



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

“RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO PARA MONITOREO AMBIENTAL EN INVERNADEROS.”

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

AUTOR: CARLOS ALBERTO GORDILLO GORDILLO

DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA

Ibarra-Ecuador

2021



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

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 DEL CONTACTO			
CÉDULA DE	100376853-6		
APELLIDOS Y	Gordillo Gordillo Carlos Alberto		
DIRECCIÓN	13 de abril 27-42 y Otavalo		
E-MAIL	cagordillo@utn.edu.ec		
TELÉFONO FIJO	062601723	TELÉFONO MÓVIL	0996028911

DATOS DE LA OBRA	
TÍTULO	“Red de sensores inalámbricos bajo protocolo lora y gestión de procesos para la analítica de datos mediante meta sistema operativo para monitoreo ambiental en invernaderos”
AUTOR	Gordillo Gordillo Carlos Alberto
FECHA	2021
PROGRAMA	Pregrado
TÍTULO	Ingeniero en Electrónica y Redes de Comunicación
DIRECTOR	MsC. Edgar Alberto Maya Olalla

CONSTANCIAS.

El autor manifiesta 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 el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 7 días del mes de Diciembre de 2021

EL AUTOR



Carlos Alberto Gordillo Gordillo

CI: 1003768536-6



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN.

MsC. Edgar Maya, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN
CERTIFICA:

Que, el presente trabajo de Titulación "RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO PARA MONITOREO AMBIENTAL EN INVERNADEROS." Ha sido desarrollado por el señor Carlos Alberto Gordillo Gordillo bajo mi supervisión.

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

A handwritten signature in black ink, appearing to be 'E. Maya', is written over a horizontal dotted line.

MsC. Edgar Alberto Maya Olalla

DIRECTOR

AGRADECIMIENTO

Agradezco en primer lugar a Dios por la vida y la familia con la que me bendijo, por guiarme siempre por el camino del bien, dándome sabiduría y fortaleza para afrontar los momentos difíciles y enseñándome a seguir adelante y a nunca darme por vencido.

A mis padres por ese apoyo incondicional y además por todo el cariño y amor que me brindan todos los días, ayudándome a siempre seguir adelante.

A mi hermano por esa oportunidad única que me brindo para poder trabajar y seguir con mis estudios, cosa que me ayudó mucho a crecer como persona.

A mi tío Alfredo, mi tía Susanita y mi primo Christian por ser ese apoyo permanente en todas las etapas de mi vida.

A mis amigos incondicionales y en especial a mi mejor amigo Cristian quien siempre me supo apoyar en cada decisión y en cada paso que daba en la vida.

Al Ingeniero Paul Rosero, director de mi trabajo de tesis, quien me brindo la oportunidad de trabajar con él en este proyecto tan importante, por su tiempo, su buena predisposición, su apoyo incondicional, sus conocimientos y por abrirme la mente a que la meta no es conformarse si no recorrer el mundo entero.

A mis asesores el Ingeniero Mauricio Domínguez y el Ingeniero Edgar Maya quienes han trabajado conmigo revisando mi trabajo de titulación con toda la paciencia posible.

A todos los docentes de la carrera de Ingeniería en Electrónica y Redes de Comunicación quienes me han impartido sus conocimientos a lo largo de la carrera universitaria, en especial al Economista Mario Mediavilla quien ha sido un pilar fundamental en mi formación universitaria.

Carlos Alberto Gordillo Gordillo

DEDICATORIA

Dedico este trabajo a:

Mis padres Jorge e Inés, por ser el apoyo incondicional en cada etapa de mi vida y estar siempre pendientes de mi apoyándome, brindándome su cariño y teniéndome paciencia en cada decisión que tomo, por sus palabras de aliento que siempre me supieron motivar a querer más y que gracias a todo eso he logrado ser la persona que soy hoy en día.

A mi hermano Jorge Luis por todos los momentos que hemos vivido juntos, por su apoyo incondicional y permanente para sacar adelante a toda la familia.

A mi tío Alfredo, mi tía Susanita y mi primo Christian por ser pilares fundamentales en mi formación como persona y como profesional, por el inmenso apoyo que me han brindado a lo largo de toda mi vida para poder salir adelante.

A mis amigos del alma y en especial a mi mejor amigo Cristian, por siempre estar ahí dispuesto a ayudar en lo que necesite, por apoyarme en cada momento y alentándome a seguir adelante.

Carlos Alberto Gordillo Gordillo

ÍNDICE DE CONTENIDOS

CONSTANCIAS.....	III
CERTIFICACIÓN.....	IV
AGRADECIMIENTO.....	V
DEDICATORIA.....	VI
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS.....	XII
ÍNDICE DE ECUACIONES.....	XIII
RESUMEN.....	XIV
ABSTRACT.....	XV
CAPÍTULO I. ANTECEDENTES.....	1
1.1 Introducción.....	1
1.2 Tema.....	1
1.3 Problema.....	1
1.4 Objetivos.....	3
1.4.1. Objetivo General.....	3
1.4.2. Objetivos Específicos.....	3
1.5 Alcance.....	4
1.6 Justificación.....	5
CAPÍTULO II REVISIÓN BIBLIOGRÁFICA.....	8
2.1. Introducción.....	8
2.2. Agricultura inteligente.....	8
2.3. Agricultura de precisión.....	9
2.3.1. Etapas de la agricultura de precisión.....	10
2.3.2. Herramientas de la agricultura de precisión.....	11
2.4. Invernadero.....	12
2.4.1. Clasificación de invernaderos.....	12
2.4. Cultivo de pimiento.....	17
2.4.1. Condiciones ambientales para el desarrollo de una planta de pimiento.....	18
2.5.1.1. Luminosidad.....	19
2.4.1.2. Temperatura.....	19
2.5.1.3. Humedad relativa.....	19
2.5.1.4. Suelo.....	19
2.5.1.5. Dióxido de Carbono CO ₂	20
2.5. Adquisición de datos.....	20
2.6.1. Recolección de datos en la agricultura de precisión.....	21

2.6.2.	Sensor.....	21
2.7.	Redes de sensores inalámbricos.....	22
2.7.1.	Elementos de las WSN.....	22
2.7.2.	Funcionamiento.....	23
2.7.3.	Arquitectura.....	23
2.7.4.	Topologías de red.....	24
2.8.	LoRa.....	25
2.8.1.	LoRaWAN.....	25
2.8.1.1.	Arquitectura de una red LoRaWAN.....	26
2.8.1.2.	Clases de dispositivos LoRaWAN.....	29
2.8.1.3.	Estructura del paquete.....	30
2.8.1.4.	Capacidad de red.....	32
2.8.1.5.	Factor de dispersión.....	32
2.8.1.6.	Tiempo en el aire de la transmisión.....	33
2.8.1.7.	Seguridad LoRaWAN.....	33
2.9.	Meta sistema operativo.....	34
2.9.1.	Funciones de un MSO.....	34
2.9.2.	Principios del meta sistema operativo.....	38
2.9.1.1.	Comunicación y Middleware.....	38
2.9.1.2.	Operación del sistema distribuido.....	39
2.9.3.	Estructura de un MSO.....	40
2.9.4.	Arquitectura del sistema.....	41
2.10.	Hardware libre.....	42
2.10.1.	Sistemas embebidos.....	43
2.10.2.	Plataformas de código abierto.....	43
CAPITULO III. DISEÑO DEL SISTEMA.....		47
3.1.	Descripción general del sistema.....	47
3.2.	Metodología de diseño.....	48
3.3.	FASE 1: Requisitos y requerimientos.....	49
3.3.1.	Características del sistema.....	50
3.3.2.	Definición de acrónimos y abreviaturas.....	51
3.4.	Requerimientos.....	51
3.4.1.	Situación actual.....	51
3.4.2.	Requerimientos Stakeholders.....	53
3.2.1.	Requerimientos iniciales del sistema.....	53
3.2.2.	Requerimientos de arquitectura.....	54
3.3.	FASE 2: Diseño del prototipo.....	55

3.3.1.	Topología de la red.....	56
3.4.	Elección de hardware y software para el sistema.....	57
3.4.1.	Elección de Hardware.	57
3.4.1.1.	Nodos 1 y 2.	57
3.4.1.2.	Nodo servidor.....	61
3.4.2.	Elección de software.	62
3.4.2.1.	Nodos 1 y 2.	62
3.4.2.2.	Nodo servidor.....	63
3.5.	Hardware destinado para el sistema.	63
3.5.1.	Microcontrolador SparkFun LoRa Gateway.....	64
3.5.2.	Sensor de luz UV SparkFun (VEML6075).....	65
3.5.3.	Sensor de temperatura y humedad de CO ₂ (SCD30).	66
3.5.4.	Módulo GPS (SAM-M8Q).....	67
3.5.5.	Sistema embebido Raspberry Pi Zero.	68
3.6.	Nodos 1 y 2.	69
3.6.1.	Diagrama de flujo nodos 1 y 2.	71
3.6.2.	Ciclo de trabajo y tiempo en el aire (ToA).....	72
3.6.3.	Tipo de clase seleccionado para la red WSN.	73
3.6.4.	Pérdidas en el espacio libre.	75
3.6.5.	Ubicación de los nodos.	76
3.6.6.	Zona de Fresnel.	80
3.6.6.1.	Zona de Fresnel entre nodo 1 y puerta de enlace.	81
3.6.7.	Potencia radiada isotrópica efectiva (PIRE).....	82
3.6.8.	Indicación de intensidad de señal recibida (RSSI).....	83
3.6.9.	Factor de Propagación y sensibilidad con relación al receptor.	83
3.7.	Nodo servidor.....	84
3.7.1.	Diagrama de flujo nodo servidor.....	85
3.8.	Diagrama de bloques de funcionamiento.....	86
CAPITULO IV. IMPLEMENTACIÓN Y PRUEBAS.....		89
4.1.	FASE 3: Implementación de la red.....	89
4.1.1.	Configuración placas SparkFun.....	89
4.1.2.	Configuración de sensores.....	90
4.1.3.	Configuración de módulos LoRaWAN RF95.....	93
4.1.4.	Configuración Meta Sistema Operativo.....	96
4.1.5.	Integración del Nodo 1 en un vehículo.....	97
4.2.	FASE 4: Verificación.....	98
4.2.1.	Pruebas en el invernadero.....	98

4.2.2.	Pruebas comunicación inalámbrica	99
4.2.3.	Lectura del Sensado	102
4.2.4.	Patrones de crecimiento	106
4.3.	Costo del sistema.....	108
4.3.1.	Costo de Hardware	108
4.3.2.	Costo de infraestructura	109
4.3.3.	Costo de Ingeniería.....	110
4.3.4.	Costo de Software	110
4.3.5.	Costo de implementación del sistema general	110
4.4.	Beneficios del sistema.....	111
4.5.	Conclusiones	112
4.6.	Recomendaciones.....	113
	REFERENCIAS.....	115
	GLOSARIO DE TÉRMINOS Y ACRÓNIMOS.....	119
	ANEXOS	120
	ANEXO 1: Datasheet Modulo Rf95	120
	Anexo 2: Datasheet sensor UV VEML6075.....	124
	Anexo 3: Datasheet sensor CO2, Temperatura, Humedad SCD30	126
	Anexo 4: Datasheet modulo GPS SAM-M8Q.....	130
	Anexo 5: Código puerta de enlace	134
	Anexo 6: Código Nodos 1 y 2.....	137
	Anexo 7: Código MSO	141
	Anexo 8: Evidencias fotográficas	146
	Anexo 9: Acciones monitor serial.....	151

ÍNDICE DE FIGURAS

Figura 1.	Agricultura inteligente.....	9
Figura 2.	Etapas de la Agricultura de Precisión.....	11
Figura 3.	Invernadero tipo plano.....	13
Figura 4.	Invernadero tipo capilla.....	13
Figura 5.	Invernadero tipo doble capilla.....	14
Figura 6.	Invernadero tipo diente de sierra.....	15
Figura 7.	Invernadero tipo parral.....	16
Figura 8.	Invernadero tipo túnel.....	16
Figura 9.	Elementos de una WSN.....	22

Figura 10. Funcionamiento de una red WSN.....	23
Figura 11. Topologías de red WSN.	25
Figura 12. Arquitectura básica de LoRa y LoRaWAN.....	27
Figura 13. Estructura del paquete LoRa.	31
Figura 14. Comunicación de nodos	39
Figura 15. Estructura simple de un MSO.....	40
Figura 16. Raspberry Pi.	44
Figura 17. Modelos de Arduino.	44
Figura 18. SparkFun LoRa Gateway - 1-Channel (ESP32).....	46
Figura 19. Diagrama del método en cascada.	49
Figura 20. Distribución de Nodos	50
Figura 21. Invernadero.....	52
Figura 22. Diagrama de Bloques del Sistema.....	56
Figura 23. SparkFun LoRa Gateway - 1-Channel (ESP32).....	64
Figura 24. SparkFun UV Light Sensor Breakout.....	65
Figura 25. Sensor de temperatura y humedad de CO ₂	66
Figura 26. Módulo GPS (SAM-M8Q).	67
Figura 27. Placa Raspberry Pi Zero.	68
Figura 28. Diagrama Circuital de los Nodos 1 y 2	70
Figura 29. Diseño de la PCB - Shield para los nodos 1 y 2.....	70
Figura 30. Vista general de los Nodos 1 y 2.....	71
Figura 31. Diagrama de Flujo del Nodos 1 y 2.....	72
Figura 32. Representación de Clase A en LoRaWAN.....	74
Figura 33. Representación de Clase B en LoRaWAN.....	74
Figura 34. Representación de Clase C en LoRaWAN.....	75
Figura 35. Línea de vista entre el nodo servidor y nodos sensores.....	76
Figura 36. Puntos de referencia de cada nodo en Radio Mobile.	78
Figura 37. Parámetros principales de la red.....	79
Figura 38. Parámetros principales del sistema de enlaces.	80
Figura 39. Resultado de la simulación del cultivo de pimiento.....	80
Figura 40. Zona de Fresnel e Intensidad de señal-nodo 1.....	81
Figura 41. Zona de Fresnel e Intensidad de señal-nodo 2.....	82
Figura 42. Diagrama Circuital del Nodo Servidor.....	84
Figura 43. Diagrama de Flujo del Nodo Servidor.....	85
Figura 44. Diagrama de Bloques Funcionamiento	86
Figura 45. Diagrama de Bloques Toma de datos de sensores.....	87
Figura 46. Diagrama de Bloques Comunicación Inalámbrica.	88
Figura 47. Diagrama de Bloques de Procesamiento y almacenamiento.....	88
Figura 48. Reconocimiento del SparkFun LoRa Gateway 1-Channel en la PC	89
Figura 49. Carga exitosa del Sketch al SparkFun LoRa Gateway 1-Channel	90
Figura 50. Uso de memoria del SparkFun LoRa Gateway 1-Channel	90
Figura 51. Declaración de librerías de los sensores	91
Figura 52. Lectura de sensores.....	91
Figura 53. Verificación de Temperatura con la página web www.wheater.com	92

Figura 54. Verificación de coordenadas GPS con la página web www.https://www.coordenadas-gps.com	92
Figura 55. Configuración de Direcciones y Frecuencia de operación	94
Figura 56. Intercambio de mensajes entre nodos	95
Figura 57. Envío de coordenadas GPS	95
Figura 58. Recepción de informacion mediante LoRa	96
Figura 59. Interfaz Anaconda Navigator	96
Figura 60. Programación interfaz gráfica en Visual Studio Code	97
Figura 61. Nodo 1 sobre vehículo autónomo.....	98
Figura 62. Vehículo autónomo dentro del invernadero.	99
Figura 63. Frecuencia LoRa.....	99
Figura 64. Muestreo espectro LoRa.....	100
Figura 65. Distancias de toma de datos.	101
Figura 66. Interfaz gráfica.....	103
Figura 67. Datos de las variables de los nodos.	103
Figura 68. Datos de la variable UV.	104
Figura 69. Datos de la variable CO2.....	104
Figura 70. Datos de la variable Temperatura.	105
Figura 71. Datos de la variable Humedad.....	105
Figura 72. Relación entre CO2 y humedad.....	106
Figura 73. Relación entre CO2 y temperatura.	107
Figura 74. Datos en la mañana.....	107
Figura 75. Datos en la tarde.	108

ÍNDICE DE TABLAS

Tabla 1. Herramientas de la agricultura de precisión.	11
Tabla 2. Parámetros óptimos para el cultivo de pimiento.....	18
Tabla 3. Sensores necesarios para monitorear el cultivo de pimiento.	21
Tabla 4. Rango de frecuencias de Lora WAN y tecnologías similares	28
Tabla 5. Rango de Frecuencias de LoRa WAN y tecnologías similares aplicables a Ecuador28	
Tabla 6. Términos abreviados y acrónimos usados durante el análisis de requerimientos.....	51
Tabla 7. Lista de Stakeholders del proyecto	53
Tabla 8. Requerimientos Iniciales del Sistema.	53
Tabla 9. Requerimientos de Arquitectura.	54
Tabla 10. Selección del microcontrolador de los Nodos 1 y 2.	57
Tabla 11. Selección del sensor de rayos UV para los Nodos 1 y 2.....	58
Tabla 12. Selección del sensor CO, temperatura y humedad para los Nodos 1 y 2.	59
Tabla 13. Selección del sensor GPS para los Nodos 1 y 2.	60
Tabla 14. Selección del sistema embebido en el Nodo Servidor.....	61
Tabla 15. Comparativa de software disponibles para los Nodos 1 y 2.....	62

Tabla 16. Comparativa de software disponibles del Nodo Servidor	63
Tabla 17. Características principales del SparkFun LoRa Gateway.....	64
Tabla 18. Características principales del sensor SparkFun UV Light Sensor Breakout.....	65
Tabla 19. Características principales del sensor de temperatura y humedad de CO ₂	66
Tabla 20. Características principales del módulo GPS (SAM-M8Q).....	67
Tabla 21. Características de la Placa Raspberry Pi Zero.	68
Tabla 22. Coordenadas geográficas de cada nodo.	77
Tabla 23. Relación de Factor de Propagación.	83
Tabla 24. Factor de Dispersión empleado.....	101
Tabla 25. Costos de Hardware	109
Tabla 26. Costos de Infraestructura	109
Tabla 27. Costos de Ingeniería.....	110
Tabla 28. Costos de Software	110
Tabla 29. Costos de implementación del sistema	111

ÍNDICE DE ECUACIONES

(Ecuación 1).....	30
(Ecuación 2).....	73
(Ecuación 3).....	75
(Ecuación 4).....	83
(Ecuación 5).....	93
(Ecuación 6).....	93
(Ecuación 7).....	93

RESUMEN

Las redes de sensores inalámbricas en la actualidad se han posicionado como una de las tecnologías que han revolucionado el mundo de la agricultura, dando paso muchas aplicaciones entre las que resalta la agricultura de precisión. El presente proyecto consiste en el desarrollo de un sistema de monitoreo de cultivos en invernaderos basado en redes de sensores inalámbricos LPWAN (Low Power Wide Area Network), con un Meta Sistema Operativo, el cual monitorea las variables ambientales que influyen en el crecimiento y desarrollo de cultivos para encontrar patrones adecuados de crecimiento mediante la implementación de algoritmos de aprendizaje de máquina y que de esta manera determinados procesos como la aplicación del riego no se basen en criterios empíricos adquiridos con la experiencia de las personas encargadas de los invernaderos, sino que en base al uso de tecnología se obtenga patrones adecuados para el desarrollo de los diferentes cultivos, además se controle la temperatura y demás variables presentes, para que de esta manera se mejore el desarrollo del cultivo.

