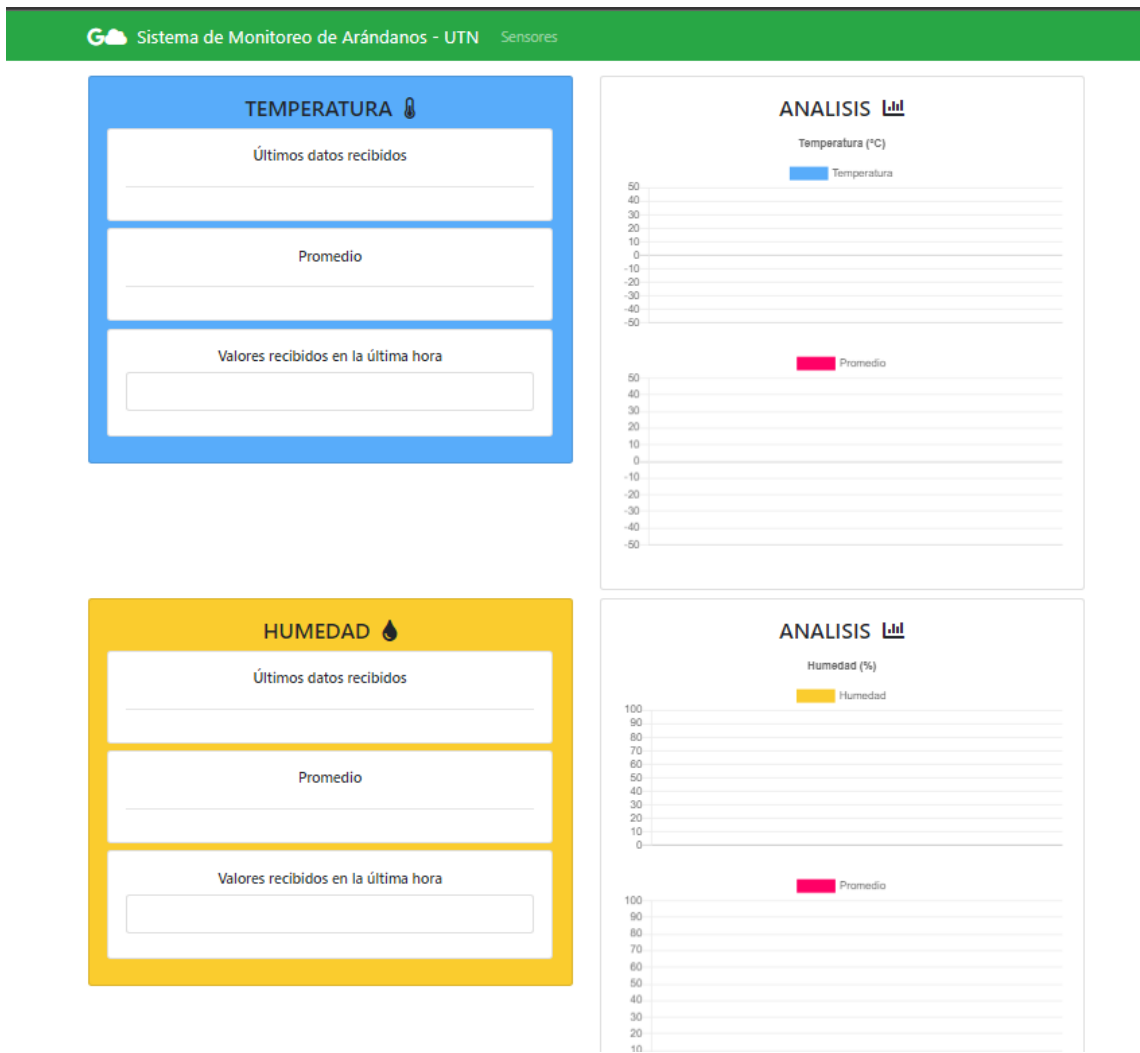
```

socket.on("lastvalues", function(values){
  if(charttemp.data.labels.length == 0 && values.length > 0){
    values.reverse()
    values.forEach( value =>{
      charttemp.data.datasets[0].data.push(value.temperatura)
      console.log(value.temperatura);
      charttemp.data.sum = parseInt(charttemp.data.sum) + parseInt(value.temperatura)
      charttemp.data.labels.push('');
      charttemp.update();
      charttempavg.data.datasets[0].data.push(parseInt(charttemp.data.sum)/charttemp.data.datasets[0].data.length)
      charttempavg.data.labels.push('');
      charttempavg.update();
      charthum.data.datasets[0].data.push(value.humedad)
      console.log(value.humedad);
      console.log(value.ph);
      charthum.data.sum = parseInt(charthum.data.sum) + parseInt(value.humedad)
      charthum.data.labels.push('');
      charthum.update();
      charthumavg.data.datasets[0].data.push(parseInt(charthum.data.sum)/charthum.data.datasets[0].data.length)
      charthumavg.data.labels.push('');
      charthumavg.update();
      chartrain.data.datasets[0].data.push(value.ph)
      chartrain.data.sum = parseInt(chartrain.data.sum) + parseInt(value.ph)
      chartrain.data.labels.push('');
      chartrain.update();
      chartrainavg.data.datasets[0].data.push(parseInt(chartrain.data.sum)/chartrain.data.datasets[0].data.length)
      chartrainavg.data.labels.push('');
      chartrainavg.update();
    });
    $('#avg-temp').html('<h4 align="center">' + (parseInt(charttemp.data.sum)/charttemp.data.datasets[0].data.length).toFixed(2) + ' °C</h4>');
    $('#avg-hum').html('<h4 align="center">' + (parseInt(charthum.data.sum)/charthum.data.datasets[0].data.length).toFixed(2) + ' %</h4>');
    $('#avg-rain').html('<h4 align="center">' + (parseInt(chartrain.data.sum)/chartrain.data.datasets[0].data.length).toFixed(2) + ' </h4>');
  }
});

```

Figura 47. Configuración de Web Socket para actualización de tiempos.

Los datos se deberían visualizar de la siguiente forma (fig. 48) separado por tipo de Sensor, el promedio, los últimos 10 valores y la gráfica de análisis en el dominio del tiempo la cual irá graficando según el dato recibido.



*Figura 48. Ejemplo de la Interfaz de Usuario.*

## Capítulo 4: Implementación y Pruebas

En este capítulo se va a detallar el proceso y todos los pasos a seguir para implementar todo el Hardware y Software y realizar las pruebas en el Invernadero para observar los datos en tiempo real en la Nube verificando el funcionamiento del prototipo.

### 4.1 Implementación en el Invernadero

Una vez se tiene listo todo el prototipo, nos dirigimos hacia la dirección del Invernadero en el Barrio San Pedro de la Esperanza; en el cual se encuentran todas las macetas de Arándanos, para pruebas del prototipo se aplicará en una maceta el Nodo de Sensor 2 el cual censará la temperatura del lugar y la humedad del suelo; por otro lado, en el agua de riego se aplicará el Nodo de Sensor 1 para lectura del PH del agua.



Figura 49. Ubicación de Sensores en la maceta.



*Figura 50. Nodo de Sensor 1 ubicado en la maceta.*



*Figura 51. Nodo de Sensores 2 ubicado en la maceta.*

Una vez ubicado los sensores y encendidos se procede a configurar dentro de la Máquina Virtual las rutas y a encender el Broker RSMB y el Puente Transparente para poder enviar los datos hacia la Nube; en la figura 52 (a) se observa el Broker MQTT activado recibiendo los paquetes de registro de los Nodos; en la figura 52 (b) se ve el puente transparente el cual está enviando el paquete JSON a la Nube; en la figura 52

- (c) se observa la configuración del Borde Router con su modulación y en la figura 52 (d) podremos observar las rutas creadas para el prototipo.

The figure consists of four terminal windows arranged in a 2x2 grid, each showing different stages of the IoT network setup:

- (a) Broker RSMB:** Shows MQTT-S messages including disconnections and connections from clients NODO2 and CWNAN00001. It also displays a REGISTER message for a topic named 'riot\_device'.
- (b) Puente Transparente:** Shows a stream of MQTT-S PUBLISH messages from a Google Cloud client, containing sensor data like humidity and temperature.
- (c) Borde Router:** Shows the configuration of a RIOT border router, including setting up interfaces, power levels, and modulation schemes.
- (d) Enrutamiento tap5:** Shows the results of ping tests to various IP addresses (2001:db8::3, 4, 5) on the tap5 interface, indicating successful connectivity.

Figura 52. (a) Broker RSMB, (b) Puente Transparente, (c) Borde Router, (d) Enrutamiento tap5.

Como todos los procesos se han ejecutado de forma exitosa y vemos que los datos se están publicando se puede revisar en la plataforma de Google Cloud, el cual nos muestra todos los datos publicándose, la diferencia es que se mostrarán algunos y otros no, debido a la rapidez de llegada de los mensajes por eso se necesitará un Web Page.

MENSAJES	MÉTRICAS	DETALLES
21 nov 2022 16:52:43	deviceId	["deviceId": "my-device", "temperature": -35, "humidity": 3, "wind_direction": 167, "date": "2022-11-21T16:52:43.000Z"]
21 nov 2022 16:52:58	deviceId	["deviceId": "my-device", "temperature": -16, "humidity": 47, "wind_direction": 209, "date": "2022-11-21T16:52:58.000Z"]
21 nov 2022 16:53:13	deviceId	["deviceId": "my-device", "temperature": 50, "humidity": 2, "wind_direction": 144, "date": "2022-11-21T16:53:13.000Z"]
21 nov 2022 16:53:28	deviceId	["deviceId": "my-device", "temperature": 28, "humidity": 77, "wind_direction": 290, "date": "2022-11-21T16:53:28.000Z"]
21 nov 2022 16:53:43	deviceId	["deviceId": "my-device", "temperature": -3, "humidity": 86, "wind_direction": 121, "date": "2022-11-21T16:53:43.000Z"]
21 nov 2022 16:53:58	deviceId	["deviceId": "my-device", "temperature": -4, "humidity": 93, "wind_direction": 160, "date": "2022-11-21T16:53:58.000Z"]
21 nov 2022 16:54:13	deviceId	["deviceId": "my-device", "temperature": -34, "humidity": 54, "wind_direction": 126, "date": "2022-11-21T16:54:13.000Z"]
21 nov 2022 16:54:28	deviceId	["deviceId": "my-device", "temperature": 36, "humidity": 19, "wind_direction": 298, "date": "2022-11-21T16:54:28.000Z"]
21 nov 2022 16:54:43	deviceId	["deviceId": "my-device", "temperature": -30, "humidity": 60, "wind_direction": 314, "date": "2022-11-21T16:54:43.000Z"]
21 nov 2022 16:54:58	deviceId	["deviceId": "my-device", "temperature": -21, "humidity": 74, "wind_direction": 156, "date": "2022-11-21T16:54:58.000Z"]
21 nov 2022 16:55:13	deviceId	["deviceId": "my-device", "temperature": 17, "humidity": 89, "wind_direction": 47, "date": "2022-11-21T16:55:13.000Z"]
21 nov 2022 16:55:28	deviceId	["deviceId": "my-device", "temperature": -24, "humidity": 55, "wind_direction": 96, "date": "2022-11-21T16:55:28.000Z"]
21 nov 2022 16:55:43	deviceId	["deviceId": "my-device", "temperature": 47, "humidity": 58, "wind_direction": 172, "date": "2022-11-21T16:55:43.000Z"]
21 nov 2022 16:55:58	deviceId	["deviceId": "my-device", "temperature": 34, "humidity": 97, "wind_direction": 133, "date": "2022-11-21T16:55:58.000Z"]
21 nov 2022 16:56:13	deviceId	["deviceId": "my-device", "temperature": 37, "humidity": 65, "wind_direction": 101, "date": "2022-11-21T16:56:13.000Z"]
21 nov 2022 16:56:28	deviceId	["deviceId": "my-device", "temperature": -7, "humidity": 58, "wind_direction": 270, "date": "2022-11-21T16:56:28.000Z"]

Figura 53. Paquetes JSON recibidos en la Nube.

Como se mencionó para poder ver los paquetes en tiempo real se los enviará a una página web la cual necesitará de una base de datos para almacenar dichos datos, es por ello que se usa MongoDB donde se puede observar la tabla creada, fig 54, y se puede observar el dato reflejado en la Página Web junto con cada uno de los análisis.

UTN > PROJECT 0

## Database Deployments

Find a database deployment...

+ Create

TdG Connect View Monitoring Browse Collections ...

FREE SHARED

Enhance Your Experience  
For production throughput and richer metrics, upgrade to a dedicated cluster now!  
Upgrade

R: 0  
W: 0  
Last 2 hours  
100.0%

Connections: 1.0  
Last 2 hours  
1.0

In: 15.4 B/s  
Out: 106.6 B/s  
Last 2 hours  
111.0 B/s

Data Size: 212.4 KB  
Last 14 days  
512.0 MB

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SEARCH
5.0.14	GCP / Iowa (us-central)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	arandanos	Create Index

Figura 54. Base de Datos en MongoDB.

## 4.2 Pruebas de Sensores

A continuación, se presentarán las pruebas de los datos censados dentro del Invernadero, demostrando los rangos apropiados para el crecimiento de los Arándanos.

### 4.2.1 Lectura de Sensores

Una vez se tienen listos todos los datos y almacenados podemos observar en la Página Web, donde tras varias horas se pudo observar que los rangos se mantenían dentro de lo apropiado, véase figura 55. Y al existir una varias se vería reflejada en la misma, al igual que una variación en la gráfica de análisis, véase la fig 56.

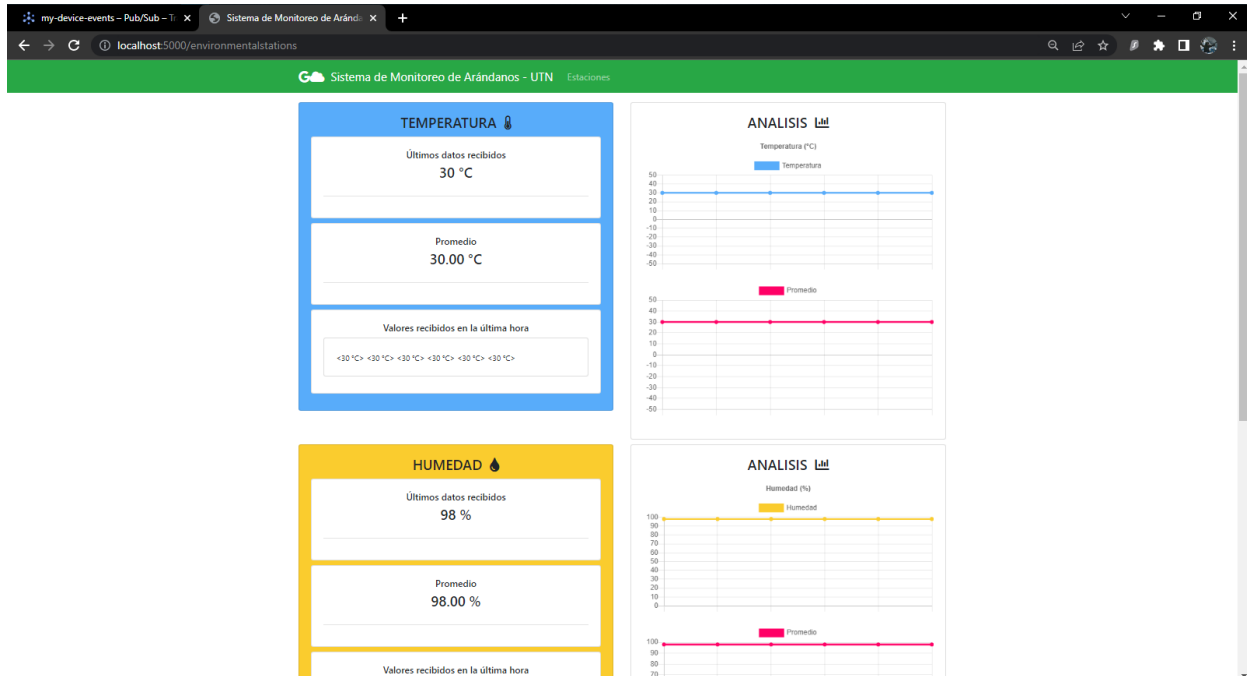


Figura 55. Visualización de datos de sensores dentro de los rangos adecuados.