El sistema puede monitorear los rayos UV, el CO₂, la temperatura y la humedad que son factores primordiales en el crecimiento de la planta, para luego formar una base de datos robusta que permita encontrar patrones adecuados de crecimiento en base a algoritmos de aprendizaje de máquina.

Se realizaron pruebas del sistema mediante la implementación del prototipo dentro del invernadero lo que permitió comprobar el correcto funcionamiento de los nodos para el monitoreo de los cultivos, el cual permite tener los patrones del crecimiento de la planta y apreciar si se encuentra desarrollándose de una manera adecuada, los datos obtenidos se los recolecta y procesa en el Meta Sistema Operativo y la visualización de la información se la realiza a través de una interfaz gráfica.

ABSTRACT

Wireless sensor networks today have positioned themselves as one of the technologies that have revolutionized the world of agriculture, giving way to many applications including precision agriculture. This project consists of the development of a greenhouse crop monitoring system based on LPWAN (Low Power Wide Area Network) wireless sensor networks, with a Meta Operating System, which monitors the environmental variables that influence crop growth and development to find suitable growth patterns by implementing machine learning algorithms and that in this way certain processes such as irrigation application are not based on empirical criteria acquired with the experience of the people in charge of greenhouses, but based on the use of technology adequate patterns are obtained for the development of the different crops, in addition, the temperature and other variables present are monitored, so that the development of the crop is improved.

The system can monitor UV rays, CO₂, temperature and humidity that are primary factors in plant growth, and then form a robust database to find suitable growth patterns based on machine learning algorithms.

System tests were carried out by implementing the prototype within the greenhouse which allowed to check the correct functioning of the nodes for crop monitoring, which allows to have the patterns of the growth of the plant and appreciate if it is developing in an appropriate way, the data obtained is collected and processed in the Meta Operating System and the visualization of the information is done through a graphical interface.

CAPÍTULO I. ANTECEDENTES

1.1 Introducción

En el presente capítulo se especifica los parámetros iniciales con los que se va a desarrollar el trabajo de titulación, por lo tanto, se establece el tema, se describe el planteamiento del problema, de la misma manera los objetivos planteados, el alcance que se propone llegar a desarrollar y la justificación respectiva del proyecto.

1.2 Tema

“RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO PARA MONITOREO AMBIENTAL EN INVERNADEROS”

1.3 Problema

La planta de pimiento, durante su ciclo vegetativo, requiere, entre otros, una cierta cantidad de humedad ambiental óptima, del que dependen directamente muchos procesos tales como la transpiración, fecundación, floración y propagación o no de enfermedades. Por otra parte, el suelo también necesita una determinada cantidad de humedad para que las plantas asimilen a través de las raíces los elementos nutritivos. Además, el suelo debe poseer una cierta temperatura, que es variable en cada fase de desarrollo de la planta; el calor del suelo permite que se lleven a cabo funciones vitales para la planta y faciliten el desarrollo de la vida microbiana (Técnico & Agrónomo, 2010). El daño es más frecuente en las hojas jóvenes del centro de la planta. Su acción ocasiona la reducción de la calidad y de la cantidad de fruta. Las plantas con gran infestación se vuelven de color café y mueren. Los áfidos tienden a extenderse rápidamente de un campo a otro transmitiendo una variedad de enfermedades virales (Productores de Hortalizas, 2004).

Actualmente, los invernaderos permiten la producción de pimiento durante todo el año en regiones que presentan condiciones adversas, facilitando la planificación de la producción al controlar las condiciones ambientales internas que pueden acelerar el crecimiento de los cultivos. Como resultado, la cosecha se realiza en menor tiempo ocupando menor espacio de terreno que un cultivo tradicional. En consecuencia, las plantas y los productos están menos expuestos a la contaminación del aire (Fao, 2012). Con la producción agrícola en invernaderos, se mejora la calidad de vida de las familias, a través de la diversificación de los alimentos que consumen y el incremento de su economía generando ingresos por venta de hortalizas y plantas aromáticas. Invertir en agricultura no solo implica destinar el capital, incluye emprender acciones sostenibles que sean manejadas por instituciones que fomenten el desarrollo, investigación e innovación de las capacidades humanas destinadas a este sector.

Los meta sistema operativo (MSO), es un esfuerzo de colaboración para crear un mecanismo robusto de uso general para crear aplicaciones para robótica, ofrece un conjunto básico de software para operar sistemas que se puede ampliar mediante la creación o el uso de paquetes existentes, Con miles de paquetes disponibles por distribución estable, algoritmos encapsuladores, controladores de sensores, etc. (Quigley et al., 2009).

Con un sistema de este tipo, se logra optimizar el proceso de control de temperatura, humedad, gases presentes en el ambiente, entre otros, lo que es idóneo para usarse en un invernadero ya que al tener un alto nivel de respuesta ante los datos enviados por los sensores, debido a que el modo de funcionamiento de MSO permite que los datos de un sensor que desea enviar información en cualquier instante de tiempo no dependan del canal de comunicación si está libre o no, sino que esos datos se añaden como clientes a los datos del sensor que esté operando en ese momento y mediante un proceso de división de tramas, se logra que toda la información de los diferentes sensores llegue al servidor y puedan ser procesados, dando una

respuesta óptima ante cualquier situación que este sucediendo en ese momento en el invernadero.

Por los motivos antes expuestos se ve la necesidad de invertir en tecnología e investigación, en un sistema embebido que determine estados y procesos específicos de cada cultivo plantado dentro de un invernadero, debido a que al usar recursos tecnológicos modernos se puede tener un sistema completamente inalámbrico y con grandes prestaciones. Lastimosamente el Ecuador no es un país desarrollador de tecnología, por lo que es sumamente difícil encontrar sistemas inteligentes que brinden ayuda para solucionar problemas existentes en el sector agrícola.

1.4 Objetivos

1.4.1. Objetivo General.

Implementar una red de sensores inalámbricos bajo tecnología LORA y gestión de procesos para la analítica de datos mediante meta-sistema operativo para monitoreo ambiental en invernaderos.

1.4.2. Objetivos Específicos.

- Realizar un estudio de campo para definir los requerimientos del sistema que permitan seleccionar los sensores y el sistema embebido que cumplan con la función de monitorear variables ambientales de crecimiento del pimiento.
- Proponer un sistema de adquisición y acoplamiento de sensores usando LoRa para la correcta recopilación de datos dentro del invernadero y posterior tratamiento.
- Diseñar un esquema de análisis de datos que permita encontrar patrones adecuados de crecimiento mediante la implementación de algoritmos de aprendizaje de máquina.

1.5 Alcance

Para la obtención de información, se realizará una revisión bibliográfica respecto al modo de configuración de MSO, además del análisis del protocolo LoRa aplicado en las redes de sensores inalámbricos (WSN) para monitoreo en invernaderos.

La tecnología LoRaWAN define el protocolo de comunicación y la arquitectura del sistema para la red, mientras que la capa física LoRa permite el enlace de comunicación de largo alcance y bajo consumo de energía, está considerado para su implementación en invernaderos donde su extensión es muy grande (superior a una hectárea de cultivo), donde tecnología wifi no puede cubrir toda el área de terreno, por lo que LoRa se convierte en un candidato idóneo para aplicaciones en el sector de la agricultura.

Con la finalidad de determinar la mejor herramienta de hardware, se analizará un estudio acerca de diferentes tipos de sensores para monitorear cultivos y que de esta manera se tenga un correcto monitoreo de la planta, además de las tecnologías que permiten tener una conexión inalámbrica entre sensores, que presenten un bajo consumo de energía y un gran alcance, además que puedan ser aplicadas en el entorno rural, específicamente en un invernadero, y se elegirá la que mejor se adapte a las necesidades del proyecto.

La implementación del prototipo se va a realizar con los sensores aptos para la agricultura encontrados en base a requerimientos del proyecto y las respectivas pruebas se las realizará en un invernadero en cual se cultiva pimiento.

Para analizar los procesos en el desarrollo del Proyecto se va a utilizar la metodología en cascada ya que es un modelo lineal que sigue una serie de fases, una a una, sin retorno; Se define el producto, se diseña, desarrolla y se presenta, en ese orden, con lo que se tendrá una correcta manera de ejecutar y evaluar cada paso que se realice en el proyecto.

Para el análisis de datos, se contará con un dispositivo que en base a una red de sensores WSN, recolecte los datos en base a las variables presentes del entorno para luego enviarlas al MSO quien procesara la información recibida y enviará una respuesta acerca del estado de una planta para así determinar parámetros necesarios para que la planta crezca sin ningún inconveniente, toda la información será mostrada en una interfaz gráfica, de modo que el administrador del invernadero pueda estar siempre al tanto del estado de los cultivos por si se necesita algún tipo de tratamiento especial en las plantas.

Al final se obtendrán las respectivas conclusiones y recomendaciones en base a los resultados determinísticos arrojados por las pruebas, para lo cual se tomará distintos datos de la planta de pimiento en el invernadero, y se procederá a analizar las variables presentes y así determinar qué factores se necesitan corregir para que la planta crezca y se desarrolle normalmente.

1.6 Justificación

La realización del presente proyecto viene orientada directamente al Plan Nacional Toda una Vida 2017-2021, en el que dentro del eje 3, específicamente en el Objetivo N.º 7, nos dice que los ciudadanos debemos fomentar una sociedad participativa, enfocándose a un estado al servicio de toda la ciudadanía, en donde la inclusión de tecnología juega un papel muy importante, tanto para el avance productivo como para el económico dentro de la sociedad (Plan Nacional “Toda una Vida”, 2017).

En el Ecuador se requiere desarrollar industrias nacionales, y de esta manera aportar con el crecimiento de la economía del país, se debe disponer criterios de desarrollo para aumentar la producción de los recursos potenciales, en este caso la agricultura, además con el incentivo del Gobierno a impulsar la transformación de la matriz productiva, se propone métodos para

tecnologizar procesos de optimización de producción en invernaderos. (Secretaría Nacional de Planificación y Desarrollo, 2012).

En redes de sensores inalámbricos (WSN), una red de bajo consumo y área extensa LoRa comunica datos a través de grandes áreas, lo que permite eliminar la transmisión de datos mediante saltos, ya que requiere una sincronización muy precisa. Además, evita la centralización ya que cuenta con nodos de procesamiento propio lo que lo hace idóneo para su implementación en invernaderos (Sacaleanu et al., 2019). Lora es la tecnología inalámbrica número uno que disponible el mercado con un bajo costo, alcance de transmisión larga y el consumo de energía óptimo. LoRa proporciona tres configuraciones BW escalables de 125 kHz, 250 kHz y 500 kHz. Por su bajo consumo de energía es preciso para su implementación en lugares donde el suministro eléctrico es nulo ya que las baterías tendrían que recargarse después de un largo periodo de tiempo y se las podría alimentar por algún tipo de energía natural renovable (Vu et al., 2018).

La mayoría de los invernaderos en el Ecuador producen diferentes tipos de productos, unos con varios requerimientos específicos para su desarrollo, pero no se cuenta con sistemas para monitorear los parámetros necesarios para que ese cultivo se desarrolle con total normalidad en base al análisis de datos recolectados por la red de sensores.

Las diferentes técnicas actuales para la monitorización de requerimientos de cultivos presentan una desventaja, la intervención de una persona para revisar todos los cultivos y determinar si su estado en el proceso de desarrollo es correcto, es un método poco habitual. El propósito del proyecto es la interacción entre el agricultor y el sistema, estableciendo parámetros determinísticos sobre parámetros del cultivo para su desarrollo, haciendo un monitoreo mediante la red de sensores se permitirá determinar si todos los factores son los correctos, caso contrario se emitirá una alerta para que se priorice o cambie este determinado

parámetro para que de esta manera la producción del cultivo se realice con los más altos estándares de calidad, contribuyéndose así al desarrollo de negocios más inteligentes y sustentables, operaciones eficientes, mayores ingresos y clientes satisfechos con la calidad del cultivo.

La funcionalidad del dispositivo tendrá lugar en un invernadero tomando varias muestras de los factores que ayudan al desarrollo del cultivo los cuales serán analizados y procesados mediante el servidor MSO para optimizar el procesamiento del sistema, el dispositivo devolverá la información en una forma rápida y precisa, de modo que el usuario podrá determinar que factor necesita ser atendido para el correcto desarrollo del cultivo.

El sistema recolectará parámetros del suelo y el aire, los procesará en MSO enviando una respuesta acerca de los datos de los sensores y se obtendrá el estado del aire y suelo, los sensores usados para la recolección de información serán: sensores de CO₂, humedad, temperatura, rayos UV, cada sensor tomando muestras de los cultivos del invernadero.

CAPÍTULO II. REVISIÓN BIBLIOGRÁFICA

2.1. Introducción.

El presente capítulo, es la base teórica necesaria para el desarrollo de la investigación en la que se abordan los temas para el desarrollo del trabajo de titulación. Se inicia analizando la agricultura inteligente y de precisión, cultivos de invernadero; de la misma manera, las condiciones ambientales necesarias para el desarrollo de una planta de pimiento. Así también, se menciona la recolección de datos mediante una red de sensores para monitoreo ambiental, el tratamiento de los datos y toma de decisiones, además.

Se definen las redes de sensores inalámbricas (por sus siglas en inglés *Wireless Sensor Networks WSN*) para el envío de resultados hacia nodos; así mismo, se describe la comunicación de la WSN mediante el protocolo LoRa, además, se describe un Meta Sistema Operativo (MSO), el cual será usado para la analítica de datos provenientes de los sensores.

Finalmente, se requiere un análisis de plataformas de hardware y software libre para la elaboración del dispositivo que cumpla con los lineamientos establecidos en la investigación y que ayude al ser humano en la toma de decisiones referentes al cultivo de pimiento.

2.2. Agricultura inteligente.

En la actualidad, uno de los principales retos que tienen que enfrentar los países basados en la agricultura, es realizar un cultivo de manera sostenible y eficiente, reduciendo el impacto causado por la contaminación, por esta razón, se convierte en una necesidad primordial el uso de herramientas tecnológicas para la realización de procesos específicos, dando paso a la implementación de la agricultura inteligente con el fin de cumplir con las necesidades que presentan los cultivos.

La agricultura inteligente es la gestión agrícola que, a través del uso de herramientas tecnológicas modernas y avanzadas, tienen como objetivo incrementar la calidad y cantidad de los productos, a través de tecnologías como: Sistema de Posicionamiento Global (GPS), escaneo de suelos, WSN, comunicaciones electrónicas integradas, manejo de datos e internet de las cosas (FAO, 2017).

Si se mide efectivamente los parámetros que contiene un ambiente delimitado y sus variaciones, se puede perfeccionar el uso de insumos y recursos agrícolas por medio de: propagación automática, datos inteligentes, entre otros. En la Figura 1 se muestra un esquema de cómo se implementa tecnología en la agricultura para optimizar procesos en el cultivo mediante: propagación automática, datos inteligentes, monitoreo del campo y condiciones del suelo y la administración del agua.



Figura 1. Agricultura inteligente.

Fuente: (Iglesias, 2018)

2.3. Agricultura de precisión.

La agricultura de precisión es el manejo jerárquico de los cultivos empleando para ello varias herramientas tecnológicas (GPS, Sensores planta-clima-suelo e imágenes multiespectrales provenientes tanto de satélites como de UAS/RPAS), que tienen la capacidad

de detectar el cambio que tiene un determinado asentamiento agrícola (García-Cervigón & José, 2015).

Para implementar la agricultura de precisión, se requiere mucha información detallada, confiable, y específica por sectores, en donde se debe cuantificar e identificar las causas del cambio del agroecosistema ya sea espacial o temporal. Cabe mencionar, no basta con solo medir las variables existentes dentro de un entorno. Además, se debe implementar nuevas prácticas en los procesos de administración de recursos naturales necesarios para el desarrollo de una planta.

2.3.1. Etapas de la agricultura de precisión.

Dentro de las diversas propuestas tecnológicas para el posicionamiento en el campo, los sistemas globales de navegación por satélites (*Global Navigation Satellite System* GNSS, por sus siglas en inglés) son en la actualidad los más populares. Fabricados para el uso militar, la disposición de su uso para civiles transformó la navegación y la medición de distancias y áreas, de manera especial en el medio rural. Su funcionamiento es diferente a los métodos usados por los navegantes en la antigüedad, que consultaban su ubicación basándose en mirar las estrellas, satélites naturales y planetas. Los sistemas actuales utilizan satélites especialmente desarrollados para este, con órbitas perfectamente conocidas y, de acuerdo con las distancias tomadas por los satélites, se determina el posicionamiento del usuario sobre la superficie terrestre (Mantovani & Magdalena, 2014).

En la Figura 2 se aprecia las diferentes etapas que debe seguir la agricultura de precisión en la cual se detalla la recolección de datos, su respectivo procesamiento, el análisis de éstos y la aplicación de insumos como parte de la toma de decisiones. Esto evidencia que la tecnología es parte primordial para efectuar el proceso.

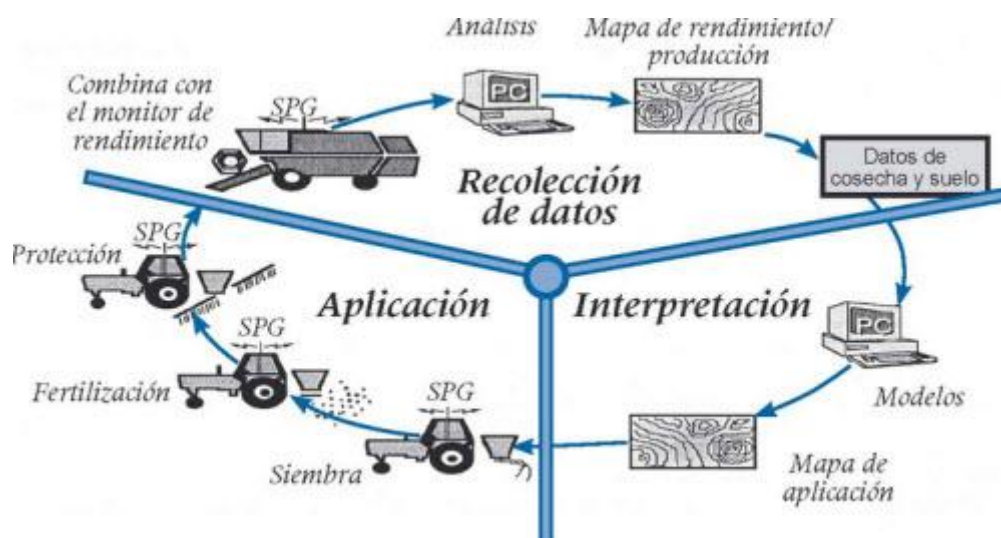


Figura 2. Etapas de la Agricultura de Precisión.