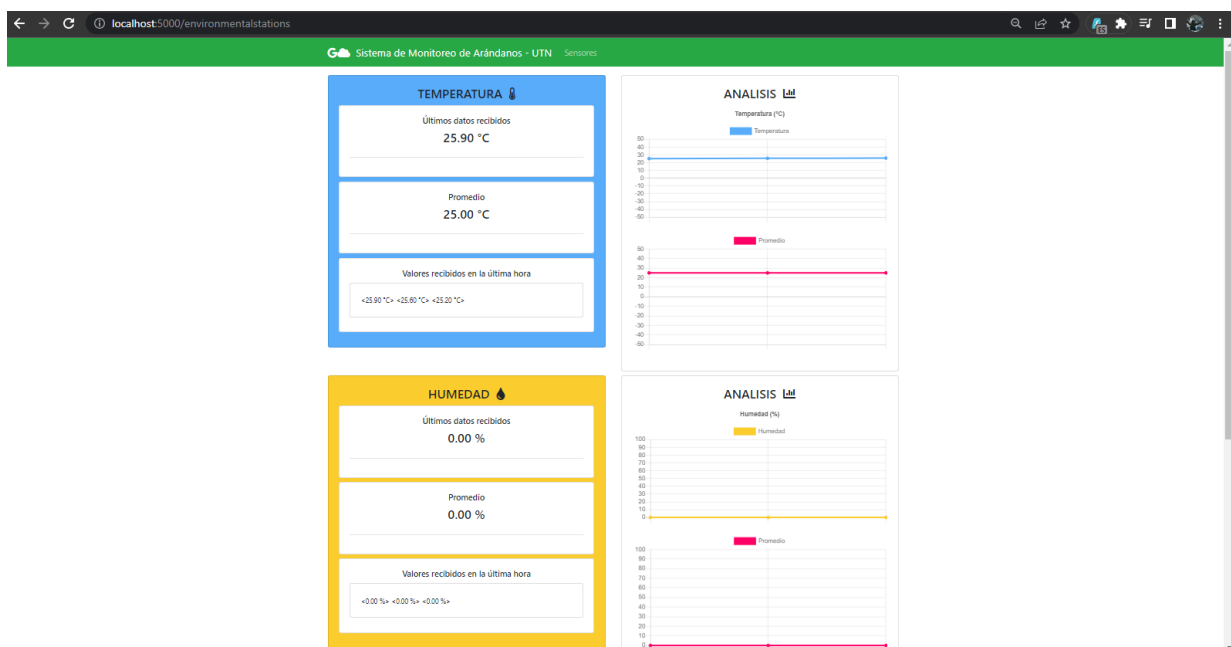


Figura 56. Visualización de datos de sensores fuera de rangos adecuados.



## 4.2.2 Análisis de Paquetes de Wireshark

Mientras los equipos se encuentran conectados se escaneó el medio buscando los paquetes que nos demuestren el envío y llegaba de los paquetes para ello se comparará con los paquetes enviados desde el Broker RSMB donde podemos ver cada uno de ellos, como se puede ver inicialmente se requiere una conexión con el Broker el cual recibe el Nodo 2 (para este caso puntual) y envía un ACK de confirmación, fig 57. Y en la figura 58 veremos el paquete de request el cual anuncia la llegada de paquetes del NODO 2 con los puertos que está usando y seguido de este paquete se observa el ACK.

```

20221128 133345.043 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133345.043 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133345.043 CWNAN00001 Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133345.107 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4681 topicid: 0 topicname: riot_devic
20221128 133345.107 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4681 topicid: 1 returncode: 0 (0)
20221128 133345.187 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133345.187 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133345.203 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133345.203 CWNAN0038I Disconnection request received from client NODO2
20221128 133345.203 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)
20221128 133400.258 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133400.258 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133400.258 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133400.322 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4682 topicid: 0 topicname: riot_devic
20221128 133400.322 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4682 topicid: 1 returncode: 0 (0)
20221128 133400.402 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133400.402 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133400.418 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133400.418 CWNAN0038I Disconnection request received from client NODO2
20221128 133400.418 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)

```

Figura 57. Log de paquetes de Conexión.

No.	Time	Source	Destination	Protocol	Length	Info
10	5.023666015	2001:db8::5	2001:db8::1	UDP	73	1883 → 1885 Len=11
11	5.023765125	2001:db8::1	2001:db8::5	UDP	65	1885 → 1883 Len=3
12	5.087693872	2001:db8::5	2001:db8::1	UDP	79	1883 → 1885 Len=17
13	5.087784031	2001:db8::1	2001:db8::5	UDP	69	1885 → 1883 Len=7
14	5.167641350	2001:db8::5	2001:db8::1	UDP	152	1883 → 1885 Len=90
15	5.183692671	2001:db8::5	2001:db8::1	UDP	64	1883 → 1885 Len=2
16	5.183792041	2001:db8::1	2001:db8::5	UDP	64	1885 → 1883 Len=2
17	15.215014064	2001:db8::4	2001:db8::1	UDP	73	1883 → 1885 Len=11
18	15.215116964	2001:db8::1	2001:db8::4	UDP	65	1885 → 1883 Len=3
19	15.279013820	2001:db8::4	2001:db8::1	UDP	79	1883 → 1885 Len=17
20	15.279124630	2001:db8::1	2001:db8::4	UDP	69	1885 → 1883 Len=7
21	15.343001547	2001:db8::4	2001:db8::1	UDP	125	1883 → 1885 Len=63

Frame 10: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface tap5, id 0

Ethernet II, Src: ae:36:db:9a:af:46 (ae:36:db:9a:af:46), Dst: 06:c0:cc:60:c7:27 (06:c0:cc:60:c7:27)

Internet Protocol Version 6, Src: 2001:db8::5, Dst: 2001:db8::1

0110 .... = Version: 6

.... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)

.... 0000 0000 0000 0000 = Flow Label: 0x000000

Payload Length: 19

Next Header: UDP (17)

Hop Limit: 63

Source Address: 2001:db8::5

Destination Address: 2001:db8::1

User Datagram Protocol, Src Port: 1883, Dst Port: 1885

Data (11 bytes)

Data: 0b040101684e4f444f32

[Length: 11]

0000 06 c0 cc 60 c7 27 ae 36 db 9a af 46 86 dd 60 00 .....6 ...F...

0010 00 00 00 13 11 3f 20 01 0d b8 00 00 00 00 00 .....? ...

0020 00 00 00 00 00 05 20 01 0d b8 00 00 00 00 00 .....

0030 00 00 00 00 00 01 07 5b 07 5d 00 13 c0 8c 0b 04 .....[ ]... ..

0040 04 01 01 68 4e 4f 44 4f 32 .....hNODO 2

Figura 58. Visualización de paquete de Conexión en Wireshark.

Una vez se conecta va a realizar el registro anunciando el tema al cual se va a conectar dentro de la Nube, de la misma forma enviará un ACK de respuesta al registro.

En la figura 59 se observa la solicitud de registro y seguido de ella el ACK confirmando el registro, así mismo se observa dentro de los paquetes de Wireshark, fig 60.

```

20221128 133345.043 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133345.043 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133345.043 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133345.107 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4681 topicid: 0 topicname: riot_devic
20221128 133345.107 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4681 topicid: 1 returncode: 0 (0)
20221128 133345.187 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133345.187 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133345.203 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133345.203 CWNAN0038I Disconnection request received from client NODO2
20221128 133345.203 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)
20221128 133400.258 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133400.258 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133400.258 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133400.322 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4682 topicid: 0 topicname: riot_devic
20221128 133400.322 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4682 topicid: 1 returncode: 0 (0)
20221128 133400.402 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133400.402 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133400.418 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133400.418 CWNAN0038I Disconnection request received from client NODO2
20221128 133400.418 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)

```

Figura 59. Log de paquetes de Registro.

No.	Time	Source	Destination	Protocol	Length	Info
10	5.023666015	2001:db8::5	2001:db8::1	UDP	73	1883 → 1885 Len=11
11	5.023765125	2001:db8::1	2001:db8::5	UDP	65	1885 → 1883 Len=3
12	5.087693872	2001:db8::5	2001:db8::1	UDP	79	1883 → 1885 Len=17
13	5.087784031	2001:db8::1	2001:db8::5	UDP	69	1885 → 1883 Len=7
14	5.167641350	2001:db8::5	2001:db8::1	UDP	152	1883 → 1885 Len=90
15	5.183692671	2001:db8::5	2001:db8::1	UDP	64	1883 → 1885 Len=2
16	5.183792041	2001:db8::1	2001:db8::5	UDP	64	1885 → 1883 Len=2
17	15.215014064	2001:db8::4	2001:db8::1	UDP	73	1883 → 1885 Len=11
18	15.215116964	2001:db8::1	2001:db8::4	UDP	65	1885 → 1883 Len=3
19	15.279013820	2001:db8::4	2001:db8::1	UDP	79	1883 → 1885 Len=17
20	15.279124630	2001:db8::1	2001:db8::4	UDP	69	1885 → 1883 Len=7
21	15.343001547	2001:db8::4	2001:db8::1	UDP	125	1883 → 1885 Len=63

> Frame 12: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface tap5, id 0 > Ethernet II, Src: ae:36:db:9a:af:46 (ae:36:db:9a:af:46), Dst: 06:c0:cc:60:c7:27 (06:c0:cc:60:c7:27) > Internet Protocol Version 6, Src: 2001:db8::5, Dst: 2001:db8::1 0110 .... = Version: 6 ... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT) ... 0000 0000 0000 0000 = Flow Label: 0x000000 Payload Length: 25 Next Header: UDP (17) Hop Limit: 63 Source Address: 2001:db8::5 Destination Address: 2001:db8::1 > User Datagram Protocol, Src Port: 1883, Dst Port: 1885 > Data (17 bytes) Data: 110a0000123872696f745f646576696365 [Length: 17]		0000 06 c0 cc 60 c7 27 ae 36 db 9a af 46 86 dd 60 00 .....6...F... 0010 00 00 00 19 11 3f 20 01 0d b8 00 00 00 00 00 .....? 0020 00 00 00 00 00 05 20 01 0d b8 00 00 00 00 00 ..... 0030 00 00 00 00 00 01 07 5b 07 5d 00 19 fd 2d 11 0a .....[ ]..... 0040 00 00 12 38 72 69 6f 74 5f 64 65 76 69 63 65 .....8riot_device
---	--	--

Figura 60. Visualización de captura de Registro en Wireshark.

Siguiendo con el proceso ahora se requerirá publicar, el NODO 2 enviará el paquete JSON el cual pasará por el Broker MQTT y seguido de ello se publicará en la Nube como se observa en la figura 61 y 62 donde también veremos el paquete captado en Wireshark.

```

20221128 133345.043 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133345.043 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133345.043 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133345.107 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4681 topicid: 0 topicname: riot_devic
20221128 133345.107 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4681 topicid: 1 returncode: 0 (0)
20221128 133345.187 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133345.187 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133345.203 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133345.203 CWNAN0038I Disconnection request received from client NODO2
20221128 133345.203 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)
20221128 133400.258 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133400.258 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133400.258 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133400.322 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4682 topicid: 0 topicname: riot_devic
20221128 133400.322 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4682 topicid: 1 returncode: 0 (0)
20221128 133400.402 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133400.402 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133400.418 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133400.418 CWNAN0038I Disconnection request received from client NODO2
20221128 133400.418 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)
    
```

Figura 61. Log de paquetes de Publicación.

The image shows a Wireshark packet capture analysis. The top pane displays a list of packets, with packet 14 highlighted. The middle pane shows the details of this packet, identifying it as an MQTT-S PUBLISH message. The bottom pane shows the raw data of the packet, which is a JSON object: {"id": "2", "datetime": "2022-06-13 22:10:10", "humedad": "0.00", "temperatura": "26.80"}. Red boxes highlight the source and destination IP addresses in the packet list, the MQTT-S PUBLISH message type in the details pane, and the JSON payload in the raw data pane.

Figura 62. Visualización de paquete de Publicación en Wireshark.

Finalmente se observa la desconexión del Nodo 2 hacia el Broker para dar paso al siguiente nodo, así mismo se envía el respectivo ACK demostrando la desconexión de ambos lados, figura 63 y 64.

```

20221128 133345.043 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133345.043 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133345.043 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133345.107 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4681 topicid: 0 topicname: riot_devic
20221128 133345.107 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4681 topicid: 1 returncode: 0 (0)
20221128 133345.187 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133345.187 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133345.203 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133345.203 CWNAN0038I Disconnection request received from client NODO2
20221128 133345.203 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)
20221128 133400.258 3 2001:db8::5:1883 <- MQTT-S CONNECT cleansession: 1
20221128 133400.258 895793104 2001:db8::5:1883 NODO2 -> MQTT-S CONNACK returncode 0 (0)
20221128 133400.258 CWNAN0000I Client connected to udp port 1885 from NODO2 (2001:db8::5:1883)
20221128 133400.322 3 2001:db8::5:1883 NODO2 <- MQTT-S REGISTER msgid: 4682 topicid: 0 topicname: riot_devic
20221128 133400.322 3 2001:db8::5:1883 NODO2 -> MQTT-S REGACK msgid: 4682 topicid: 1 returncode: 0 (0)
20221128 133400.402 3 2001:db8::5:1883 NODO2 <- MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0
20221128 133400.402 3 ::ffff:127.0.0.1:53338 GoogleBridge -> MQTT-S PUBLISH msgid: 0 qos: 0 retained: 0 (0)
20221128 133400.418 895793104 2001:db8::5:1883 NODO2 <- MQTT-S DISCONNECT duration: 0
20221128 133400.418 CWNAN0038I Disconnection request received from client NODO2
20221128 133400.418 3 2001:db8::5:1883 NODO2 -> MQTT-S DISCONNECT duration: 0 (0)

```

Figura 63. Log de paquetes de Desconexión.