Fuente: (Canamero et al., n.d.).

2.3.2. Herramientas de la agricultura de precisión.

Para facilitar el estudio de la agricultura de precisión, se usan varias herramientas que se muestran en la Tabla 1, las que se plantean en categorías: Sistema de posicionamiento global, monitoreo de rendimiento y mapeo, percepción remota, dispositivos electrónicos, redes de comunicación, sistemas de información geográfica, dosis de fertilizantes y densidad de siembra variable y banderillero satelital.

Tabla 1. Herramientas de la agricultura de precisión

Herramienta	Descripción
Sistema de posicionamiento global (GPS)	Permite localizar y ubicar cualquier punto en tiempo real.
Monitoreo de rendimiento y mapeo	Brinda información de la productividad del campo (rendimiento en ciertas zonas de terreno).
Percepción remota	Se trata de obtener información para su análisis mediante sensores en un área determinada.
Dispositivos electrónicos	Computadores encargados del almacenamiento, procesamiento y representación de información en pantalla.
Redes de comunicación	Parte encargada de transferir la información que han recopilado los sensores.
Sistema de información geográfica (GIS)	Muestra y almacena información espacial de forma estructurada.

<i>Dosis de fertilizantes y densidad de siembra variable</i>	Dosifica los insumos a aplicar según la información en GIS.
<i>Banderillero satelital</i>	Usada para que los equipos sigan una trayectoria mediante GPS.

Fuente: (García & Flego, n.d.).

2.4. Invernadero.

Es una estructura cerrada muy similar a una casa, que generalmente sus paredes y techo se encuentran recubiertas por un plástico o cualquier tipo de material transparente, en el cual se pueden generar condiciones climáticas artificiales reguladas. Esto se realiza para mejorar el desarrollo normal de la planta y cultivar productos de forma eficiente y eficaz tales como: hortalizas, flores, pimiento, tomate, pepino melón y otros (FAO, s.f.). Sin importar la estación del año en que se encuentre, se debe resaltar que tiene otras características favorables como: evitar el exceso de penetración de luz, temperaturas extremas, enfermedades, insectos entre otros.

2.4.1. Clasificación de invernaderos.

Los invernaderos presentan su clasificación con referencia a su tipo de perfil externo, además de su movilidad (Aguado da Costa, 2012).

- *Plano.*

Este tipo de invernadero se utiliza en zonas poco lluviosas como en regiones áridas y secas. Como se muestra en la Figura 3. Los inconvenientes más comunes son el poco volumen de aire, además de una ventilación deficiente, todo esto provoca un peligro de hundimiento por las bolsas de agua de lluvia que se forman en la lámina de plástico que posteriormente puede causar goteo de agua de lluvia sobre las plantas.



Figura 3. Invernadero tipo plano.

Fuente: (Infoagro, 2010).

- ***Capilla.***

Los invernaderos de capilla tienen el techo formando por uno, o dos, planos inclinados, como se muestra en la Figura 4. Dentro de sus ventajas están su fácil construcción y conservación ya que acepta todo tipo de plástico en la cubierta, posee ventilación vertical en paredes muy fácil de colocar con grandes superficies, cuenta con grandes facilidades para evacuar el agua de lluvia.



Figura 4. Invernadero tipo capilla.

Fuente: (HortiCultivos, 2017).

- ***Doble capilla.***

Están formados por naves yuxtapuestas, como se muestra en la Figura 5. El ingreso de aire lo hace por la ventilación cenital en la cumbrera de los dos escalones de la unión de las naves. Tiene un proceso de construcción más dificultoso que los demás, además de ser mucho más cara.

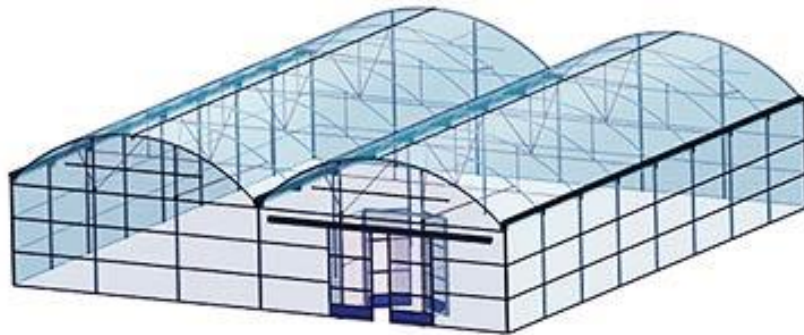


Figura 5. Invernadero tipo doble capilla.

Fuente: (Novagric, 2016).

- ***Diente de sierra.***

Está formado por la unión de varias naves a “un agua “. Estos invernaderos tienen un tipo de ventilación adecuada, ya que a la ventilación normal se une la cenital que tiene por los lados de los dientes de sierra, como se muestra en la Figura 6. En este tipo de invernadero hay que tener previsto la evacuación de las aguas procedentes de la lluvia, para evitar que entre todo el flujo de agua hacia el interior del invernadero.



Figura 6. Invernadero tipo diente de sierra.

Fuente: (Hernandez & Alvarez, 2012).

- ***Parral.***

Se derivan de los antiguos parrales dedicados al cultivo de uva, se componen por dos elementos básicos, una estructura vertical, constituida por soportes interiores (rígidos) y perimetrales; más otra horizontal formada mediante la superposición de dos mallas flexibles de alambre galvanizado. Esta última forma parte de la estructura base y sirve además para soportar y sujetar el material plástico, como se muestra en la Figura 7. Necesitan perforar su cubierta plana para evitar el colapso de la estructura cuando llueve, lo que los hace poco herméticos (eficientes) (Vega, 2014). Los inconvenientes que presenta este tipo de invernadero son el peligro de hundimiento por bolsas de agua, la ventilación es deficiente y la instalación de ventanas cenitales es muy complicada.

No existe un adecuado estanco al agua de lluvia y al aire, lo que provoca una elevada humedad y pérdidas de calor en el interior, entre las principales ventajas presenta un bajo costo, una gran adaptabilidad a la geometría del terreno, una mayor resistencia al viento, aprovecha el agua de lluvia en periodos secos y tiene una gran uniformidad luminosa.



Figura 7. Invernadero tipo parral.

Fuente: (Rufepa, 2020).

- ***Túnel o semicilindro.***

Se caracteriza por la forma de su cubierta, y por su estructura totalmente metálica, como se muestra en la Figura 8. Entre las ventajas de este tipo de estructura están que presenta pocos obstáculos en la estructura, posee una elevada altura que facilita la circulación de aire, presenta una buena estanqueidad a la lluvia y aire, permite la instalación de ventilación cenital, tiene un adecuado reparto de luminosidad y una fácil instalación al ser prefabricados.



Figura 8. Invernadero tipo túnel.

Fuente: (Novagric, 2015).

- ***Por su movilidad.***

Los invernaderos móviles están diseñados para poder cubrir varias superficies de cultivo a lo largo de todo el año, según las necesidades de protección climática y, también, para cambiar el suelo del invernadero a lo largo del tiempo, cambiando de lugar de instalación. Se suelen utilizar macro túneles, de fácil instalación y con plásticos de una sola campaña. Según el material de estructura son de madera, de palos de madera y alambre (parral), metálica (de perfil en ángulo, de tubo circular o cuadrangular, de hierro redondo o cabilla), de hormigón y pilares de plástico reciclado.

2.4. Cultivo de pimiento.

El pimiento es una variedad de hortaliza que al consumirlo proporciona varios beneficios al ser humano principalmente en lo que hace referencia a su salud y a su nutrición. Es uno de los alimentos que más proporciona vitamina C, B y fibra. Ya que es muy beneficioso para el sistema nervioso y cerebral, y además siendo muy rico en antioxidantes y en vitamina A, previniendo enfermedades crónicas y degenerativas, favoreciendo además la secreción gástrica y vesicular y mejorando el estreñimiento (Pinto, 2013).

La planta es de tipo herbácea; que tiene un método de poda para la regeneración y pueda durar muchos años. Su altura puede tener valores entre 0,5 metros de altura en especies que son cultivadas al aire libre y 2 metros en algunas especies cultivadas dentro de un invernadero. El tallo principal al inicio es un fuste erecto que, a una cierta altura (“cruz”), se bifurca en dos o tres tallos hijos que son secundarios; después de brotar muchas hojas, cada uno de estos tallos se bifurca (“cruces”) en muchos nuevos tallos “nietos” y así sucesivamente va desarrollando. Los tallos son muy frágiles y se parten con facilidad (Serrano, 2011).

Su veloz crecimiento de los frutos produce una reducción del crecimiento vegetativo, motivo por el cual, principalmente en aquellos cultivos en los que no se aprecia mucha área

foliar, resulta conveniente sacar la primera producción. Además, se debe deshacerse del primer fruto en los casos de baja luminosidad o temperatura. Si una planta presenta sombreado, esto determina una menor producción de flores y además de una reducción del porcentaje de cuajado (UNICEN, n.d.).

2.4.1. Condiciones ambientales para el desarrollo de una planta de pimiento.

La planta de pimiento, durante su ciclo vegetativo, necesita ciertos parámetros de humedad ambiental óptima, del que necesitan directamente otros procesos tales como la fecundación, floración, transpiración y propagación o no de enfermedades. El suelo, de la misma manera, necesita también de un cierto contenido de humedad para que las plantas asimilen a través de las raíces los elementos nutritivos. De igual forma el suelo debe poseer una determinada temperatura, que puede variar en cada fase de desarrollo de la planta; el calor del suelo describe que se lleven a cabo funciones vitales para la planta y faciliten el desarrollo de la vida microbiana (Reche, 2010). En la Tabla 2, se muestra los parámetros óptimos para cultivar pimiento.

Tabla 2. Parámetros óptimos para el cultivo de pimiento

<i>Parámetro</i>	<i>Valor</i>
<i>Luminosidad</i>	Entre 440 y 680 lux.
<i>Temperatura</i>	Superior a los 15°C.
<i>Humedad relativa</i>	La humedad relativa óptima tiene valores entre el 50 % y el 70 %.
<i>Suelo</i>	Nivel de pH entre 5.5 y 7.
<i>CO₂</i>	El nivel idóneo de CO ₂ para plantaciones de pimiento debe ser de 800 ppm.

2.5.1.1. Luminosidad.

La luz que incide sobre la corteza terrestre puede ser superior a 100000 lux, en donde la fotosíntesis se realiza con una intensidad máxima comprendida entre las longitudes de onda de 440 y 680. La planta de pimiento exige mucha luminosidad, sobre todo en los primeros estados de desarrollo de la planta, además del proceso de floración (ECOAGRICULTOR, 2014).

2.4.1.2. Temperatura.

Los saltos térmicos (diferencia de temperatura entre la máxima diurna y la mínima nocturna) producen desequilibrios vegetativos. La casualidad es que en bajas temperaturas durante el proceso de desarrollo del botón floral (entre 15 y 10° C) genera el crecimiento de flores con alguna de las siguientes anomalías: pétalos sin desarrollar y sin curvar, generación de una diversidad de ovarios que logran desarrollar frutos repartidos alrededor del principal, acortamiento de pistilo y de estambres, engrosamiento de ovario y pistilo, fusión de anteras, etc. Las temperaturas bajas también atrae la generación de frutos de menor tamaño, que logran presentar deformaciones, reducen la viabilidad del polen y favorecen la formación de frutos partenocárpicos (Martínez, 2011).

2.5.1.3. Humedad relativa.

La humedad relativa óptima tiene valores entre el 50 % y el 70 %. Si son muy altas favorecen a la proliferación de enfermedades aéreas y obstaculizan la fecundación. La coincidencia de elevadas temperaturas y reducida humedad relativa logra ocasionar la caída de flores y de frutos recién cuajados (Martínez, 2011).

2.5.1.4. Suelo.

Para cultivar pimiento es preferible los suelos profundos y francos, con un nivel de pH entre 5.5 y 7. No son recomendables los suelos con ineficiente drenaje, que presentan tendencia al anegamiento, ya que la especie es sensible a la asfixia radicular y esto favorece el desarrollo

de enfermedades (César & Álvarez, 2006). Tiene cierta tolerancia a la salinidad tanto agua de riego como el suelo (Pinto, 2013). Los suelos óptimos para el crecimiento del pimiento son los sueltos y arenosos (no arcillosos, ni pesados), profundos, ricos en materia orgánica y sobre todo con un excelente drenaje. Los suelos asfixiantes y encharcadizos favorecen el desarrollo de hongos en raíces y la pudrición consiguiente de éstas.

2.5.1.5. Dióxido de Carbono CO₂.

El CO₂ es un gas imprescindible en la vida de los vegetales para la realización de la fotosíntesis, en donde la mayoría de los vegetales el nivel óptimo de asimilación de CO₂ está comprendido a temperaturas de 18 y 23°C (Cermeño, 2002). Sin una excelente iluminación, no existirá actividad fotosintética adecuada, en donde la tasa de absorción de CO₂ es proporcional a la cantidad de luz recibida, motivo por el cual el medio día se convierte en la hora oportuna para que se dé el enriquecimiento carbónico por los niveles de luminosidad presentes (InfoAgro, n.d.). El nivel idóneo de CO₂ para plantaciones de pimiento debe ser de 800 ppm.

2.5. Adquisición de datos.

Todos los fenómenos que se producen en el mundo siguen varias de las leyes de la física por la cual, se llevan a cabo procesos y fenómenos cambiantes, que no se encuentran en estado estático. Varias de esas características pueden ser percibidas y representan variables en donde el funcionamiento de un sistema depende de eventos en el tiempo que pueden ser: ubicación, magnitud, velocidad, tiempo, condiciones ambientales, entre otros. Por esta razón, el ser humano necesita instrumentos que le permitan entender dicha información ya sea de forma visual, sonido o estímulos físicos.

A estos instrumentos se los conocen comúnmente con el nombre de transductores o sensores que convierten los fenómenos físicos en señales eléctricas. La adquisición de datos consiste en

medir un parámetro físico y transformarlo en voltaje, corriente, temperatura, presión, sonido, entre otros, que generalmente se lo realiza mediante transductores o sensores (National Instruments, n.d.).

2.6.1. Recolección de datos en la agricultura de precisión.

Se usa para obtener información del entorno usando diferentes fuentes, insistiendo en que la información debe estar disponible, la cantidad de los datos es muy elevada y depende del tipo y número de sensores que se encuentren distribuidos en una zona de análisis. Los sensores más usados para estos fines son: humedad del suelo, geolocalización, temperatura, rayos UV, cantidad de CO₂, entre otros, todas estas variables son monitorizadas a través de procesos tecnológicos.

2.6.2. Sensor.

(Ramírez et al., 2014) define un sensor como un transductor el cual convierte un fenómeno físico en una señal eléctrica que se puede medir, el resultado dependerá del tipo del sensor que se esté empleado y éste puede ser: voltaje, corriente y resistencia eléctrica variantes en el tiempo. De acuerdo (Sparkfun, s.f.) en la Tabla 3 se muestra algunos el tipo de sensores y modelo fenómenos físicos y el sensor asociado para obtener su medida.

Tabla 3. Sensores necesarios para monitorear el cultivo de pimiento.

<i>Sensor</i>	<i>Modelo</i>	<i>Fenómeno</i>
<i>UV Light Sensor Breakout</i>	UV715	Luz solar
<i>CO₂</i>	SCD30	Calidad del aire
<i>Humedad</i>	SCD30	Cantidad de agua en el aire
<i>Temperatura</i>	SCD30	Nivel térmico
<i>GPS</i>	SAM-M8Q	Posicionamiento global

2.7. Redes de sensores inalámbricos.

Según (Kodali et al., 2017) las redes WSN son redes conectadas inalámbricamente que consisten en dispositivos distribuidos y espaciados, empleando sensores apropiados para monitorear condiciones ambientales o físicas de un entorno. Los cuales requieren una puerta de enlace que brinda conectividad con una red cableada y los nodos que están distribuidos; al necesitar conexión inalámbrica para unir el resto de los nodos se necesita de un protocolo el cual depende de los requerimientos del sistema y del usuario, mencionándose algunos que pueden ser usados para levantar este tipo de redes: IEEE 802.15.4, LoRa, IEEE 802.11 o enlaces de radio de 900 MHz.

2.7.1. Elementos de las WSN.

Los constantes avances tecnológicos han permitido el surgimiento de nuevos dispositivos llamados nodos. Los cuales están compuestos por microcontrolador, sensores, memoria y algún protocolo estándar de conexión (RS232, Ethernet, etc.). A los nodos se añaden routers, que logran utilizar menores distancias de comunicación y rutas alternativas de entrega de datos, en un acaso se presentan fallas en los nodos. Finalmente existe en cada WSN, un dispositivo que recibe toda la información generada por los nodos y coordina el funcionamiento de la red llamado estación base. En esta máquina (una PC o un sistema embebido), los datos son mostrados, almacenados o enviados a Internet. La Figura 9 muestra los elementos que integran las WSN (Iacono et al., 2012).

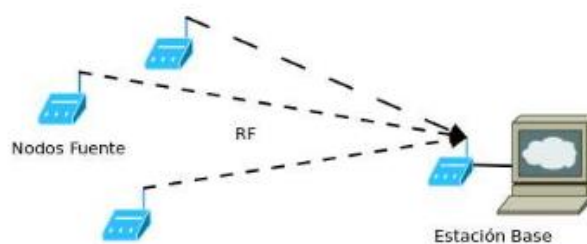


Figura 9. Elementos de una WSN

Fuente: (Iacono et al., 2012).

2.7.2. Funcionamiento.

Los sensores detectan una variación de una magnitud física específica, es decir se encuentran adquiriendo datos. Esta información es enviada al transceptor de cada nodo el cual envía la información al microcontrolador y procede a tratar los datos, una vez realizado el procesamiento, el microcontrolador envía una señal de respuesta al transceptor de cada nodo y finalmente a los equipos terminales (Nayibe et al., 2014). En la Figura 10 se representa una red WSN en donde se muestra sensores, transceptores y el microcontrolador sobre una topología tipo árbol.

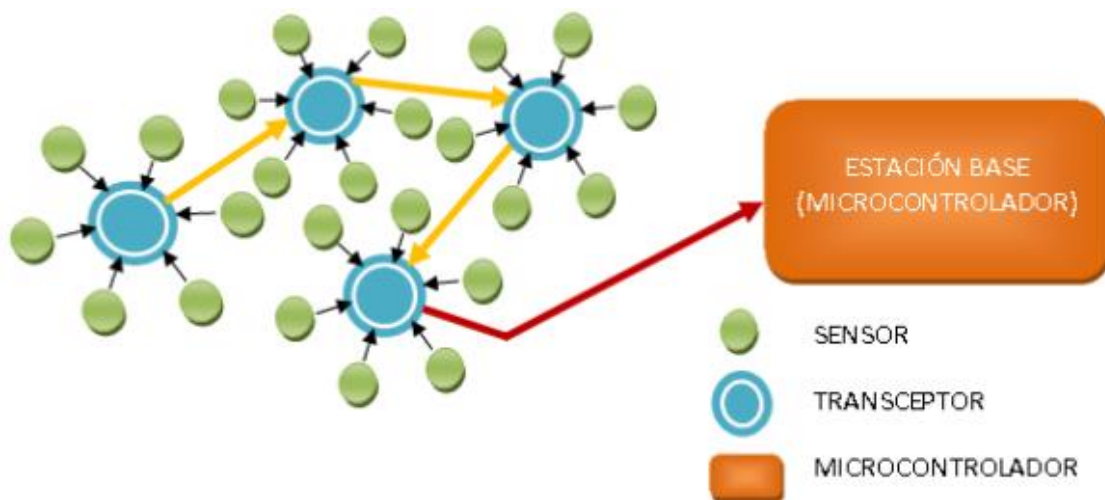


Figura 10. Funcionamiento de una red WSN.