15	5.183692671	2001:db8::5	2001:db8::1	UDP	64	1883	→	1885	Len=2
16	5.183792041	2001:db8::1	2001:db8::5	UDP	64	1885	→	1883	Len=2
17	15.215014064	2001:db8::4	2001:db8::1	UDP	73	1883	→	1885	Len=11
18	15.215116964	2001:db8::1	2001:db8::4	UDP	65	1885	→	1883	Len=3
19	15.279013820	2001:db8::4	2001:db8::1	UDP	79	1883	→	1885	Len=17
20	15.279124630	2001:db8::1	2001:db8::4	UDP	69	1885	→	1883	Len=7
21	15.343001547	2001:db8::4	2001:db8::1	UDP	125	1883	→	1885	Len=63

> Frame 15: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface tap5, id 0

> Ethernet II, Src: ae:36:db:9a:af:46 (ae:36:db:9a:af:46), Dst: 06:c0:cc:60:c7:27 (06:c0:cc:60:c7:27)

> Internet Protocol Version 6, Src: 2001:db8::5, Dst: 2001:db8::1

0110 .... = Version: 6

> .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)

.... 0000 0000 0000 0000 = Flow Label: 0x000000

Payload Length: 10

Next Header: UDP (17)

Hop Limit: 63

Source Address: 2001:db8::5

Destination Address: 2001:db8::1

> User Datagram Protocol, Src Port: 1883, Dst Port: 1885

> Data (2 bytes)

Data: 0218

[Length: 2]

Figura 64. Visualización de captura de Desconexión en Wireshark.

### 4.2.3 Pruebas de Distancias

Para asegurar la conexión en caso de que llegarán a alejarse los nodos el uno del otro, se procederá a realizar un ping en diferentes distancias para poder comprobar que se mantendrán conectados los equipos al alejarse. La primera prueba se realizó a 10 metros de distancia; con un ping de 100 paquetes, se puede observar que el RSSI varía entre -50 y -55 [dbm]. El RSSI nos indicará la intensidad recibida según la distancia entre los enlaces de conexión, por el cual entre más cercano a cero esté más fuerte será la señal.

```

12 bytes from 2001:db8::4: icmp_seq=38 ttl=64 rssi=-53 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=39 ttl=64 rssi=-54 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=40 ttl=64 rssi=-53 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=41 ttl=64 rssi=-53 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=42 ttl=64 rssi=-52 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=43 ttl=64 rssi=-53 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=44 ttl=64 rssi=-52 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=45 ttl=64 rssi=-53 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=46 ttl=64 rssi=-52 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=48 ttl=64 rssi=-53 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=49 ttl=64 rssi=-53 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=50 ttl=64 rssi=-54 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=51 ttl=64 rssi=-53 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=52 ttl=64 rssi=-52 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=53 ttl=64 rssi=-53 dBm time=31.586 ms
13 bytes from 2001:db8::4: icmp_seq=54 ttl=64 rssi=-51 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=55 ttl=64 rssi=-53 dBm time=31.585 ms
12 bytes from 2001:db8::4: icmp_seq=56 ttl=64 rssi=-54 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=57 ttl=64 rssi=-54 dBm time=31.586 ms

```

Figura 65. Captura de ping a los 10 metros.

En el resumen de los paquetes transmitidos podemos ver que únicamente se perdió un paquete con un porcentaje de pérdida de 1% y un tiempo de viaje máximo de 169.424 ms, mínimo de 31.582 ms y un promedio 33.159 ms.

```

riot@riot: ~/RIOT/examples/gnrc_border_router
12 bytes from 2001:db8::4: icmp_seq=94 ttl=64 rssi=-53 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=95 ttl=64 rssi=-53 dBm time=31.606 ms
12 bytes from 2001:db8::4: icmp_seq=96 ttl=64 rssi=-52 dBm time=31.585 ms
12 bytes from 2001:db8::4: icmp_seq=97 ttl=64 rssi=-52 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=98 ttl=64 rssi=-51 dBm time=31.585 ms
12 bytes from 2001:db8::4: icmp_seq=99 ttl=64 rssi=-50 dBm time=31.586 ms

--- 2001:db8::4 PING statistics ---
100 packets transmitted, 99 packets received, 1% packet loss
round-trip min/avg/max = 31.582/33.159/169.424 ms
> ping 2001:db8::4 -c 100

```

Figura 66. Resumen de Estadísticas de envío de ping a los 10 metros.

La siguiente prueba se realizará a 50 metros de distancia; de la misma forma se envió 100 paquetes, y la variación del RSSI va de entre -75 a -80 [dbm].

```

> ping 2001:db8::4 -c 100
ping 2001:db8::4 -c 100
12 bytes from 2001:db8::4: icmp_seq=0 ttl=64 rssi=-74 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=1 ttl=64 rssi=-76 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=2 ttl=64 rssi=-79 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=3 ttl=64 rssi=-78 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=4 ttl=64 rssi=-78 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=5 ttl=64 rssi=-78 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=6 ttl=64 rssi=-79 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=7 ttl=64 rssi=-78 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=8 ttl=64 rssi=-81 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=9 ttl=64 rssi=-80 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=10 ttl=64 rssi=-79 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=11 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=12 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=13 ttl=64 rssi=-78 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=14 ttl=64 rssi=-77 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=15 ttl=64 rssi=-77 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=16 ttl=64 rssi=-77 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=17 ttl=64 rssi=-76 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=18 ttl=64 rssi=-77 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=19 ttl=64 rssi=-78 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=20 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=21 ttl=64 rssi=-79 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=22 ttl=64 rssi=-77 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=24 ttl=64 rssi=-76 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=25 ttl=64 rssi=-76 dBm time=31.586 ms

```

Figura 67. Captura de ping a los 50 metros.

En el resumen de los paquetes transmitidos podemos ver que únicamente se perdió 4 paquetes con un porcentaje de pérdida de 4% y un tiempo de viaje máximo de 108.603 ms, mínimo de 31.583 ms y un promedio 33.165 ms.

```

12 bytes from 2001:db8::4: icmp_seq=98 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=99 ttl=64 rssi=-79 dBm time=87.396 ms

--- 2001:db8::4 PING statistics ---
100 packets transmitted, 96 packets received, 4% packet loss
round-trip min/avg/max = 31.583/33.165/108.603 ms
> ping 2001:db8::4 -c 100

```

Figura 68. Resumen de Estadísticas de envío de ping a los 50 metros.

A continuación, la prueba se realizará a 100 metros de distancia y de la misma se envió 100 paquetes, podemos observar la variación de RSSI va de -85 [dbm] en adelante.

```

> ping 2001:db8::4 -c 100
ping 2001:db8::4 -c 100
12 bytes from 2001:db8::4: icmp_seq=3 ttl=64 rssi=-89 dBm time=68.793 ms
12 bytes from 2001:db8::4: icmp_seq=4 ttl=64 rssi=-88 dBm time=87.396 ms
12 bytes from 2001:db8::4: icmp_seq=6 ttl=64 rssi=-85 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=7 ttl=64 rssi=-86 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=8 ttl=64 rssi=-87 dBm time=183.562 ms
12 bytes from 2001:db8::4: icmp_seq=10 ttl=64 rssi=-88 dBm time=88.747 ms
12 bytes from 2001:db8::4: icmp_seq=12 ttl=64 rssi=-89 dBm time=50.210 ms
12 bytes from 2001:db8::4: icmp_seq=13 ttl=64 rssi=-80 dBm time=155.818 ms
12 bytes from 2001:db8::4: icmp_seq=14 ttl=64 rssi=-89 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=15 ttl=64 rssi=-86 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=17 ttl=64 rssi=-89 dBm time=31.588 ms
12 bytes from 2001:db8::4: icmp_seq=18 ttl=64 rssi=-90 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=19 ttl=64 rssi=-84 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=21 ttl=64 rssi=-91 dBm time=68.793 ms
12 bytes from 2001:db8::4: icmp_seq=22 ttl=64 rssi=-91 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=23 ttl=64 rssi=-85 dBm time=68.793 ms
12 bytes from 2001:db8::4: icmp_seq=38 ttl=64 rssi=-86 dBm time=50.210 ms
12 bytes from 2001:db8::4: icmp_seq=39 ttl=64 rssi=-85 dBm time=87.439 ms
12 bytes from 2001:db8::4: icmp_seq=41 ttl=64 rssi=-86 dBm time=50.210 ms
12 bytes from 2001:db8::4: icmp_seq=43 ttl=64 rssi=-85 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=44 ttl=64 rssi=-87 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=48 ttl=64 rssi=-84 dBm time=50.209 ms
12 bytes from 2001:db8::4: icmp_seq=50 ttl=64 rssi=-87 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=51 ttl=64 rssi=-84 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=52 ttl=64 rssi=-84 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=53 ttl=64 rssi=-85 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=54 ttl=64 rssi=-87 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=55 ttl=64 rssi=-85 dBm time=32.898 ms

```

Figura 69. Captura de ping a los 100 metros.

El Resumen a los 100 metros es similar al de 50 metros, se pierden la misma cantidad de paquetes al igual que tienen el mismo tiempo de viaje.

```

--- 2001:db8::4 PING statistics ---
100 packets transmitted, 96 packets received, 4% packet loss
round-trip min/avg/max = 31.583/33.165/108.603 ms

```

Figura 70. Resumen de Estadísticas de envío de ping a los 100 metros.

Finalmente se realiza una prueba con una distancia de aproximadamente 150 metros en donde existe una pérdida bastante considerable de paquetes haciendo que no sea para nada óptimo el uso a esta distancia, dada la extensión del terreno tampoco se requerirá una distancia tan amplia de aplicación. Se observa la variación del RSSI que va de 81 [dBm] con 29 paquetes perdidos, con un porcentaje de pérdida de 29% y el tiempo de viaje máximo de 50.209 ms, mínimo de 31.585 ms y el promedio de 31.848 ms.

```

> ping 2001:db8::4 -c 100
ping 2001:db8::4 -c 100
12 bytes from 2001:db8::4: icmp_seq=0 ttl=64 rssi=-82 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=1 ttl=64 rssi=-78 dBm time=31.585 ms
12 bytes from 2001:db8::4: icmp_seq=2 ttl=64 rssi=-80 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=3 ttl=64 rssi=-81 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=4 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=5 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=6 ttl=64 rssi=-80 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=7 ttl=64 rssi=-81 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=8 ttl=64 rssi=-81 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=9 ttl=64 rssi=-80 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=10 ttl=64 rssi=-81 dBm time=31.585 ms
12 bytes from 2001:db8::4: icmp_seq=11 ttl=64 rssi=-81 dBm time=31.587 ms
12 bytes from 2001:db8::4: icmp_seq=12 ttl=64 rssi=-81 dBm time=31.586 ms
12 bytes from 2001:db8::4: icmp_seq=13 ttl=64 rssi=-80 dBm time=31.587 ms

--- 2001:db8::4 PING statistics ---
100 packets transmitted, 71 packets received, 29% packet loss
round-trip min/avg/max = 31.585/31.848/50.209 ms

```

Figura 71. Captura de ping a los 150 metros.

#### 4.2.4 Análisis con SDR

Una parte importante para poder verificar el tipo de modulación que se llevará a cabo es importante realizar un análisis de la señal usando el SDR, para ello necesitamos configurar el equipo con la frecuencia central que como se mencionó anteriormente es 864.225 [Mhz] que se encuentra en el canal 1 de la tabla de OFDM dentro del estándar IEEE 802.15.4, al recibir la señal podemos observar la forma de la onda la cual está en el dominio de la frecuencia y como se puede observar la Magnitud Normalizada de 0.45 [db], véase la figura 68.



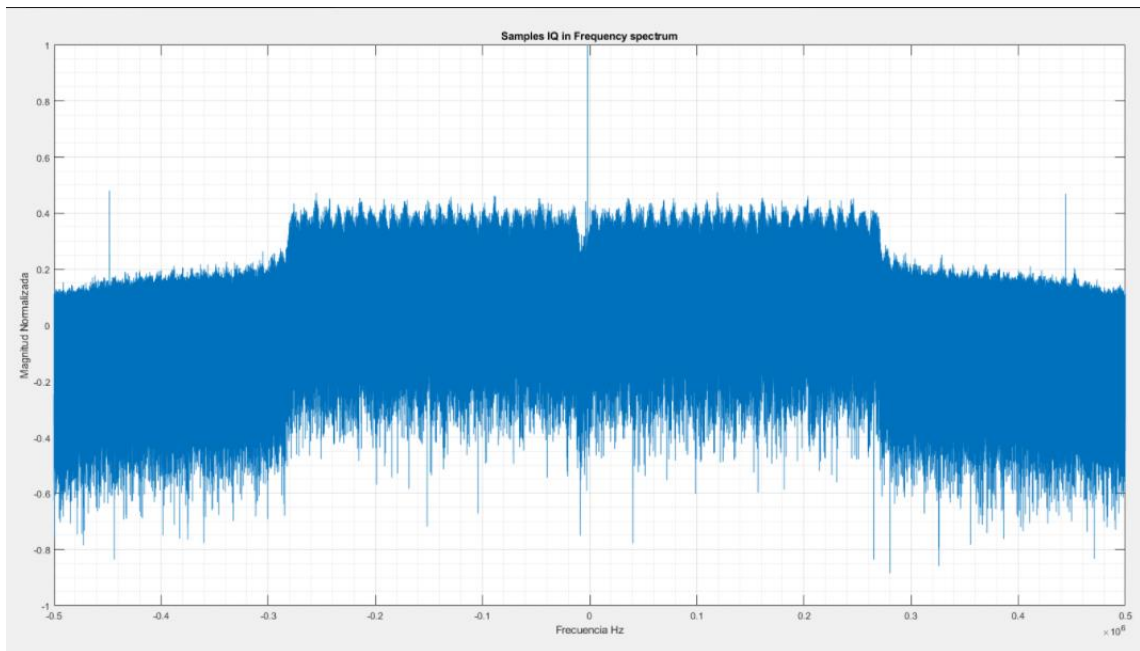


Figura 72. Espectro de la Señal.

En la siguiente figura 69 (a) se puede observar 3 gráficas la primera de ellas nos muestra cada señal enviada en dicha frecuencia tomando como muestras 8 envíos de paquetes seguidos de su ACK. En la figura 69 (b) se puede observar el espectro de la señal en función de la frecuencia normalizada, además podemos notar el espacio de canal el cual es de aproximadamente 650 [Khz] desde el inicio de la forma de la onda hasta el final, teóricamente se deberían tener 800 [KHz] en este se indica el muestreo del espectro esta se desplaza y baja dentro de la zona limitada por el ancho de banda.

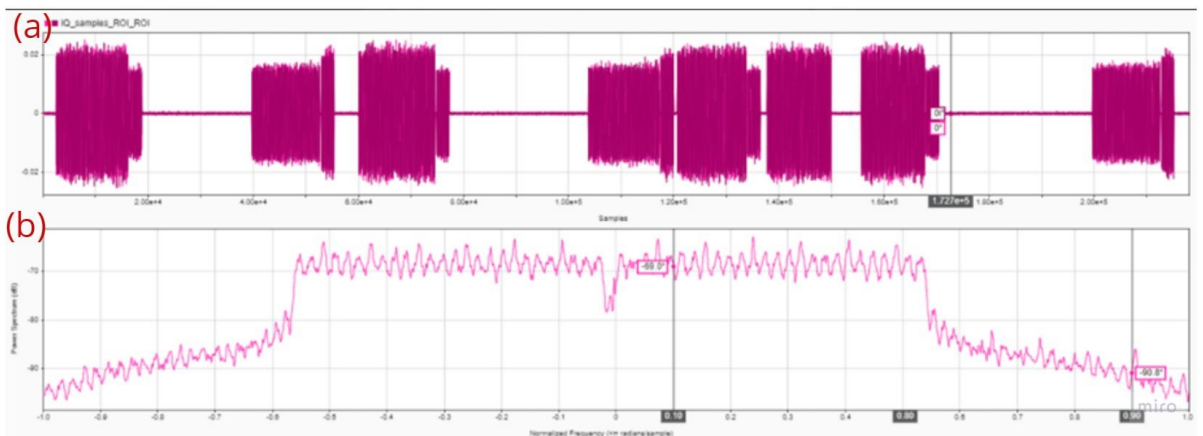


Figura 73. Análisis del espectro de la señal por muestra.