Fuente: (Nayibe et al., 2014).

2.7.3. Arquitectura.

La arquitectura de las WSN al momento de implementar una red es la reutilización de elementos, pero este tipo de redes presentan limitaciones de diseño de hardware por ende se debe considerar la capacidad de las interfaces para transmitir información y que garantice portabilidad. Existen dos tipos de arquitecturas: centralizada en donde los nodos se comunican con el Gateway y distribuida en donde los nodos se comunican con otros nodos dentro de su alcance (Córdoba & Buitrago, 2013).

2.7.4. Topologías de red.

Las redes de sensores inalámbricos se agrupan de diferentes formas en busca de la mejor ubicación entre nodos para el envío y recepción de información. Estas agrupaciones se denominan topologías de la red, algunas de éstas se pueden ver en la Figura 11 (Nayibe et al., 2014).

Los nodos WSN se organizan en tres tipos de topología de red: estrella, árbol y tipo malla; el dispositivo representado por el círculo de color negro se denomina Gateway y el resto son los nodos que transmiten información dependiendo de la topología en la que se encuentren configurados.

- ***Estrella***

La topología Tipo Estrella, especifica que: cada uno de los nodos que constituyen la red, se conectan a un solo dispositivo central ya sea un conmutador, computadora o puerta de enlace. El dispositivo central es considerado el servidor, y cada uno de los nodos son considerados clientes (Wordpress, 2017).

- ***Árbol***

La topología Tipo Árbol es una topología que trabaja del mismo modo que la topología tipo estrella y la topología en malla, ya que cada uno de los nodos, se conectan a un nodo de mayor categoría, para luego llegar a la comunicación a la puerta de enlace, en otras palabras, los nodos más comunes se comunican a sus nodos principales que envían sus datos a la puerta de enlace (Mora, 2019).

- ***Malla***

La topología Tipo Malla, es una topología de red en la que cada dispositivo o equipo están conectados a todos los equipos de la red entre sí. Una de sus características principales es

que cada uno de los mensajes de un nodo hacia otro, puede llegar por distintos caminos, pero si la topología de red tipo malla está completamente conectada, no puede existir ninguna interrupción en la comunicación.

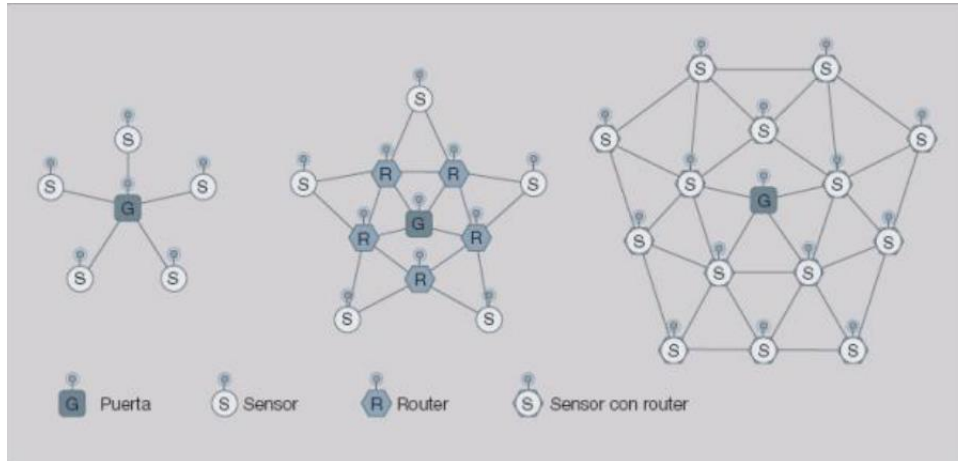


Figura 11. Topologías de red WSN.

Fuente: (Nayibe et al., 2014).

2.8. LoRa.

LoRa (por sus siglas en inglés *Long Range*), es una técnica patentada de modulación de espectro expandido de la empresa Semtech. Utiliza la técnica de modulación de radio basada en *Chirp Spread Spectrum* (CSS) y trabaja en las bandas ISM (de las siglas en inglés, *Industrial, Scientific, and Medical*), por debajo de 1GHz (433, 868 y 915MHz según la versión del microchip a usar). La capa física de LoRa se puede usar con cualquier capa MAC; sin embargo, LoRaWAN es propuesto para MAC actualmente ya que opera una red en una topología en estrella simple (SEMTECH, n.d.).

2.8.1. LoRaWAN.

Según (De Poorter et al., 2017), LoRaWAN es la capa de acceso al medio creada por LoRa Alliance, define el propio protocolo de comunicación MAC y la arquitectura del sistema para la red. Las redes LoRaWAN utilizan típicamente la topología de red en estrella donde los Gateways transmiten los paquetes entre los dispositivos finales y un servidor de red central.

Por otra parte, el protocolo y arquitectura LoRaWAN, determinan la vida útil de la batería de un nodo, la capacidad de red, la calidad de servicio y la diversidad de aplicaciones de la red; además admiten comunicaciones bidireccionales de bajo costo, móviles y seguras para Internet de las cosas (por sus siglas en inglés *Internet of Things* IoT), diseñado para garantizar la fiabilidad y seguridad (Pérez, 2017).

2.8.1.1. Arquitectura de una red LoRaWAN.

La arquitectura de LoRa está dirigida al esquema de la Figura 12, contiene 4 capas estructurados de la siguiente forma:

- Capa Física: Conformada por los nodos finales o nodos sensores que recolectan la información del medio.
- Capa Enlace de Datos: Conformada por la puerta de enlace que recibe la información de todos los nodos implementados en la red.
- Capa de Red: Conformada por los medios de transmisión como tecnologías 3G, Ethernet, Wifi etc. Comunica la capa de enlace de datos con la capa aplicación o la nube.
- Capa Aplicación: Dirigida a los servicios o aplicaciones en la Nube, almacenamiento, presentación, seguridad y tratamiento de la información mediante el uso de protocolos TCP/IP (Pérez, 2017).

Cada nodo perteneciente a la arquitectura de red LoRa, sin importar el número de dispositivos, se conectan mediante radio frecuencia a la puerta de enlace, la cual es la encargada de administrar cada dato que llegue del equipo terminal, la puerta de enlace puede conectarse a través de la capa de red a diferentes servicios de almacenamiento, y tratamiento de datos en la capa aplicación, para su manipulación y presentación (Pérez, 2017). Los dispositivos permanecerán en reposo, hasta que reciban una señal de parte de los sensores conectados activando el proceso de envío de datos a la puerta de enlace.

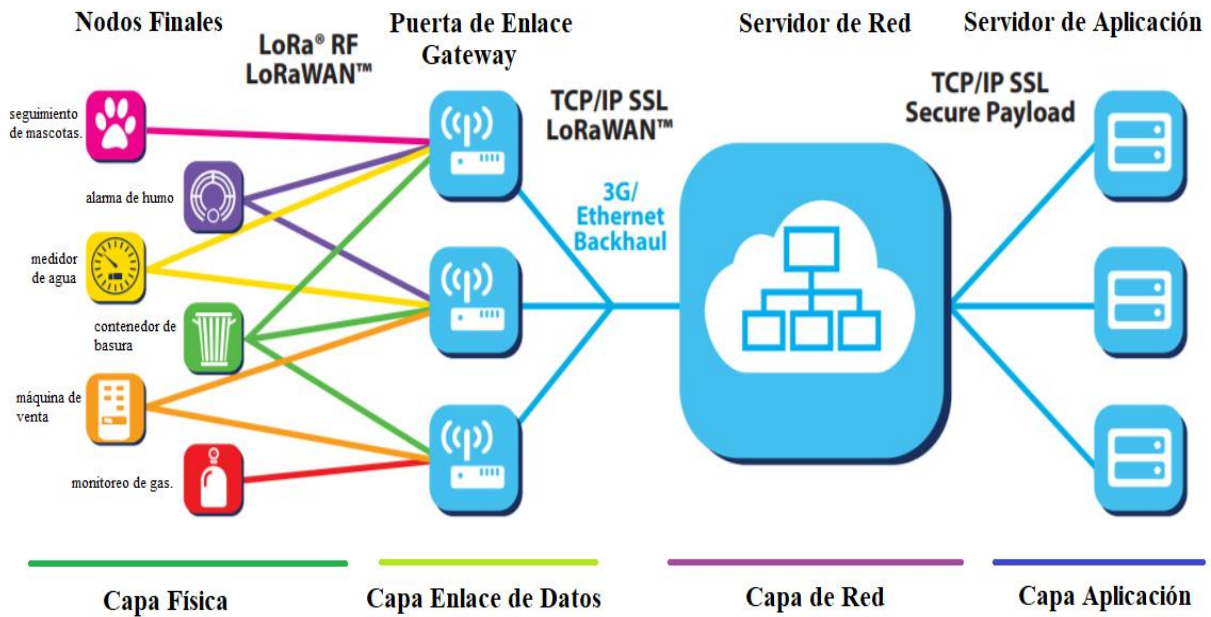


Figura 12. Arquitectura básica de LoRa y LoRaWAN

Fuente: (LoRa Alliance, 2015).

2.8.1.1. Canales y rango de frecuencias.

Las tecnologías LoRa, trabajan en diferentes rangos de frecuencia, dependiendo de la zona geográfica en la que se encuentre. Según (García, 2017) las bandas de frecuencia en Europa es la banda ISM de 863- 870 MHz, aplicada la regulación por el Instituto Europeo de Normas de Telecomunicaciones conocida por sus siglas en inglés ETSI. Dicha frecuencia utiliza 8 canales que el dispositivo los elije de forma arbitraria. En América del Norte, América del SUR, Australia, Singapur e Israel, utiliza la banda ISM de 902-928 MHz, en los que pueden trabajar con 13 canales y un ancho de banda de 2.16MHz por cada canal utilizado, datos que se puede apreciar en la Tabla 4.

Dichos canales se basaron en los que utiliza la tecnología de Radio frecuencia xBee de 900 MHz.

Tabla 4. Rango de frecuencias de Lora WAN y tecnologías similares

Característica	Weightless -N	Weightless -P	Sigfox	LoRaWAN	RPMA
Banda de frecuencia	EU: 868Mhz USA: 900Mhz	Ghz ISM	EU: 868 Mhz USA: 902 Mhz	EU: 433/868 Mhz USA: 912 Mhz	2.4 Ghz
Rango estimado	3 KM (Urbano)	2 KM (Urbano)	3-8km (U) 30-50km(R)	3-8km (U) 30-50km(R)	1-3km (U) 5-10km(R)
Enlace propuesto(dB)			156dB	156dB	172dB
Throughput	10 bps – 10kbps	200bps- 100kbps	EU: 100bps USA: 600bps más de 140 mensajes por día	0.3 – 50kbps adaptable	Uplink: 100kbps Downlink: 600kpbs
Consumo de energía			Tx: < 50 mA Rx: 10-40 mA Stand By: < 0.01 mA	Tx: < 50 mA Rx: 10-40 mA Stand By: < 0.01 mA	Tx: 750 mA Rx: 300 mA Stand By: < 0.072 mA
Tamaño del paquete	20 bytes en adelante	Mínimo 10 bytes	12 bytes	Definida por el usuario	6 bytes a 10kbps
Dispositivos por puntos de acceso	Ilimitado	Ilimitado	1 millón	1 millón	< 500.000
Seguridad	Encriptación AES-128	Encriptación AES-128/256	N/A	Encriptación y autenticación AES CCM (128-bits)	Encriptación 128/256
Estado	Desplegado	Bajo despliegue	En despliegue	En despliegue	En despliegue

Fuente: UPC. Obtenida de: <https://www.upc.edu/es>

En base a la información establecida en la Tabla 4 se obtiene los datos de las tecnologías dirigidas a IoT, en la Tabla 5 se especifica las características de LoRaWAN y Sigfox aplicables a Ecuador y el continente latinoamericano.

Tabla 5. Rango de Frecuencias de LoRa WAN y tecnologías similares aplicables a Ecuador

Característica	Sigfox	LoRaWAN
Banda de frecuencia	902 Mhz	912 Mhz

Rango estimado	3-8km (Urbano) 30-50km (Rural)	3-8km (Urbano) 30-50km (Rural)
Enlace propuesto(dB)	156dB	156dB
Rendimiento	600bps más de 140 mensajes por día	0.3 – 50kbps adaptable
Consumo de energía	Tx: < 50 mA Rx: 10-40 mA Stand By: < 0.01 mA	Tx: < 50 mA Rx: 10-40 mA Stand By: < 0.01 mA
Tamaño del paquete	12 bytes	Definida por el usuario
Dispositivos por puntos de acceso	1 millón	1 millón
Seguridad	N/A	Encriptación y autenticación AES CCM (128-bits)
Estado	En despliegue	En despliegue

Fuente: UPS. Recuperado de: <https://dspace.ups.edu.ec/bitstream/123456789/16897/1/UPS-ST003910.pdf>

2.8.1.2. Clases de dispositivos LoRaWAN.

Entre los objetivos principales de LoRaWAN, se encuentra, optimizar el consumo de energía. Por lo cual, en los dispositivos finales se crean ventanas de tiempo específicas para recibir datos, haciendo mucho menor el consumo de energía. A continuación, se muestran las tres clases que existen:

- Clase A

Los nodos de Clase A se caracterizan por consumir la menor cantidad de energía, sirven para sistemas en los que los dispositivos no deben recibir datos continuamente. También, dichos nodos solo permiten la comunicación desde el Servidor, por medio de dos ventanas de recepción, que, al hacer la petición, las ventanas se abren por un ligero periodo de tiempo después de una transmisión ascendente, así el dispositivo recibirá un paquete de retorno en el que contendrá un asentimiento (por sus siglas en inglés *Acknowledgement* ACK) del dato enviado. Mediante dicha comunicación el servidor deberá esperar hasta la próxima transmisión de enlace ascendente para poder enviar otro dato (Bustos, 2018).

- Clase B

Los nodos de Clase B contienen las mismas características de clase A, pero contienen la capacidad de recibir datos sin la necesidad de enviar un paquete, es decir, permiten a cada nodo estar sincronizados, con la puerta de enlace, de tal forma que puedan negociar tiempos de recepción de paquetes en sentido puerta de enlace-nodo. Con este proceso, las ventanas de recepción se abren en periodos de tiempo acordados con la puerta de enlace principal (Bustos, 2018).

- Clase C

Los nodos de Clase C son los equipos que mayor consumo energético tienen, entre sus dos clases antecesoras, ya que dichos dispositivos, permanecen escuchando todo el tiempo, proporcionando los mejores tiempos de respuesta y capacidad de envío de la puerta de enlace a los nodos finales. Se entiende que, en este caso, las ventanas de recepción se encuentran abiertas (excepto cuando transmiten) (Bustos, 2018).

2.8.1.3. Estructura del paquete.

Para definir la tasa de bits (RS) se necesita un determinado factor de propagación (FP) y el ancho de banda (BW) expresando el resultado en la siguiente ecuación.

$$RS = \frac{BW}{2^{SF}} \quad (\text{Ecuación 1})$$

Donde:

- RS= Velocidad de símbolo
- BW= Ancho de Banda
- SF= Factor de propagación (Ertürk et al., 2019).

La estructura del paquete LoRa se muestra en la Figura 13. Un paquete comienza con el preámbulo, programable de 6 a 65535 símbolos, al cual la radio agrega 4,25 símbolos.

Posteriormente, sigue un encabezado opcional, que describe la longitud y la tasa de FEC (por sus siglas en inglés *Forward Error Correction*) de la carga útil, e indica la presencia de un CRC opcional de 16 bits para la carga útil. El encabezado siempre se transmite con una tasa de FEC de 4/8 y tiene su propio CRC. Después del encabezado opcional, está la carga útil, que puede contener de 1 a 255 bytes. Al final de la carga útil, se puede incluir un CRC opcional de 16 bits (Bor et al., 2016).

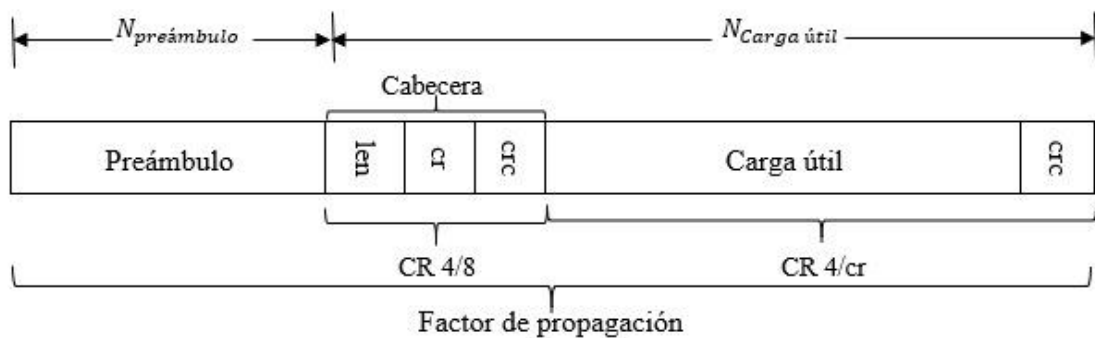


Figura 13. Estructura del paquete LoRa.

Fuente: (Bor et al., 2016).

Editado: Gordillo, Carlos

- **Preámbulo:** Una secuencia de chirps (variación de frecuencia) de tamaño variable que se utiliza para sincronización de tiempo y frecuencia.
- **Símbolos de sincronización de trama:** Dos chirps modulados cuyo valor puede usarse como identificador de red. Un transceptor LoRa de hardware soltará cuadros que contengan un símbolo de sincronización que no coincida con un valor preconfigurado.
- **Símbolos de sincronización de frecuencia:** Dos chips conjugados seguidos por un chirp con duración de $\frac{T}{4}$, que se pueden usar para sincronización de frecuencia.
- **Cabecera (Opcional):** Campo que contiene la longitud de la carga útil, la velocidad de datos utilizada, un bit que indica la presencia de una verificación de redundancia cíclica

(CRC) de la carga útil y una suma de verificación de encabezado de 1 byte. Un CR de 4 siempre se usa en combinación con el modo de velocidad reducida para el encabezado. El encabezado puede transmitirse explícitamente (modo explícito) o dejarse fuera de la trama (modo implícito). En el último caso, el transmisor y el receptor deben configurar la velocidad de codificación y el bit de presencia CRC de antemano.

- Carga útil: Campo de longitud variable que contiene los datos de capa de Control de acceso medio (MAC) transmitidos y un CRC de 2 bytes de estos datos (Robyns et al., 2018).

2.8.1.4. Capacidad de red.

Según (Moya, 2018), la capacidad de una topología en estrella para recibir mensajes de manera eficiente de los nodos finales de largo alcance depende específicamente de la capacidad del Gateway. Por lo cual para obtener una alta capacidad en la red LoRaWAN, se debe lograr una adaptación de la velocidad de datos (ADR) y usar un transceptor multicanal (multi-módem) en el Gateway para recibir mensajes simultáneos en múltiples canales. Algunas de las causas que pueden afectar a la capacidad de la red son el número de canales presentes, velocidad de datos (data rate), la longitud de la carga útil (payload) y la frecuencia con la que transmitan los nodos.

2.8.1.5. Factor de dispersión.

El factor de dispersión (SF) es la relación entre la velocidad de símbolo y la velocidad de chip. Un factor de dispersión más alto aumenta la relación señal / ruido (SNR) y, por lo tanto, la sensibilidad y el rango, pero también aumenta el tiempo de emisión del paquete. El número de chips por símbolo se calcula como 2^{sf} . Por ejemplo, con un SF de 12 (SF12) se usan 4096 chips / símbolos. Cada aumento en SF reduce a la mitad la velocidad de transmisión y, por lo tanto, duplica la duración de la transmisión y, en última instancia, el consumo de energía. El factor de dispersión se puede seleccionar de 6 a 12. SF6, con la transmisión de mayor velocidad,

es un caso especial y requiere operaciones especiales. Por ejemplo, se requieren encabezados implícitos. Las comunicaciones de radio con diferentes SF son ortogonales entre sí y es posible la separación de la red utilizando diferentes SF (M. Bor et al., 2016).

2.8.1.6. Tiempo en el aire de la transmisión.

El tiempo aire de una transmisión LoRa depende, además del tamaño de la carga útil, de la combinación de SF, BW y CR. La duración de una transmisión se puede calcular con la calculadora de módem Semtech LoRa. Debe notarse que, dependiendo de la configuración de comunicación seleccionada, un paquete de datos puede tener variaciones significativas en el tiempo de aire. Por ejemplo, un paquete de 20 bytes puede variar entre 9 ms y 2.2 s. Por lo tanto, la selección de parámetros de comunicación tiene un tremendo impacto en la escalabilidad de una implementación de LoRa (M. C. Bor et al., 2016).

2.8.1.7. Seguridad LoRaWAN.

LoRaWAN, trabaja mediante dos capas de seguridad, una vinculada a la verificación de autenticidad del nodo que envía los datos, y otra, dirigida al usuario y el acceso que tenga a los datos almacenados.

Según (Gaybor Murillo & Maridueña Chunga, 2018), se utiliza encriptaciones AES (Estándar de Encriptación Avanzado), mediante el identificador IEE EUI64, los cuales son globalmente únicos, haciendo de LoRaWAN una red de fiar.

LoRaWAN, incluye un esquema de corrección de errores variable ante posibles pérdidas, el cual mejora la robustez, de la señal emitida con redundancia.

Al usar la modulación de Chirp Spread Spectrum, se vuelve robusto, ante mecanismos de degradación de canales, ya sea atenuación, desvanecimiento, creación de múltiples trayectos y el Efecto Doppler (Gaybor Murillo & Maridueña Chunga, 2018).

2.9. Meta sistema operativo.

La robótica ha experimentado muchos avances en los últimos años, con los robots adquiriendo nuevas habilidades y la capacidad de emprender nuevas tareas. Sin embargo, el uso de robots no se ha generalizado tanto como lo han predicho muchas personas, incluso el desarrollo de robots industriales se ha estancado.

La falta de una plataforma estándar ha impedido con frecuencia que diferentes tipos de software robótico puedan comunicarse entre sí, lo que ha limitado la capacidad de los desarrolladores de usar múltiples aplicaciones dentro de un solo robot. Esto también ha evitado que los ingenieros puedan reutilizar el software y colaborar de otra manera. Sin embargo, todo esto puede estar a punto de cambiar, ya que un número cada vez mayor de empresas e investigadores están trabajando con la plataforma de código abierto Meta Sistema Operativo (MSO) no es solo un sistema operativo, aunque incluye algunas funciones. También es un marco que proporciona módulos de software para realizar actividades en diferentes zonas geográficas.

La tecnología también ayuda a controlar y coordinar los módulos. Un MSO es un marco de software que simplifica la tarea de escribir software complejo. Con un MSO, es posible obtener rápidamente software de última generación para muchos aspectos del sistema. Los defensores esperan que los MSO permitan a los desarrolladores crear de manera flexible y económica sistemas más capaces. Sin embargo, la tecnología enfrenta desafíos antes de convertirse en la plataforma robótica dominante (Garber, 2013).

2.9.1. Funciones de un MSO.

Un MSO debe cumplir con funciones y operaciones específicas acordes a la aplicación a la que vaya a ser sometido, concretamente, permite utilizar ordenadamente los diferentes recursos

que ofrecen cada uno de los nodos secundarios, siendo estos sujetos a las órdenes de un nodo principal, es decir un control centralizado.

De esta manera, este sistema de computación distribuida por nodos, puede estar conformada por varios dispositivos con altas capacidades de procesamiento, así como también de dispositivos de bajos recursos, cubriendo distintas arquitecturas que van desde sistemas embebidos simples hasta supercomputadores.

Las funciones que un MSO proporciona van acorde a la aplicación a la que vaya a ser sometido, en la que la capacidad de balanceo de procesamiento y recursos de sistema viene a ser el punto principal, ya que esto permite que un nodo maestro asigne o reasigne recursos a sus nodos secundarios, haciendo que la disponibilidad de la información que maneja el sistema en general aumente.

Un MSO permite la disminución de costos debido a que ya no es necesario que cada nodo posea un sistema complejo de procesamiento de sus datos, debido a que solo envían la información para que pueda ser procesada por el nodo principal, y así se crea un sistema centralizado en el cual los dispositivos secundarios no procesan ningún tipo de información, tan solo cumplen órdenes (Bottrighi & Terenziani, 2016).

Cuando se habla de un SO, viene de la mano sus procesos y funciones a ser ejecutadas, en donde cada una de ellas viene a ser fundamental para su consecuente permitiendo así elaborar una sucesión de procedimientos que posibilitan la culminación de un objetivo planteado por el SO.

Acorde a lo mencionado anteriormente, los Sistemas Operativos cuentan con una preconfiguración inicial que permite llevar a cabo todas sus tareas, es por eso que la unidad central de procesamiento es el ente principal que distribuye cada una de esas tareas a los entes

secundarios, formando así un conjunto de características que ayudan a cumplir las necesidades para las que fue diseñado.

En base a Hsu (2017), se detallan a continuación las funciones principales que un sistema operativo debe cumplir para ejecutar una cadena de tareas:

- **Gestión de Procesos**

La gestión de procesos en la CPU es la parte más importante para proporcionar los recursos necesarios a un programa que debe ejecutarse cuando se necesita el uso de RAM, el tiempo de ejecución de la CPU y el acceso a archivos para controlar un proceso en particular.

El Sistema Operativo se encarga de destruir, crear, iniciar y detener de manera uniforme los procesos, de acuerdo con una prioridad especificada, permitiendo realizar una secuencia de acciones de comunicación entre procesos. La línea de acción prioritaria antes mencionada es tomada de manera arbitraria por el sistema, en ocasiones por causa de la no ejecución de procesos no críticos.

Es importante señalar que el Sistema Operativo proporciona habilidades de concurrencia cuando se trata de asignar procesos, lo que hace posible ejecutar múltiples subprocesos de manera ordenada o parcialmente desordenada sin afectar directamente el resultado general y el propósito de ese proceso.

- **Gestión de la Memoria**

Por otra parte, la memoria principal del sistema es otro de los componentes fundamentales para llevar a cabo una tarea en el SO, la que consiste en una recopilación de datos que es compartida en conjunto con la unidad central de procesamiento, y permite a cada proceso asignarle un espacio de memoria volátil adecuado para cumplir con todos sus requerimientos, obteniendo de esta manera un recurso compartido asignable de manera

dinámica y aprovechable, sin desperdiciar así espacios de memoria innecesarios y permitiendo una eliminación instantánea de información no requerida.

- **Hilos de ejecución**

Los subprocesos de ejecución en las funciones del sistema operativo son aquellos que componen el proceso común, es decir, son los subprocesos que ocupan espacio de memoria y recursos de procesamiento para ser ejecutados, por lo que un proceso común se compone de uno o más subprocesos de ejecución dependiendo de su aplicación.

De esta forma, un hilo de ejecución puede estar en diferentes estados, bloqueado, listo o activo durante un tiempo determinado, lo que permite que un hilo componga una tarea para ser ejecutada en diferentes procesadores del sistema, según sea el caso. ser. puede ser apropiado, lo que permite que las arquitecturas de multiprocesamiento actuales aprovechen al máximo y destacan las características de concurrencia de trabajo.

- **Gestión del Almacenamiento Secundario**

Una vez que se ha completado un proceso genérico a través de sus cadenas de ejecución, es necesario tener en cuenta la asignación de almacenamiento no volátil para el cual se ha desarrollado el proceso, permitiendo que se recopile la información especificada, predeterminada, con el fin de obtener una base de datos que usted puede llamar a determinados archivos que ya han sido creados y así acelerar el proceso de desarrollo, evitando ejecuciones innecesarias y repetitivas.

Cabe destacar que este tipo de almacenamiento se utiliza para recopilar información del usuario que no es necesaria para la ejecución de los procesos del sistema.

- **Administración del sistema de Entrada/Salida (I/O)**

La gestión de los distintos puertos de entrada y salida del sistema permite la interacción entre el hardware y el software, ya que gracias a controladores específicos es posible interactuar en el sistema a través de dispositivos externos diferentes como antena, monitor, teclado, entre otros.

La administración de los recursos de entrada y salida del sistema operativo es esencial cuando se conectan diferentes dispositivos de hardware al software, lo que permite escalar el sistema informático de acuerdo con las necesidades y objetivos del usuario de una manera razonable.

2.9.2. Principios del meta sistema operativo.

2.9.1.1. Comunicación y Middleware.

Para hacer que el sistema sea más práctico, genérico, modular y extensible, el sistema está construido en el middleware, que realiza la comunicación entre diferentes nodos en el sistema a través de protocolos TCP / IP. En realidad, un MSO es un sistema operativo de código abierto, que puede proporcionar los servicios que incluyen abstracción de hardware, control de dispositivos de bajo nivel, formato de datos de conmutación estándar entre procedimientos y gestión de paquetes. Como los robots y agentes en el sistema de servicio generalmente están equipados con un dispositivo de bajo nivel y algunos sensores, la selección de MSO puede resolver estos problemas por sus características distribuidas y genéricas. Además, mediante estas ventajas MSO y con los paquetes del MSO, el sistema puede ser altamente modular y fácil de extender.

Un MSO proporciona dos métodos principales para comunicarse entre procedimientos: Servicio y Tópico. En realidad, la conexión del tópico es mucho más compleja que el servicio.

Aquí para simplificar, la comunicación entre los nodos del MSO se puede describir como en la Fig. 14 (Ma et al., 2018).

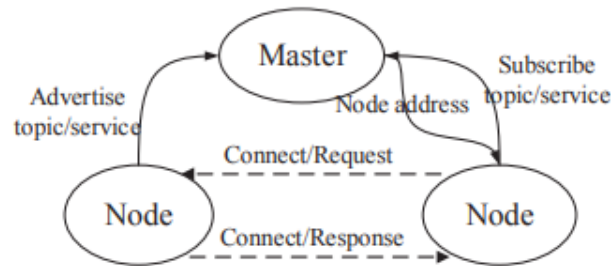


Figura 14. Comunicación de nodos

Fuente: (Ma et al., 2018).

2.9.1.2. Operación del sistema distribuido.

- Paso 1: La capa de servicio de red del sistema inicia el servicio. y cambia al estado de escucha. El módulo de administración de usuarios es responsable de monitorear la situación de inicio de sesión del usuario remoto. Los nodos y el módulo de gestión de servicios son responsables de la situación de registro en el nodo inteligente, incluido el robot en el entorno de la red de área local inalámbrica de interior.
- Paso 2: Si el nodo inteligente interior se registra con éxito, el nodo y el módulo de gestión de servicios renovarían la lista de servicios en la base de datos. Si un usuario remoto inicia sesión en ese momento, el módulo de administración del usuario autenticará al usuario. Si el usuario pasa la verificación, el sistema le otorgará al usuario la autoridad de operación entre pares.
- Paso 3: El usuario que tiene la autoridad de operación puede seleccionar el modo de control y dispensar las tareas a través de la interfaz hombre máquina. La interfaz interactiva proporciona el menú de usuarios para una variedad de opciones de control, los usuarios pueden seleccionar el mejor modo para dar instrucciones de tareas simplemente a través de la interfaz.

- Paso 4: Después de la distribución de las tareas, el módulo de gestión de tareas es responsable de descomponer las tareas complejas en un servicio ejecutable y combinar el servicio. Llama al servicio correspondiente desde el robot a través de los nodos y el módulo de gestión de servicios. Durante este momento, el servidor rechazará las tareas indisponibles directamente.
- Paso 5: el nodo recibe la solicitud de servicio para comenzar a programar el nodo del controlador y realizar las tareas.
- Paso 6: la información de estado y la información de retroalimentación en el proceso en ejecución se envían a los nodos y al módulo de gestión de servicios. Luego la información también se entregará a usuarios remotos. Sin embargo, la información de monitoreo de video que se envía a los usuarios remotos proviene de la cámara del robot a través del módulo de administración de video en la capa de servicio de red (Ma et al., 2018).

2.9.3. Estructura de un MSO.

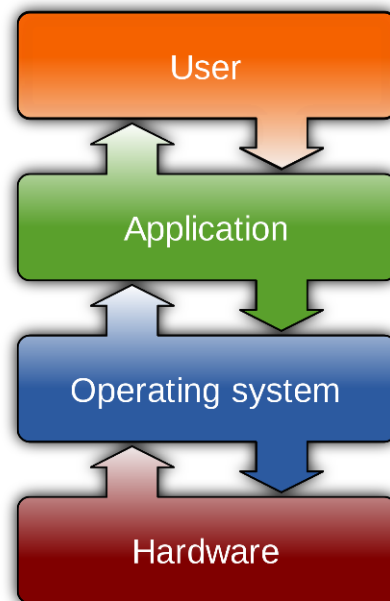


Figura 15. Estructura simple de un MSO.

Fuente: (Arpaci & Remzi, 2016).

Considerando la Figura 15, un sistema operativo (SO) definido por software que funciona en conjunto con el hardware del dispositivo inteligente, lo que ayuda a coordinar y administrar todos los recursos y acciones que el usuario realiza en un momento específico, siendo el eje principal y de conexión de los procesos y acciones de un sistema al hardware.

Es importante señalar que la estructura de un meta sistema operativo y un sistema operativo simple tiene la misma jerarquía y asignación de procesos, pero la diferencia está principalmente basada en hardware, donde en un sistema operativo, se ubica geográficamente en un punto en particular. es decir, se asignan tareas en un mismo lugar tanto para hardware como para software, mientras que en MSO el hardware se distribuye en diferentes áreas geográficas que realizan funciones de recolección de datos que serán enviados posteriormente, para ser procesados por el nodo maestro, que contiene mayores recursos de hardware y software.

De esta forma, el MSO consta de dispositivos que permiten recolectar los datos o información necesarios, que serán enviados y recibidos por un sistema central de análisis de información (SO), que procesará todos los datos. El usuario final puede ver y comprender la aplicación.

2.9.4. Arquitectura del sistema.

El marco del sistema de robot móvil en red está diseñado con los requisitos funcionales del sistema. El sistema consta de tres capas de estructura de red, que se compone de la capa de cliente web, la capa de servicio de red y la capa de control local del robot. La estructura y las funciones de cada capa son las siguientes.

- Capa de cliente web: Esta capa proporciona principalmente una interfaz interactiva de control y envío de tareas para usuarios remotos y monitorea continuamente la ejecución de tareas y el funcionamiento del equipo robótico a través de la interfaz hombre máquina.

- La capa de servicio de red: esta capa es el núcleo del sistema, incluido el servidor web y el servidor de video. La función principal de la capa es recibir instrucciones de la tarea de la capa del cliente web y luego llevar a cabo el análisis, la planificación y la descomposición de la tarea. Los resultados se enviarán a la capa de control local del robot o se podría llamar directamente al servicio correspondiente. Al mismo tiempo, también necesita recibir la información de retroalimentación y la información de estado, que se enviarán a la capa del cliente web para su supervisión.
- La capa de control local del robot: Esta capa se ejecuta principalmente en el controlador del robot móvil que se comunica con nodos inteligentes ambientales. El sistema operativo de robot de código abierto MSO se utiliza para integrar los recursos auxiliares en entornos interiores. El robot móvil y sus recursos asociados proporcionan lógicamente una variedad de nodos, incluidos nodos de computación complejos, nodos de implementación física, nodos de conocimiento de datos y nodos de detección dinámica, que proporcionan los servicios básicos que incluyen navegación autónoma y evitación de obstáculos, exploración autónoma y construcción de mapas en entornos desconocidos. En el proceso de implementación del sistema, la capa es la principal responsable del registro de los nodos en línea, el envío de información como retroalimentación a la capa superior, como el estado de ejecución de la tarea, los datos de operación del equipo y el flujo de video de monitoreo etc. (Garber, 2013).

2.10. Hardware libre.

El hardware libre es un enfoque de desarrollo en el cual se crean dispositivos “cuyo código fuente, especificación de procesos de fabricación y diseño conceptual están disponibles de forma tal que ofrezcan: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las mejoras” (Plataforma de desarrollo en hardware libre, 2012), por ende, se puede acceder a los planos respectivos con el fin de participar en la evolución de la tecnología.

Con hardware libre, se tiene libertad de uso, estudio y modificación, distribución y redistribución de mejoras.

2.10.1. Sistemas embebidos.

Un Sistema Embebido (ES) es un sistema de propósito específico en el cual, el computador es encapsulado completamente por el dispositivo que él controla. A diferencia de los computadores de propósito general, los sistemas embebidos son diseñados para una aplicación específica, es decir, estos sistemas realizan un grupo de funciones previamente definidas y una vez el sistema es diseñado, no se puede cambiar fácilmente su funcionalidad.

Debido a su interacción con el entorno deben cumplir restricciones temporales estrictas, el término sistemas de tiempo real es utilizado para enfatizar este aspecto; son heterogéneos, es decir, están compuestos por componentes de hardware y además tienen grandes requerimientos en términos de confiabilidad (Camargo Bareño, 2011).

2.10.2. Plataformas de código abierto.

Cuando se habla de plataformas de código abierto, se hace referencia a desarrollos que trabajan para desarrollar nuevas ideas dentro de una comunidad para aportar a las bases del conocimiento; actualmente se cuenta con algunas plataformas que permiten implementar proyectos a nivel básico como nivel avanzado, las plataformas más conocidas en el mercado para el desarrollo de dispositivos electrónicos son: Raspberry Pi, Arduino, Sparkfun.