Se toma los dos primeros paquetes enviamos junto con su ACK y se puede observar las mismas gráficas que la anterior con la duración de cada paquete enviado, agregándole el ruido que puede ser captado por el aire, figura 70. Al primer paquete se

le nota una duración de 3 segundos y su ACK una duración de 0.5 segundo, 1.5 segundos de espera el siguiente paquete tiene una duración de 3.5 segundos y su ACK 0.5 segundo. Dando un total de 9 segundos en los dos primeros paquetes con su respectivo ACK.

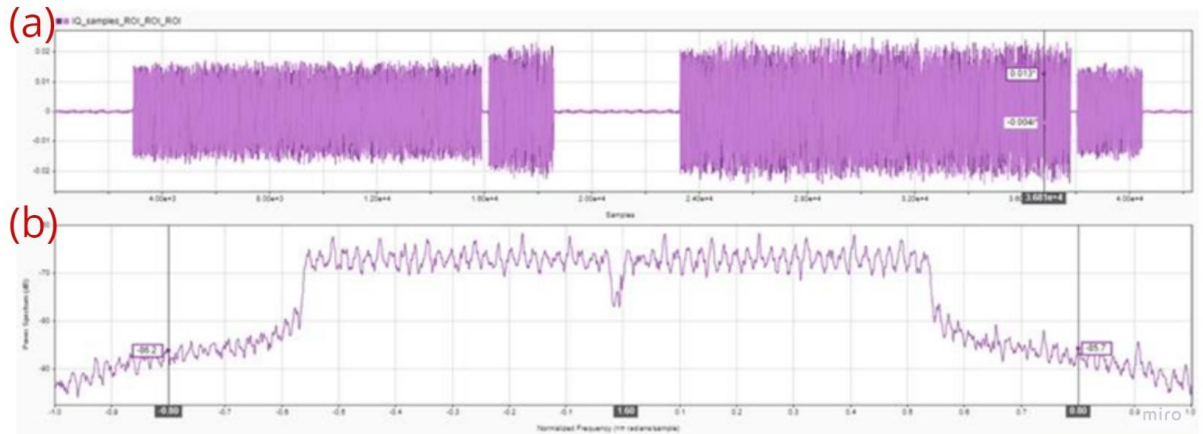


Figura 74. Análisis de dos paquetes con su ACK.

## Conclusiones

- Principalmente para poder conocer los principales aspectos a ser incluidos en el prototipo del cultivo de arándanos en un invernadero, se realiza una investigación de cada uno de los agentes involucrados en la producción de la planta; los cuales son PH en el agua de riego, temperatura del ambiente y la Humedad Relativa del Suelo para conocer si se encuentra bien hidratada la maceta, esto nos permitió entender cada uno de esos aspectos que repercuten y es de vital importancia en su proceso de crecimiento.
- Los Nodos sensores y el Nodo Central tienen una comunicación eficiente con las pruebas de alcance que se supo realizar en cada una de las distancias, al estar muy cerca entre los nodos y sin obstáculos que impidan la visión se logró cubrir el espacio del invernadero donde se va a monitorear. Las pruebas a distintas distancias muestran la efectividad del prototipo hasta los 100 metros a partir de los 150 se registran importantes pérdidas de paquetes, de acuerdo al desarrollo del proyecto teóricamente los módulos se podrían implementar a los 200 metros, luego de la realización de análisis de cobertura para la implementación de los módulos se determinó un máximo alcance de 100 metros de nodo a nodo.
- Esta WSN (Red de Sensores) se desarrolla con el fin de poder monitorear cada uno de los parámetros necesarios para una buena producción y poder registrar dichos datos permitiendo que el agricultor en el invernadero pueda tomar la decisión que procure pertinente. Con ello se pretende la mejora de la calidad de producción de cada maceta de Arándanos evitando así pérdida de producto por mal cuidado y desatención de los parámetros de PH, Humedad Relativa del Suelo y la Temperatura del ambiente, los cuales son de suma importancia en la germinación y producción de la planta.
- En las pruebas realizadas con cada sensor, se pudo observar que existen muchos parámetros que pueden variar los valores de dicho sensor, estos parámetros pueden ser resistencias y capacitores dañados, PH del agua alterada, factores de temperatura y humedad del aire mal calibrados, etc. Por ello, es importante escoger el más preciso de los sensores y que no se vea afectado por el ambiente, si estos valores no se cuidados pueden resultar en el daño del cultivo y la menor calidad del mismo. Dado este caso es imperativo mantener siempre los sensores en buenas condiciones con sus valores calibrados.
- La eficiencia del sistema en relación con el consumo de energía no fue óptima debido a que las baterías que se consideraron para el diseño del equipo tuvieron una duración

de 12 horas consecutivas, al ser este un sistema que debe estar en constante sensado del cultivo y de cada uno de los parámetros de la maceta se debe tener conseguir un consumo de energía más eficiente.

## Recomendaciones

- Es muy importante considerar la distancia del lugar, es decir la extensión del mismo y si existirá algún medio que corte la línea de vista entre los equipos, impidiendo que estos puedan comunicarse entre sí ya que como se mencionó si existe interrupciones pueden llegarse a perder paquetes por lo tanto la comunicación entre los nodos.
- Previa a la manipulación de cada equipo es imperativo que se revisen las hojas de datos técnicos de cada uno de los equipos para un manejo adecuado de los mismos y que se logre el mejor funcionamiento.
- Al igual que es importante la revisión de las Hojas de Datos se debe realizar una correcta calibración de cada uno de los sensores, ya que en ambientes diferentes las circunstancias van a variar resultando en una indebida aplicación dentro del cultivo.
- Para poder optimizar el sistema se recomienda un análisis más profundo de las bandas de aplicación, al ser este equipo proveniente de Europa las únicas bandas que han sido probadas son las europeas, en un estudio más profundo se deberá analizar estos aspectos.
- Se recomienda establecer algún mecanismo que permita utilizar energía autosustentable y que esto permita recargar las baterías para alimentar el circuito y el prototipo en general, así mantener el sistema activo siempre. Otra opción para lograr mejor este aspecto y alargar el tiempo de vida se recomienda configurar en los nodos sensores una función de modo sleep para que únicamente el sistema se active en el momento que los datos estén llegando en un rango demasiado elevado.
- Una recomendación para aplicar a futuro es aplicar el procesamiento más extendido de los datos para poder predecir el tiempo en el que los rangos podrían elevarse o en su defecto disminuir. Estos rangos apropiados en el PH para la producción son de 4,5 a 5,5, para la humedad del suelo el rango va entre 80% al 100%; por otro lado, para la temperatura dentro del vivero varía en la noche alrededor de 10° y durante el día alrededor de 30°.
- Como se pudo observar muchos de los datos en las capturas de Wireshark no se encuentran encriptados y pueden ser interceptados por cualquier radio frecuencia y obtener las credenciales de acceso, por ello es necesario implementar sistemas de seguridad y encriptación de los datos.

## Bibliografía

- Adenowo, A. A., & Adenowo, B. A. (2013). Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach. *International Journal of Scientific & Engineering Research*, V4-17.
- Aosong Electronics Co. (2020). *DHT22 Datasheet*. Obtenido de Aosong Electronics Co.,Ltd: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>
- Arduino. (2022). *Arduino UNO R3 Overview* . Obtenido de UNO R3: <https://docs.arduino.cc/hardware/uno-rev3>
- ARDUINO. (2022). *Serial [Communication]*. Obtenido de Arduino Documentation, Functions: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>
- Baccelli, E., Hahm, O., Günes, M., Wählich, M., & Schmidt, T. (2013). RIOT OS: Towards an OS for the internet of things. *ResearchGate*.
- Bradner, S. &. (1999). *Benchmarking Methodology for Network Interconnect Devices*. IETF.
- DFRobot. (2020). *Capacitive Soil Moisture Sensor: SEN0193*. Obtenido de DFRobot Website: [https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193_Web.pdf)
- DFRobot Gravity. (2020). *Gravity: Analog pH Sensor/Meter Kit V2 SKU:SEN0161-V2*. Obtenido de Sigma Electronica Website: <https://www.sigmaelectronica.net/wp-content/uploads/2019/02/SEN0161-v2.pdf>
- Dolci, R. (2017). IoT Solutions for Precision Farming and Food Manufacturing: Artificial Intelligence Applications in Digital Food. *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, 384-385.
- Fang, Y., Nunez, G. H., da Silva, M. N., Phillips, D. A., & Munoz, P. R. (2020). A Review for Southern Highbush Blueberry Alternative Production Systems. *Agronomy*, 10.
- Google Cloud. (2022). *¿Qué es la IaaS?* Obtenido de Google Cloud - Documentation: <https://cloud.google.com/learn/what-is-iaas?hl=es>
- Herrera, C., Dien Tran, T., Egas, C., Pereira, V., & Sá Silva, J. (2016). Reliability in Wireless Sensor Networks for Environment Monitoring. *Revista Politécnica*, 37 (1).
- IEEE. (2019). *1900.1-2019 - IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management*. IEEE.
- IEEE Computer Society. (2020). *IEEE Standard for Low-Rate Wireless Networks*. Carleton University: IEEE.
- INDUSTRIAL SHIELDS. (2019). *Open Mote B User Guide* . Obtenido de INDUSTRIAL SHIELD Website: [https://www.industrialshields.com/es\\_ES/shop/product/is-omb-001-openmote-b-721#attr=](https://www.industrialshields.com/es_ES/shop/product/is-omb-001-openmote-b-721#attr=)
- Ivanov, S., Bhargava, K., & Donnelly, W. (2015). Precision Farming: Sensor Analytics. *IEEE Intelligent Systems*, 76-80.

- Jing, Y., Yuzhi, Z., Dan, D., Xiao, W., Ping, Y., Lingfang, C., . . . Zetao, L. (2017). *An early warning system of diseases and pests for blueberry based on WSN*. Dalian: IEEE.
- Lazarescu, M. (2013). Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 45-54.
- Matin, M., & Islam, M. M. (2012). Overview of Wireless Sensor Network. *Wireless Sensor Networks - Technology and Protocols*, 3-5.
- MongoDB. (2022). *Data as a Service (DaaS) Explained*. Obtenido de MongoDB - Documentation: <https://www.mongodb.com/initiatives/data-as-a-service>
- Ochmian, I., Błaszak, M., Lachowicz, S., & Piwowarczyk, R. (2020). The impact of cultivation systems on the nutritional and phytochemical content, and microbiological contamination of highbush blueberry. *scientific reports*, 16.
- Ochmian, I., Grajkowski, J., Mikiciuk, G., Ostrowska, K., & Chelpinski, P. (2009). Mineral composition of high blueberry leaves and fruits depending on substrate type used for cultivation. *Journal of Elementology*, 509-516.
- República Del Ecuador Consejo Nacional De Planificación. (2013). *PLAN NACIONAL PARA EL BUEN VIVIR*. Quito, Ecuador: SENPLADES.
- Seah, W. K., Eu, Z. A., & Tan, H.-P. (2009). Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges. *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, 1-5.
- Steve. (27 de Septiembre de 2021). *Steve's Internet Guide*. Obtenido de MQTT Brokers and Cloud Hosting Guide: <http://www.steves-internet-guide.com/mqtt-hosting-brokers-and-servers/>
- Trickey, D. G. (2020). An investigation into RIOT-OS for use in 6LoWPAN sensor networks. *BSc Computer Science*, 79-80.
- Wiangtong, T., & Sirisuk, P. (2018). IoT-based Versatile Platform for Precision Farming. *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, 438-441.
- Xie, Z.-S., & Wu, X.-C. (2009). STUDIES ON SUBSTRATES FOR BLUEBERRY CULTIVATION. *International Society for Horticultural Science*.

- 802.15 WG - Wireless Specialty Networks (WSN) Working Group. (2020). *IEEE 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks*. IEEE.
- Bradner, S., & McQuaid, J. (1999). *Benchmarking Methodology for Network Interconnect Devices*. IETF.
- BURBANO GARCIA, J. L. (2014). *Red WSN para el control y monitoreo de una sistema de riego por goteo de una plantación de fresas en la granja experimental Yuyucocha - UTN*. Yuyucocha: UTN.
- CARDENAS DELGADO, J. E. (2021). *Sistema de adquisición de variables climáticas para invernaderos: Manejo de datos*. Ibarra: UTN.
- CASTILLO IMBAQUINGO, D. X. (2012). *Diseño e implementación de un sistema de alarma comunitaria a base de módulos inalámbricos utilizando tecnología ZIGBEE*. Ibarra: UTN.
- GAVILANEZ ZUMARRAGA, L. A. (2021). *Medidor de CO2 y Temperatura para invernadero UTN*. Ibarra: UTN.
- IEEE. (2019). *1900.1-2019 - IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management*. IEEE.
- Jing, Y., Yuzhi, Z., Dan, D., Xiao, W., Ping, Y., Lingfang, C., . . . Zetao, L. (2017). *An early warning system of diseases and pests for blueberry based on WSN*. Dalian: IEEE.
- Junjalearnvong, T., Habara, T., Okumura, R., Mizutani, K., & Harada, H. (2017). *A dynamic routing protocol supporting mobile nodes in Wi-SUN FAN systems*. Bali, Indonesia: IEEE.
- Mizutani, K., Okumura, R., Mizutani, K., & Harada, H. (2020). *A Home Area Heterogeneous Wireless Management Scheme by Wi-SUN FAN and Wi-Fi Systems*. Las Vegas, NV, USA: IEEE.
- Mochizuki, K., Obata, K., Mizutani, K., & Harada, H. (2016). *Development and field experiment of wide area Wi-SUN system based on IEEE 802.15.4g*. Reston, VA, USA: IEEE.
- PALACIOS ECHEVERRIA, J. A. (2017). *Diseño de una red de sensores (WSN) con tecnología 802.15.4, basado en el concepto agricultura de precisión para el control y monitoreo de cultivos de hortalizas bajo invernadero en la granja la pradera de la Universidad Técnica del Norte*. Ibarra, Ecuador: UTN.
- República Del Ecuador Consejo Nacional De Planificación. (2013). *PLAN NACIONAL PARA EL BUEN VIVIR*. Quito, Ecuador: SENPLADES.
- Shahid, A., Fontaine, J., Camelo, M., Haxhibeqiri, J., Saelens, M., Khan, Z., & Moerman, I. (2019). *A Convolutional Neural Network Approach for Classification of LPWAN Technologies: Sigfox, LoRA and IEEE 802.15.4g*. Boston, MA, USA: IEEE.
- Simões Martins, N., Calado, M. d., Pombo, J. A., & Mariano, S. J. (2016). *Blueberries field irrigation management and monitoring system using PLC based control and wireless sensor network*. Florence, Italy: IEEE.



## Anexos (Hojas de Datos/ Código/ Encuesta/ Entrevista)

Video de Demostración

[Presentación del Trabajo de Grado - Johanna Yépez - UTN](#)