- **Raspberry Pi:** La Fundación Raspberry Pi es una organización benéfica con sede en el Reino Unido que trabaja para poner el poder de la informática y la creación digital en manos de personas de todo el mundo, esto para que más personas puedan aprovechar el poder de la informática y las tecnologías digitales para el trabajo, resolver problemas importantes y expresarse creativamente (Raspberry, n.d.). En la Figura 16, se muestra la descripción del hardware del Raspberry Pi.

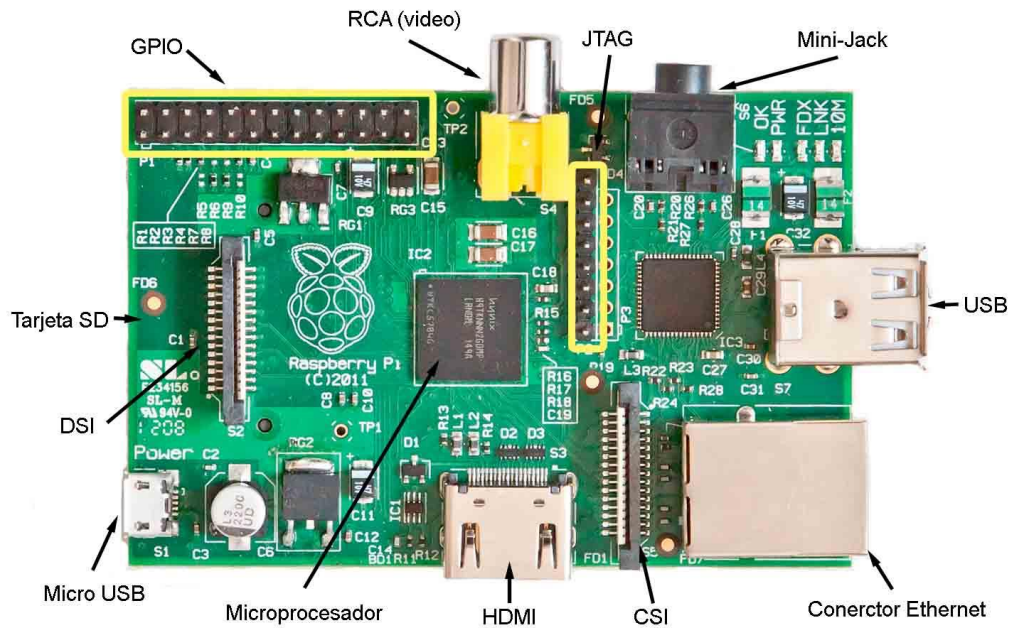


Figura 16. Raspberry Pi.

Fuente: (Diosdado, 2018).

- Arduino:** Arduino es ante todo una empresa de software y hardware de código abierto. La Comunidad Arduino se refiere a la comunidad de usuarios y proyectos que diseña y utiliza placas de desarrollo basadas en microcontroladores. Estas placas de desarrollo se conocen como módulos Arduino, que son plataformas de creación de prototipos de código abierto. La placa de microcontrolador simplificada viene en una variedad de paquetes de placas de desarrollo (Arduino, 2015).



Figura 17. Modelos de Arduino.

Fuente: (Arduino, 2015).

El enfoque de programación más común es utilizar Arduino IDE, que utiliza el lenguaje de programación C. Esto le da acceso a una enorme Biblioteca Arduino que está en constante crecimiento gracias a la comunidad de código abierto. Este es el IDE de Arduino una vez que se ha abierto. Se abre en un boceto en blanco donde puede comenzar a programar de inmediato. Primero, se debe configurar la placa y la configuración del puerto que permitirá cargar el código. Después se conecta la placa Arduino a la PC a través del cable USB (Arduino, 2015).

- *Sparkfun*

Sparkfun es un minorista en línea que produce y vende los widgets y piezas que terminan en muchos proyectos del mercado, prototipos e incluso en la Estación Espacial internacional. Nathan Seidle comenzó la compañía después de volar una placa de circuito en 2003, mientras estudiaba en la Universidad de Colorado. En ese momento, las placas de circuito eran realmente difíciles de conseguir; tenía que enviar su tarjeta de crédito por fax a otro país y esperar que obtuviera su hardware en seis a ocho semanas. Así nació Sparkfun.com, y ahora vende más de 3000 partes diferentes de todas las formas y tamaños para sus necesidades electrónicas digitales. Desde un Arduino básico hasta módulos GPS, se puede encontrarlos con toda la documentación que se necesita para comenzar a usar Sparkfun (Runberg, 2015).

El SparkFun 1-Channel LoRa Gateway es un potente dispositivo con capacidad para 3 redes gracias a un módulo ESPRO WROOM integrado y un módem RFM95W LoRa. El RFM95W maneja la banda de 915MHz mientras que el ESP32 se encarga de las capacidades de Bluetooth y WiFi. Uno de los usos ideales es convertir los mensajes de radio LoRa (de largo alcance) en paquetes de datos a los que puede acceder a través de la web. Es una herramienta perfecta y de bajo costo para monitorear una docena de dispositivos LoRa y transmitir sus mensajes a la nube. Se Complementa con un conector Qwiic y una matriz compatible de placas de conexión de espigas ESP32, el Gateway también puede servir como una plataforma de desarrollo ESP32

/ RFM95W de uso general. LoRa Gateway puede actuar como una puerta de enlace o un dispositivo, pero no ambos al mismo tiempo. Para estar realmente seguro de que su configuración funciona como se espera, debe tener otro dispositivo LoRa para escuchar y otra puerta de enlace LoRa para transmitir. Para usar la radio de 915 MHz en la puerta de enlace, necesitará una antena, para lo cual se tiene dos opciones. Se puede cortar una longitud de cable de núcleo sólido a aproximadamente tres pulgadas para una conexión de antena de orificio pasante con alivio de tensión o se puede usar una antena de 915MHz con un conector U.FL para antenas de mayor rendimiento (Sparkfun, n.d.-c).

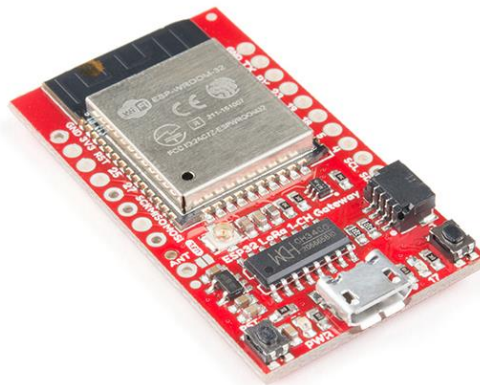


Figura 18. SparkFun LoRa Gateway - 1-Channel (ESP32).

Fuente: (Sparkfun, n.d.-c).

CAPITULO III. DISEÑO DEL SISTEMA

En este capítulo, se utiliza como metodología el modelo en cascada, el cual emplea un proceso secuencial, ayudando a realizar un diseño del sistema de una manera adecuada y organizada. Al iniciar con el desarrollo de este capítulo se lleva a cabo el análisis de la situación actual del invernadero ubicado en el Sector de Piquiucho, seguido del estudio de requisitos o requerimientos, además el diseño del sistema, también la elección de los componentes que integran el proyecto y la ubicación de los nodos y los diagramas de flujo del sistema.

3.1. Descripción general del sistema

El sistema propuesto se basa en una red de sensores inalámbricos para el monitoreo del cultivo de pimiento, el cual consta de un nodo central y 2 nodos que están constantemente en movimiento al interior del invernadero.

En cuanto, al nodo central, se refiere a una puerta de enlace entre la red de sensores inalámbricos y la estación central, tiene como fin almacenar y procesar todos los datos recolectados de los nodos sensores, disminuyendo la carga computacional de procesamiento en los nodos para de esta manera obtener un cultivo eficiente.

Por otra parte, los nodos sensores, recolectan variables entre Co_2 , rayos UV, temperatura y humedad del cultivo de pimiento para posteriormente enviar la información al nodo central por medio de la comunicación con Módulos *LoRaWAN*.

Finalmente, se desea contar con un sistema controlado para el desarrollo del cultivo de pimiento con respecto a los parámetros a medir, que permitan obtener un producto de buena calidad, ayudando hacer más fácil la visualización del crecimiento.

3.2. Metodología de diseño.

La metodología por emplearse para la presente investigación es el “Modelo en cascada”, el cual se lo conoce como un modelo de ciclo de vida lineal-secuencial. En este modelo, cada fase debe completarse antes de que la siguiente fase pueda empezar. Además, se utiliza para el desarrollo de proyectos pequeños, al final de cada fase, se lleva a cabo la revisión para determinar si el proyecto avanza adecuadamente y si se desea continuar o descartar el mismo. Por último, las pruebas comienzan únicamente después de que el desarrollo se haya completado.

Las actividades fundamentales del proceso de desarrollo de software se llevan a cabo como fases separadas y consecutivas. Estas actividades del modelo en cascada constan básicamente de 4 fases que son:

- ***Análisis y definición de requerimientos.***

Se trabaja con los clientes y los usuarios finales del sistema para determinar el dominio de aplicación y los servicios que debe proporcionar el sistema, así como sus restricciones. Con esta información se produce el documento de “Especificación de Requerimientos del Sistema”.

- ***Diseño del sistema y del software.***

Durante el proceso de diseño del sistema se distinguen cuáles son los requerimientos de software y hardware. Después se propone una arquitectura completa del sistema, es decir, durante la etapa de diseño se identifican los subsistemas que componen el sistema y se describe cómo funciona cada uno y las relaciones entre éstos.

- ***Implementación y validación de unidades.***

Consiste en codificar y probar los diferentes subsistemas por separado. La prueba de unidades implica verificar que cada una cumpla su especificación (proveniente del diseño).

- ***Integración y validación del sistema.***

Una vez que se probó que funciona individualmente cada una de las unidades, éstas se integran para formar un sistema completo que debe cumplir con todos los requerimientos del software. Cuando las pruebas del sistema completo son exitosas, éste se entrega al cliente. (Ojeda & Fuentes, 2012).

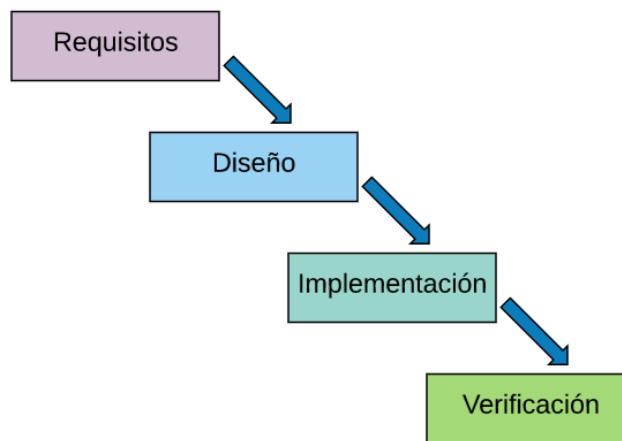


Figura 19. Diagrama del método en cascada.

Fuente: (Domínguez, 2020).

3.3. FASE 1: Requisitos y requerimientos.

Para iniciar con la Fase 1, se da a conocer factores importantes que son: requerimientos operacionales, de usuario, de sistema y de arquitectura que se involucran en el proyecto, por lo que es necesario poseer un análisis investigativo que provea información necesaria sobre los problemas que intervienen en la investigación, posterior a esto, se especifica los requerimientos que cumplan con las necesidades de análisis.

Para determinar los requerimientos que demanda un diseño de red de sensores inalámbricos es necesario realizar un análisis del lugar donde se va a desarrollar este proyecto, a continuación, se dará a conocer la situación actual y las dimensiones del invernadero donde se

va a trabajar, ayudando a conocer la problemática y las condiciones en las que se encuentra actualmente. Para ello, es necesario de visitas de campo al lugar de la implementación.

3.3.1. Características del sistema.

En el invernadero se realizará evaluaciones un tiempo después de haberse realizado la implementación de dicho proyecto para lograr justificar los resultados.

Para obtener una mejor gestión de desarrollo del proyecto, se opta por dividir en nodos como se expone a continuación: los Nodos 1 y 2, se consideran al área de cultivo de pimiento y el Nodo 3 (Servidor), comprende el área de interpretación de los datos censados. En la Figura 20, se muestra una visión general del sistema distribuido por Nodos.

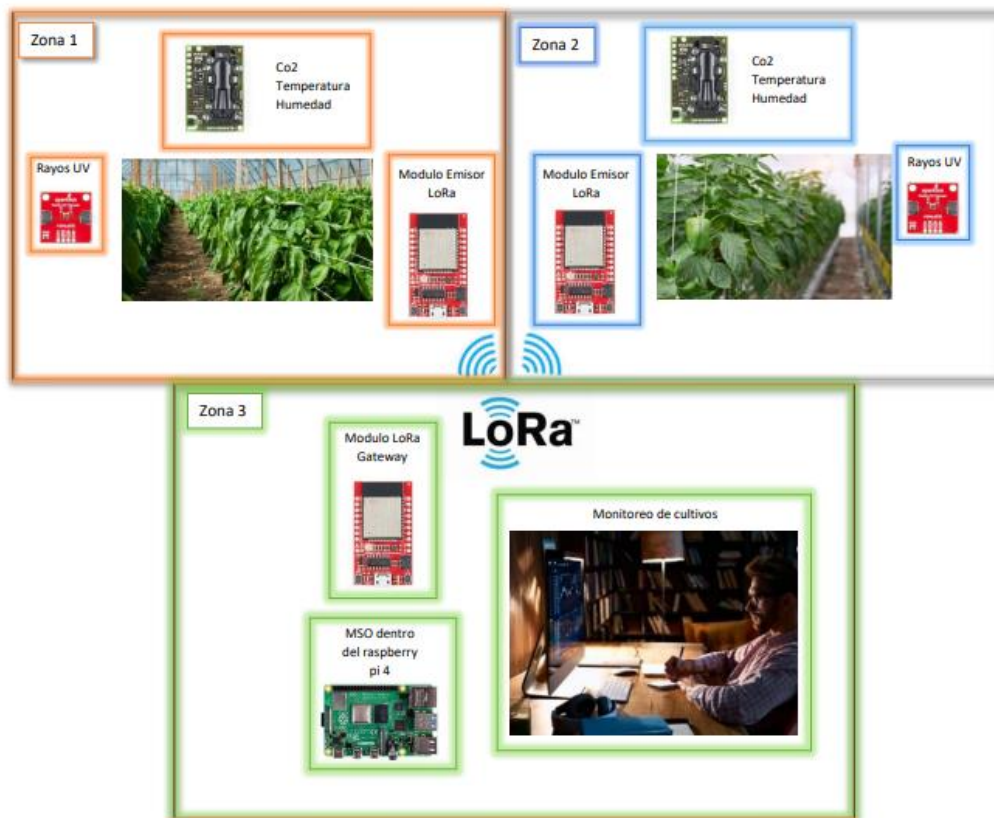


Figura 20. Distribución de Nodos

3.3.2. Definición de acrónimos y abreviaturas.

Para el mayor entendimiento y manejo de los términos en la fase de requerimientos Stakeholders, sistema y arquitectura, se utilizará acrónimos y abreviaturas que se describen en la Tabla 6.

Tabla 6. Términos abreviados y acrónimos usados durante el análisis de requerimientos.

Acrónimos y Abreviados	Descripción
StSR	Requerimientos Stakeholders
SySR	Requerimientos del Sistema
SRSH	Requerimientos de Hardware y Software

3.4. Requerimientos.

Para este proyecto los requerimientos del sistema se establecen en base al estándar ISO/IEC/IEEE 29148:2011, ya que es un estándar completo que brinda normas para la aplicación de los requisitos y procesos relacionados con los que se van a implementar; específicamente se utiliza para la implementación de sistemas y productos de software y servicios a lo largo del ciclo de vida.

Los criterios que se tuvieron en cuenta para determinar la prioridad de los requerimientos del proyecto fueron la importancia para el usuario, el rendimiento del sistema, el riesgo para el diseño y la implementación del presente proyecto.

3.4.1. Situación actual.

El presente proyecto se desarrolla en el invernadero “El Rosal”, ubicada en la comunidad “Piquiucho” en la Parroquia Los Andes perteneciente al Cantón Bolívar, la cual comprende una extensión de aproximadamente 10 hectáreas de terreno, cuenta con cultivos de legumbres, verduras, hortalizas y frutas, posee un tipo de suelo mixto, por lo que se trabaja la tierra (labrarla) antes de sembrar, para que capte de una buena manera las semillas de los cultivos.

La zona de acción de la investigación considera un invernadero, como se muestra en la Figura 21, la cual tiene una extensión de 300m², dedicados prioritariamente al cultivo de pimiento y tomate.



Figura 21. Invernadero

Actualmente el invernadero, no dispone de infraestructura tecnológica en cuanto al monitoreo de cultivos dentro de los invernaderos. Razón por la cual el propietario ha implementado sistemas de producción bajo criterios de buenas prácticas agrícolas con la finalidad de mejorar los métodos convencionales de producción, haciendo énfasis en la inocuidad del producto y que el proceso productivo no genere mayor impacto al ambiente, además, de optimizar recursos y asegurar la calidad del producto, es decir, las personas que trabajan en el invernadero deben recorrer en su totalidad todos los cultivos para inspeccionar si se está desarrollando de manera adecuada. Esto tiene sus desventajas, por lo que muchas veces los invernaderos se encuentran lejos con respecto al lugar donde se encuentren las herramientas para el cuidado de estos.

3.4.2. Requerimientos Stakeholders

Los stakeholders son el conjunto de las partes interesadas del proyecto. En la Tabla 7, se describen los stakeholders que desempeñan un papel diferente de grado de responsabilidad y compromiso, buscando una manera correcta para que el trabajo sea satisfactorio.

Tabla 7. Lista de Stakeholders del proyecto

Stakeholders
1. Invernadero “El Rosal”
2. Sr. Carlos Gordillo-Desarrollador del Proyecto
3. Ing. Paul Rosero-Director inicial del trabajo de titulación
4. Ing. Edgar Maya-Director designado del trabajo de titulación
5. Ing. Mauricio Domínguez- Asesor del presente trabajo de titulación.
6. Ing. Edgar Jaramillo- Asesor del presente trabajo de titulación.

3.2.1. Requerimientos iniciales del sistema.

Los requerimientos iniciales hacen referencia a todas las funcionalidades que debe tener el sistema, comprenden los requisitos de uso, las interfaces, los modos, y estados físicos; los cuales se muestran en la Tabla 8, teniendo en cuenta que de esta manera se puede identificar claramente lo que se necesita para el sistema.

Tabla 8. Requerimientos Iniciales del Sistema.

SySR			
#	Requerimiento	Prioridad	Relación
		Alta Media Baja	
REQUERIMIENTOS DE INTERFAZ			
SySR1	Es necesario ingresar a una interfaz gráfica para visualizar los datos.	X	

SySR2	Comunicación con un Meta Sistema Operativo para disminuir el procesamiento.	X
SySR3	Pines I2C de entrada y pines seriales para salida de información del sistema embebido	X
REQUERIMIENTOS DE USO		
SySR4	El sistema debe estar energizado.	X
REQUERIMIENTOS DE MODOS/ESTADO		
SySR5	El sistema debe estar en funcionamiento todo el tiempo.	X
REQUERIMIENTOS FISICOS		
SySR6	Los nodos deben estar ubicados a una altura adecuada con respecto al suelo, para evitar interferencias en la transmisión.	X

3.2.2. Requerimientos de arquitectura.

En la Tabla 9, se muestra los requerimientos funcionales, donde se incluyen los requisitos de diseño, de software, de hardware y eléctricos; ayudando posteriormente en la elección de los componentes de software y hardware que serán parte del sistema.

Tabla 9. Requerimientos de Arquitectura.

SRSH			
#	Requerimiento	Prioridad	Relación
		Alta Media Baja	
REQUERIMIENTOS DE SOFTWARE			
SRSH1	El software libre para la programación debe ser compatible con la placa de desarrollo a elegir.	X	
SRSH2	Contar con disponibilidad de librerías entre los módulos y la placa de	X	

	procesamiento.	
SRSH3	Sistema operativo y lenguaje de programación de código abierto.	X
REQUERIMIENTOS DE HARDWARE		
SRSH4	Sensores capaces de detectar los fenómenos ambientales dentro del invernadero.	X
SRSH5	Sensores que trabajen en ambientes externos y se acoplen a las características del sistema.	X
SRSH6	Capacidad de procesamiento rápido de los datos recolectados.	X
SRSH7	Sistema embebido que soporte el tratamiento de información en tiempo real.	X
SRSH8	Se requiere de un bajo consumo de energía de los módulos de comunicación inalámbrica	X
SRSH9	El tamaño de los elementos que conforman los nodos deben ser de tamaño reducido.	X
SRSH10	Debe existir disponibilidad de los dispositivos en el mercado local a un precio asequible.	X
REQUERIMIENTOS ELÉCTRICOS		
SRSH11	Fuente de alimentación de 5V.	X
REQUERIMIENTOS DE DISEÑO		
SRSH12	Implementación de un sistema de bajo costo	X

3.3. FASE 2: Diseño del prototipo.

Una vez concluida la etapa de análisis en donde se determinó la situación actual del invernadero “El Rosal” se pudo determinar los requerimientos técnicos y operacionales que

necesita el proyecto para su funcionamiento. Por lo que a continuación, para iniciar con la fase de diseño se da a conocer la topología de la red por medio de un diagrama de bloques, seguido de la elección de los componentes del hardware y software y finalmente el desarrollo y la ubicación de los nodos.

En la Figura 22, se muestra el diagrama de bloques del funcionamiento del sistema electrónico propuesto, además de las partes que llevará cada bloque y cómo van a estar interconectadas. Como se explicó en la Sección 3.2.1.3, se propone dividir al proyecto en nodos tal como se muestra a continuación: los Nodos 1 y 2, comprenden al área de cultivo de pimiento y el Nodo 3, siendo el área de interpretación de los datos censados.

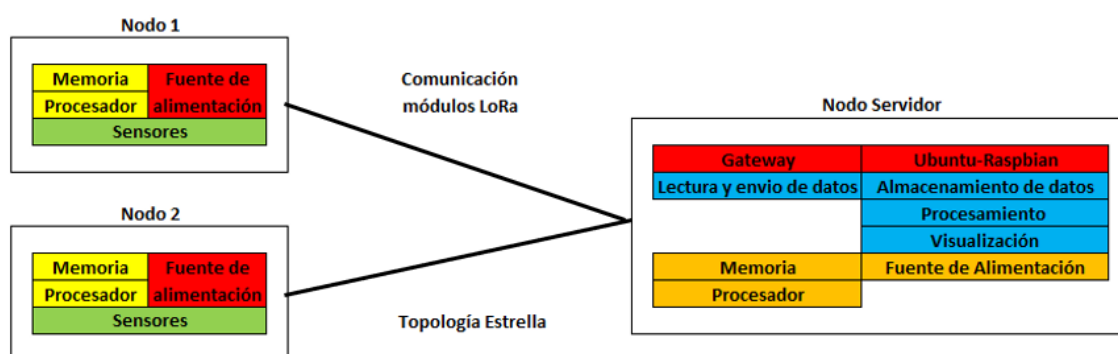


Figura 22. Diagrama de Bloques del Sistema.

3.3.1. Topología de la red.

Los aspectos relevantes que se toman en cuenta para la elección de la topología de la red son: el consumo de energía y el ambiente de trabajo; por lo que al tratarse de un ambiente al aire libre la posibilidad de tener obstáculos es mínima para lograr una comunicación inalámbrica exitosa, además, los nodos deben tener autonomía energética, por lo que se requiere un bajo consumo de energía.

Según (Allience, 2015) la arquitectura de red LoRaWAN se implementa únicamente en una topología de estrella, debido a que los nodos sensores pueden trabajar en modo sleep cuando no tengan que transmitir datos, lo que permite tener un consumo de energía mínimo.

3.4. Elección de hardware y software para el sistema.

Una vez establecido el diagrama de bloques del sistema y la topología de red a desarrollarse se procede a la elección de hardware y software en base a los requerimientos tratados (véase, Sección 3.3.2), mediante tablas comparativas entre los diferentes sistemas embebidos que pueden cubrir todos o la mayoría de los requerimientos asignándose una valoración de 0 y 1 (1: cumple o 0: no cumple), eligiendo el que mayor valoración obtenga.

3.4.1. Elección de Hardware.

En esta sección se determina los componentes generales del proyecto con el fin de optimizar la distribución de los recursos de hardware, por lo cual, se estableció dividir en nodos sobre el área de desarrollo del sistema tal como fue descrita en la Sección 3.2.1.3.

3.4.1.1. Nodos 1 y 2.

Los Nodos 1 y 2 se refieren específicamente al cultivo pimiento, por lo cual como primer componente se elige el microcontrolador que procese los datos recolectados por los sensores, en base a los requerimientos mencionados en la Sección 3.3.2. La Tabla 10, muestra las diferentes opciones que pueden ser idóneas para este proyecto.

Tabla 10. Selección del microcontrolador de los Nodos 1 y 2.

Hardware	Requerimientos				Valoración
	SRSH7	SRSH8	SRSH9	SRSH11	Total
Arduino Uno	1	0	1	1	3
Arduino Mega	1	0	1	1	3
Sparkfun LoRa Gateway	1	1	1	1	4
NodeMcu	1	0	1	1	3

1 Cumple

0 No cumple

Elección: Sparkfun LoRa Gateway

Mediante el análisis realizado en la Tabla 10, se obtiene que el microcontrolador que cumple con la mayor parte de requerimientos que necesita el proyecto es el Sparkfun LoRa Gateway ya que cuenta con un tamaño reducido y un bajo consumo energético, aptos para el desarrollo del sistema debido a que tiene integrado los módulos necesarios para la comunicación inalámbrica como son los módulos LoRa y Esp; en comparación con el Arduino UNO que solo es el microcontrolador sin ningún módulo integrado. Las características básicas del dispositivo seleccionado se muestran en la Sección 3.6.1.

El sensor de rayos UV es un parámetro importante a medir para el desarrollo de la planta de pimiento, por lo cual la Tabla 11 muestra los diferentes modelos que pueden llegar a ser parte del sistema.

Tabla 11. Selección del sensor de rayos UV para los Nodos 1 y 2.

Hardware	Requerimientos				Valoración
	SRSH5	SRSH7	SRSH8	SRSH9	Total
Uv Guva-s12sd	0	1	1	1	3
UV Light Sensor Breakout	1	1	1	1	4
Uv MI8511	0	1	1	1	3

1 Cumple

0 No cumple

Elección: UV Light Sensor Breakout

Mediante el análisis realizado en la Tabla 11, se obtiene que el sensor que cumple con todos los requerimientos que necesita el proyecto para su funcionamiento es el sensor UV Light Sensor Breakout fabricado por Sparkfun ya que cuenta con comunicación I2C , un bajo consumo energético y una alta resistencia a la corrosión, apta para el desarrollo del sistema; en relación con el sensor Uv Guva-s12sd el cual tiene comunicación serial y no se podría enviar tan fácilmente los datos ya que para cada sensor se necesitaría un puerto serial. Las características básicas del dispositivo seleccionado se muestran en la Sección 3.6.2.

Se integra un sensor de CO2, temperatura y humedad ya que es un parámetro importante para medir para el desarrollo de la planta de pimienta, por lo cual la Tabla 12 muestra los diferentes modelos que pueden llegar a ser parte del sistema.

Tabla 12. Selección del sensor CO, temperatura y humedad para los Nodos 1 y 2.

Hardware	Requerimientos				Valoración
	SRSH5	SRSH7	SRSH8	SRSH9	Total
SCD30	1	1	1	1	4
CM1106S	0	1	1	1	3
MQ-2	0	1	1	1	3

1 Cumple

0 No cumple

Elección: SCD30

Mediante el análisis realizado en la Tabla 12, se obtiene que el sensor que cumple con la mayor parte de requerimientos que necesita el proyecto para que funcione adecuadamente es el sensor SCD30 fabricado por Sparkfun, debido a que cuenta con un tamaño reducido, además de tener la capacidad de medir los 3 fenómenos en un mismo módulo sin la necesidad de ocupar

más de 1 sensor, en comparación con los otros sensores que necesitan el acoplamiento más de un módulo para lograr medir estos 3 parámetros. Las características básicas del dispositivo seleccionado se muestran en la Sección 3.6.3.

Por otra parte, se integra un sensor GPS ya que es usado por el sistema de movilidad de la red de sensores para el monitoreo de toda la plantación de pimiento, por lo cual la Tabla 13 muestra los diferentes modelos que pueden llegar a ser parte del sistema.

Tabla 13. Selección del sensor GPS para los Nodos 1 y 2.

Hardware	Requerimientos				Valoración
	SRSH5	SRSH7	SRSH8	SRSH9	Total
SIM 808	0	1	1	1	3
SparkFun GPS Breakout	1	1	1	1	4
MQ-2	0	1	1	1	3

1 Cumple

0 No cumple

Elección: SparkFun GPS Breakout

Mediante el análisis realizado en la Tabla 13, se obtiene que el sensor que cumple con los requerimientos que necesita el proyecto es el sensor SparkFun GPS Breakout fabricado por Sparkfun, debido a que cuenta con un tamaño reducido, además de tener comunicación I2C que se acopla a los demás sensores, con un rango sumamente preciso, en comparación con el SIM 808 que tiene un menor rango de precisión y el envío de datos se los hace a través del puerto serial. Las características básicas del dispositivo seleccionado se muestran en la Sección 3.6.4.

3.4.1.2. *Nodo servidor.*

El Nodo Servidor se refiere al nodo central o gateway el cual está conformado por un módulo Sparkfun LoRa Gateway, el cual fue elegido ya que los Nodos 1 y 2 cuentan con el mismo módulo para tomar datos y realizar la comunicación inalámbrica, el sistema embebido que se incluye en esta parte debe ser el encargado de enviar peticiones a los nodos y receptor la información para finalmente procesa el MSO, por lo cual en la Tabla 14, se muestran diferentes tipos de sistemas embebidos que pueden funcionar en el sistema.

Tabla 14. Selección del sistema embebido en el Nodo Servidor.

Hardware	Requerimientos				Valoración
	SRSH5	SRSH7	SRSH8	SRSH9	Total
Raspberry Pi Zero	1	1	1	1	4
Intel Galileo	0	1	1	1	3

1 Cumple

0 No cumple

Elección: Raspberry Pi Zero

En base a la Tabla 14, la placa que cumple con todos los requerimientos para el funcionamiento del sistema es la Raspberry Pi Zero, debido a que cuenta con un procesador potente (capaz de procesar información en tiempo real), memoria RAM (512 MB) y un consumo de energía bajo (350mA); en relación en la placa Intel Galileo, la cual cuenta con una memoria RAM (256MB), y un consumo de energía alto (380mA). Las características básicas del dispositivo seleccionado se muestran en la Sección 3.6.5.

3.4.2. Elección de software.

El software a escoger debe tener relación en base a los aspectos definidos en la etapa de requerimientos (véase, Sección 3.3), de esta forma se garantiza un desempeño óptimo. De igual manera, como se realizó la elección del hardware del sistema (véase, Sección 3.5.1), la elección del software se lo realiza por Nodos.

3.4.2.1. Nodos 1 y 2.

Para la elección del software, los parámetros que intervienen deben cumplir con los requerimientos de la Tabla 9. Por lo cual, en la Tabla 15, se detallan diferentes tipos de software de programación que se pueden utilizar para el desarrollo de los Nodos 1 y 2.

Tabla 15. Comparativa de software disponibles para los Nodos 1 y 2.

Software	Requerimientos			Valoración
	SRSH11	SRSH12	SRSH13	Total
Python	0	1	1	3
Arduino IDE	1	1	1	4
1 Cumple				
0 No cumple				
Elección: Arduino IDE				

A través de la Tabla 15, se da a conocer que las dos plataformas evaluadas cumplen con todos los requerimientos para el desarrollo del proyecto, sin embargo, debido a que la placa de procesamiento que prevaleció en la parte de Hardware (véase, Sección 3.5.1.1) es un Sparkfun LoRa Gateway se elige el software Arduino IDE para tener compatibilidad, además de ser una multiplataforma de código abierto.

3.4.2.2. *Nodo servidor.*

Este Nodo lo conforma un sistema embebido principal, el cual debe procesar los datos enviados desde los nodos, almacenarlos, procesarlos y mostrarlos en una interfaz local, por lo que se busca un software que tenga capacidad de procesamiento alta, más aún que el los nodos 1 y 2. Los requerimientos del Nodo Servidor se pueden solucionar por medio de un sistema operativo montado en el sistema embebido (Raspberry Pi Zero), en la Tabla 16, se muestra diferentes tipos de software que pueden funcionar.

Tabla 16. Comparativa de software disponibles del Nodo Servidor

Software	Requerimientos			Valoración
	SRSH11	SRSH12	SRSH13	Total
Ubuntu-Raspbian	1	1	1	3
Windows	0	1	1	4

1 Cumple

0 No cumple

Elección: Ubuntu-Raspbian

En base a la Tabla 16, se concluye que el software que cumple con todos los requerimientos para el funcionamiento del sistema es Ubuntu Raspbian, por su compatibilidad con la placa de procesamiento (Raspberry Pi Zero) escogida en la Sección 3.5.2.2, además de contar con un sistema operativo de código abierto netamente para fines de programación, en relación con Windows, que es un sistema operativo con licencia de pago.

3.5. Hardware destinado para el sistema.

Este sistema se conforma por una red de sensores inalámbricos, ubicados en los nodos 1 y 2 pertenecientes al cultivo de pimiento y Nodo Servidor refiriéndose al nodo central. Para la

elección de los componentes se lo realiza referenciando los requerimientos expuestos en la Tabla 9.

3.5.1. Microcontrolador SparkFun LoRa Gateway.



Figura 23. SparkFun LoRa Gateway - 1-Channel (ESP32).

Fuente: (Sparkfun, n.d.-c).

La Figura 23, muestra la placa SparkFun LoRa Gateway, es fabricado por una de las empresas más reconocidas en el área, es compatible y se programa con Arduino IDE. A continuación, en la Tabla 17, se detallan las características relevantes de la placa.

Tabla 17. Características principales del SparkFun LoRa Gateway.

Especificaciones	Propiedades
Microcontrolador:	Moteino basado en ATmega328p
Voltaje Operativo:	5V
Tensión de Entrada:	7-12V
Corriente modo Wifi:	150 mA
Corriente modo LoRa:	70 mA
Banda de Frecuencias:	868-915 MHz
Pines digitales de Entrada/Salida:	7

Pines análogos de entrada:	3
Pines I2C:	2
Alcance:	2 Km de distancia

Fuente: Datasheet de la placa SparkFun LoRa Gateway.

3.5.2. Sensor de luz UV SparkFun (VEML6075).



Figura 24. SparkFun UV Light Sensor Breakout.

Fuente: (Sparkfun, n.d.-d).

El sensor de luz ultravioleta VEML6075 como se muestra en la Figura 24 y se detalla en el Anexo 2, es la última solución de detección ultravioleta de SparkFun. El VEML6075 implementa un fotodiodo simple para medir los niveles de radiación UVA (320-400nm) y UVB (280-320nm). Con esta ruptura, podrá leer la intensidad de estos tipos de luz en irradiancia y, a partir de ahí, calcular el índice UV. A continuación, en la Tabla 18, se muestra las características del sensor UV.

Tabla 18. Características principales del sensor SparkFun UV Light Sensor Breakout.

Especificaciones	Propiedades
Voltaje de Operación:	1.7V-3.6V
Corriente:	480 μ A
Resolución UVA:	2.1 muestras/ μ W/cm ²

Fuente: Datasheet del sensor UV Light Sensor Breakout (VEML6075).

3.5.3. Sensor de temperatura y humedad de CO₂ (SCD30).

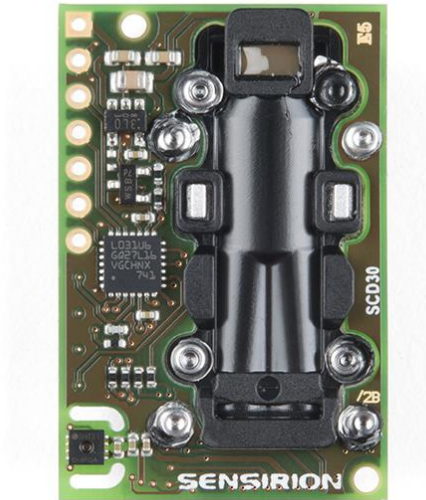


Figura 25. Sensor de temperatura y humedad de CO₂

Fuente: (Sparkfun, n.d.-a).

El SCD30 de Sensirion como se muestra en la Figura 25 y se detalla en el Anexo 3, es un sensor de CO₂ basado en infrarrojos no dispersivos (NDIR) de alta calidad capaz de detectar de 400 a 10000 ppm con una precisión de $\pm (30 \text{ ppm} + 3\%)$. Para mejorar la precisión, el SCD30 tiene sensores de temperatura y humedad incorporados, así como comandos para establecer la altitud actual. Para mayor precisión, el SCD30 también acepta lecturas de presión ambiental. A continuación, en la Tabla 19, se muestra las características del sensor de temperatura y humedad de CO₂.

Tabla 19. Características principales del sensor de temperatura y humedad de CO₂.

Especificaciones	Propiedades
Voltaje de operación:	3.3V - 5.5V
Dimensiones:	35 mm x 23 mm x 7 mm
Rango de medición:	400 ppm – 10.000 ppm
Precisión:	$\pm(30 \text{ ppm} + 3\%)$

Corriente:	19 mA
Consumo de energía:	120 mJ

Fuente: Datasheet del sensor de temperatura y humedad de CO₂ (SCD30).

3.5.4. Módulo GPS (SAM-M8Q).

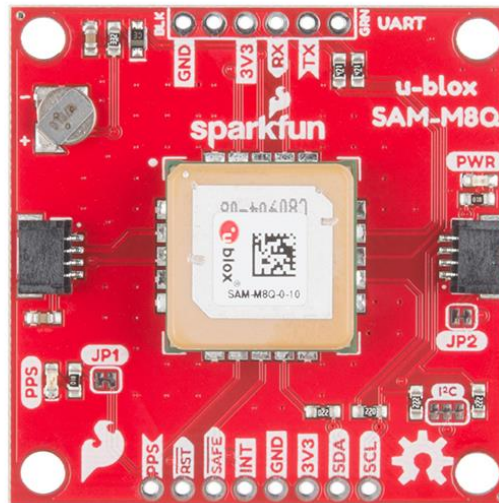


Figura 26. Módulo GPS (SAM-M8Q).

Fuente: (Sparkfun, n.d.-b).

El SparkFun SAM-M8Q GPS Breakout es una placa GPS de alta calidad con opciones de configuración sumamente impresionantes, como se muestra en la Figura 26 y se detalla en el Anexo 4. El SAM-M8Q es un receptor GNSS de 72 canales, lo que significa que puede recibir señales de las constelaciones GPS, GLONASS y Galileo. Esto aumenta la precisión y disminuye el tiempo de bloqueo y, gracias a la batería recargable incorporada, tendrá energía de respaldo que permite que el GPS se bloquee en segundos. A continuación, en la Tabla 20, se muestra las características del módulo GPS (SAM-M8Q).

Tabla 20. Características principales del módulo GPS (SAM-M8Q).

Especificaciones	Propiedades
Voltaje de Operación:	3.3V

Corriente:	29mA
Precisión horizontal:	2.5m
Tasa de actualización máxima:	18 Hz
Altitud máxima:	50000m
Velocidad máxima:	500 m/s

Fuente: Datasheet del módulo GPS (SAM-M8Q).

3.5.5. Sistema embebido Raspberry Pi Zero.

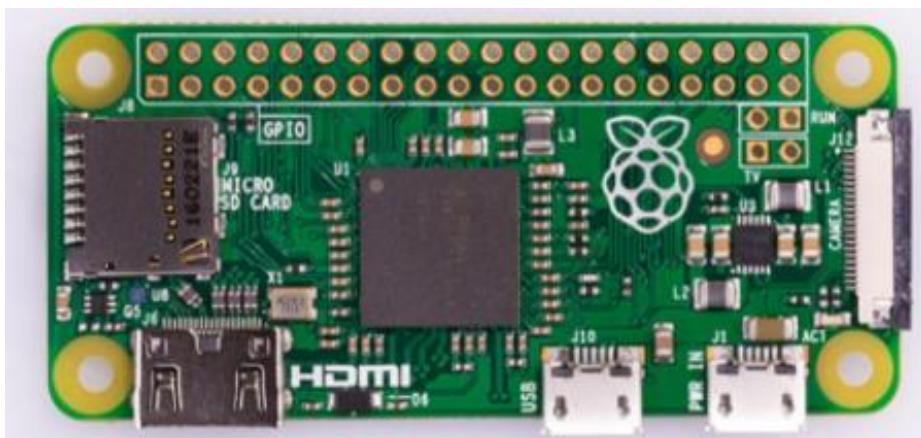


Figura 27. Placa Raspberry Pi Zero.

Fuente: (Tzivaras, 2017).

La placa Raspberry PI Zero, es una placa con gran nivel de procesamiento y a la vez compacta como se muestra en la Figura 27, se escoge esta placa, debido a que tiene un procesador rápido, su precio es reducido, haciéndolo asequible al usuario y además se encuentra disponible en el mercado nacional. A continuación, en la Tabla 21, se detallan las características destacadas de esta placa:

Tabla 21. Características de la Placa Raspberry Pi Zero.

Especificaciones	Propiedades
Procesador:	Broadcom BCM283, ARM 11

RAM:	512 MB
Voltaje:	5V
Salida Video:	mini-HDMI a 1080p.
Micro SD:	Si
Puerto USB:	2 micro-USB
Pines:	40

Fuente: Datasheet placa Raspberry Pi Zero.

3.6. Nodos 1 y 2.

Los nodos 1 y 2 contienen los sensores encargados de adquirir los datos provenientes del cultivo y transmitirlos hacia el nodo servidor para que se pueda almacenar y procesar la información mediante el MSO. Los nodos tienen sensores de rayos UV, CO2, temperatura, humedad y un módulo GPS, una placa de procesamiento Sparkfun LoRa Gateway, un módulo de comunicación LoRa RF96 integrada en el procesador, y una fuente de energía, como muestra el diagrama circuital de la Figura 28.

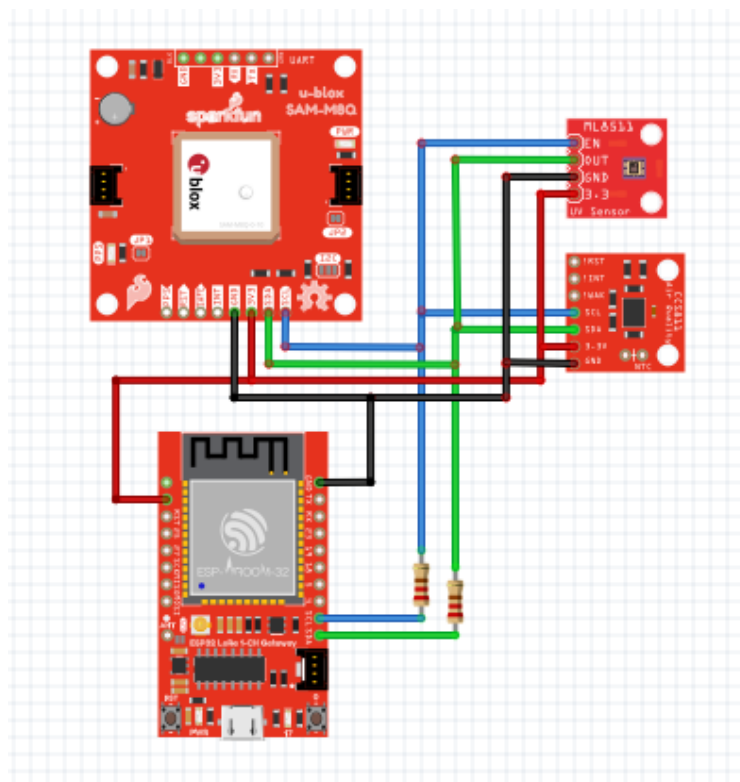


Figura 28. Diagrama Circuital de los Nodos 1 y 2

Los sensores miden diferentes parámetros tanto para el cultivo de pimiento, la placa SparkFun LoRa Gateway procesa los datos enviados por los sensores y el módulo LoRa RF96 integrado en la placa realiza la comunicación inalámbrica con el nodo servidor para el envío de información.

Para que los sensores funcionen correctamente se opta por realizar una placa o Shield que acople todos los sensores mediante el software *Eagle* como se muestra en la Figura 29, el cual es un programa de diseño de placas de circuito impreso (PCB). Los componentes que contienen los Shield son los sensores y los conectores hembra.

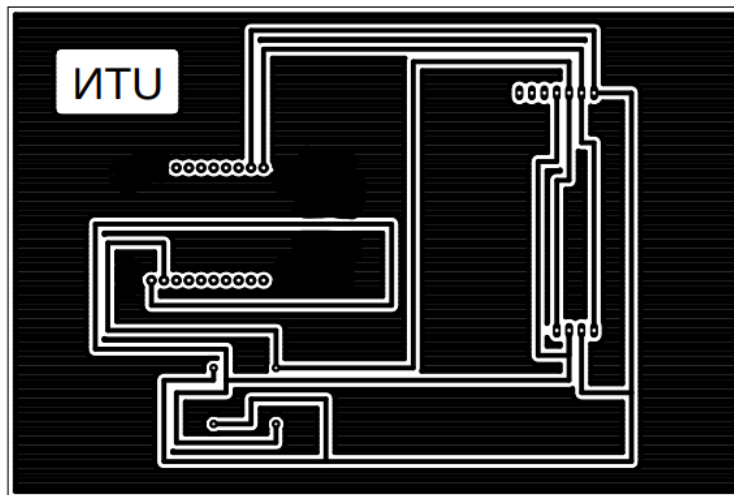


Figura 29. Diseño de la PCB - Shield para los nodos 1 y 2.

Una vez realizado el Shield, permite acoplar todos los sensores con gran facilidad entre ellos, lo que facilitaría el trabajo si se desea cambiar cualquier componente que se tenga que reemplazar o por mantenimiento. En la Figura 30, se muestra el montaje de las placas de los nodos 1 y 2 en el Shield.

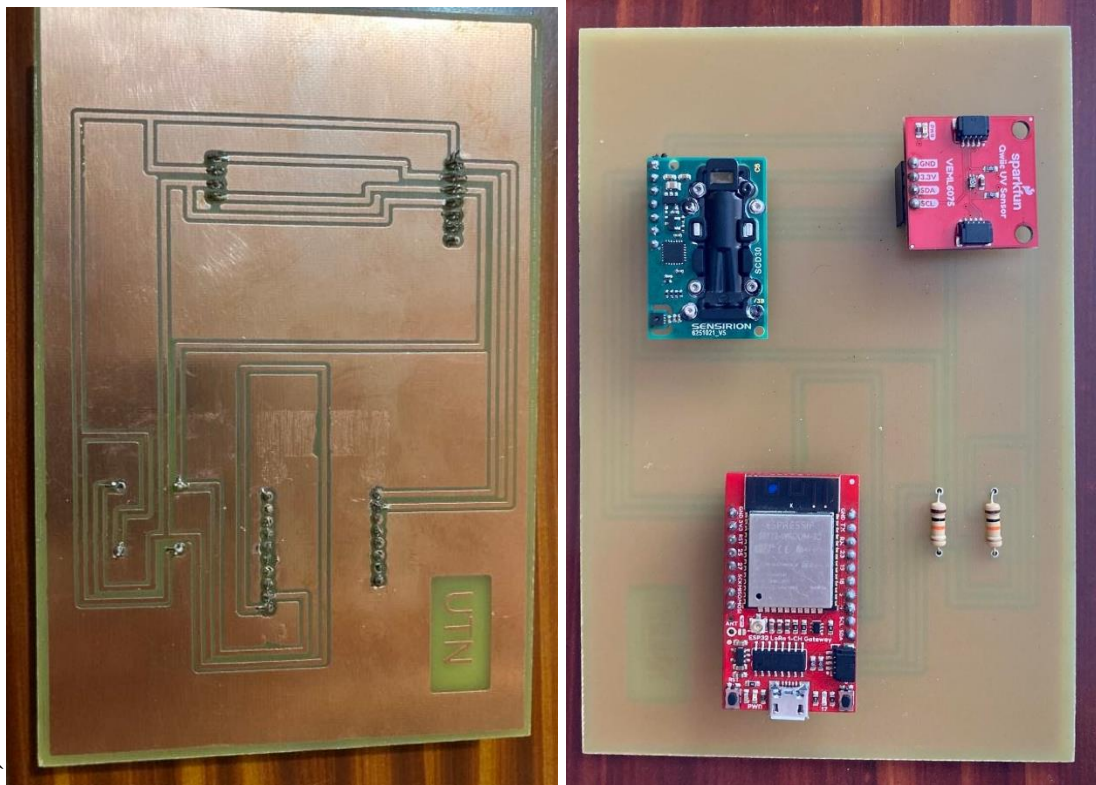


Figura 30. Vista general de los Nodos 1 y 2.

3.6.1. Diagrama de flujo nodos 1 y 2.

Este apartado pretende ilustrar de manera general el funcionamiento de los nodos sensores, acorde al diagrama de flujo de la Figura 31, en un principio el nodo realiza el proceso de inicialización de variables, comunicación I2C y las respectivas librerías de los sensores, consecutivamente, la adquisición de los datos provenientes de los sensores, generando la trama de comunicación LoRaWAN para finalmente, enviar los datos al Gateway y esperar la confirmación que los datos llegaron correctamente.

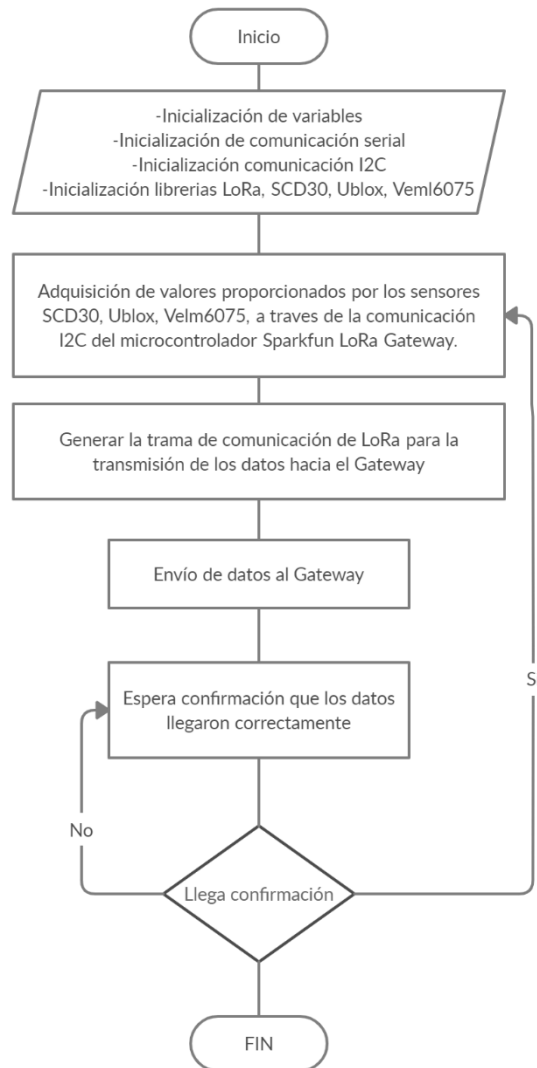


Figura 31. Diagrama de Flujo del Nodos 1 y 2.

3.6.2. Ciclo de trabajo y tiempo en el aire (ToA).

Al enviar la señal desde un nodo recolector especificado en la red de sensores, lleva cierto tiempo antes que la señal llegue a su destino, por lo que se debe realizar el cálculo del tiempo en el aire de la señal tomando en cuenta el ciclo de trabajo del módulo LoRa, el ciclo de trabajo es una proporción de tiempo durante el cual opera un módulo medido generalmente en porcentaje. En el caso de dispositivos que trabajan en la banda de 915Mhz, no existe un tiempo de trabajo especificado ya que las redes establecidas están en la banda de 868Mhz en Europa

tomando en cuenta el 1% de ciclo de trabajo como valor referencial en el cálculo del tiempo en el aire de la señal (Castells et al., 2018).

Gracias al tiempo de trabajo establecido por la regulación ISM en base a la banda de 868Mhz se hace una relación al mismo tiempo de trabajo de la banda en 915 Mhz, del mismo modo para las bandas en China y Australia (Castells et al., 2018).

La fórmula para calcular el Tiempo en el Aire de la señal es:

$$ToA = \text{Tiempo establecido de envío} * (100 - (\text{ciclo de trabajo})) \quad (\text{Ecuación 2})$$

Donde:

- Tiempo establecido de envío= 200ms (valor establecido por el Administrador).
- Ciclo de trabajo= 1%.

$$ToA = 200ms * (100 - (1))$$

$$ToA = 200ms * (99)$$

$$ToA = 19.8seg.$$

Una vez calculado el tiempo en el aire de la señal (ToA) se obtiene en segundos el tiempo que se debe esperar antes de enviar un nuevo mensaje entre la comunicación de la puerta de enlace y cada uno de los nodos recolectores.

3.6.3. Tipo de clase seleccionado para la red WSN.

Como se especificó en el Capítulo II acerca de las clases de los dispositivos LoRaWAN, se debe establecer una clase para el trabajo óptimo entre el nodo recolector y la puerta de enlace, para ello se hace una comparación con las 3 clases existentes y se procede a seleccionar la de mejor rendimiento para el sistema de inundaciones.

La clase A especifica que un nodo final puede transmitir una señal, después de dicho proceso el mismo nodo escucha una respuesta de la puerta de enlace para poder transmitir, el nodo final abre dos ranuras de recepción en T_1 y T_2 segundos, entonces la puerta de enlace puede responder dentro del tiempo T_1 o en el tiempo de T_2 , pero no en ambas al mismo tiempo. En la Figura 32 se puede apreciar el proceso de la clase A.

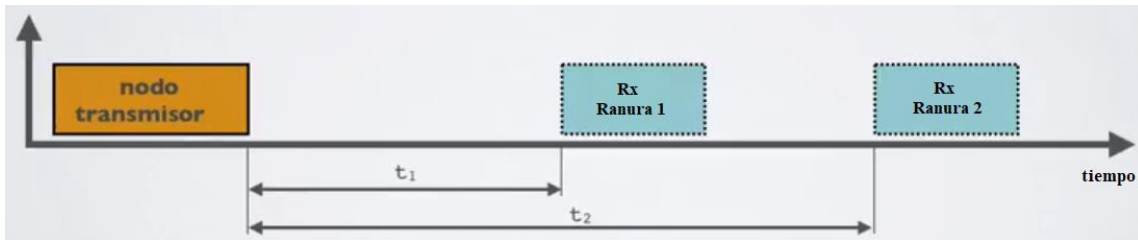


Figura 32. Representación de Clase A en LoRaWAN.

Fuente: (Noguera, 2020)

La clase B aplica los mismos tiempos y proceso de la clase A, a diferencia que en la clase B el nodo final recibe una lluvia sincronizada de señales en el tiempo enviados desde la puerta de enlace, permitiendo a la puerta de enlace saber que el nodo está escuchando y transmitiendo. En la Figura 33 se puede apreciar el proceso de la clase B tomando en cuenta la compatibilidad con la clase A y el tiempo de periodo de ping en el que los dispositivos verifican la comunicación permanente entre ellos.

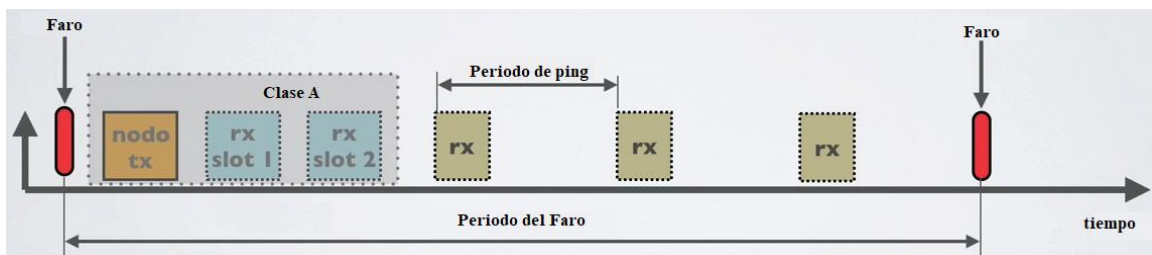


Figura 33. Representación de Clase B en LoRaWAN.

Fuente: (Noguera, 2020)

La clase C del mismo modo aplica las ranuras de recepción de la clase A, pero consolidando que un dispositivo de clase C puede escuchar respuestas continuamente por un tiempo

establecido por el desarrollador. En la Figura 34 se aprecia la compatibilidad de la clase A con la clase C y el tiempo que permanece activo el canal esperando un nuevo enlace.

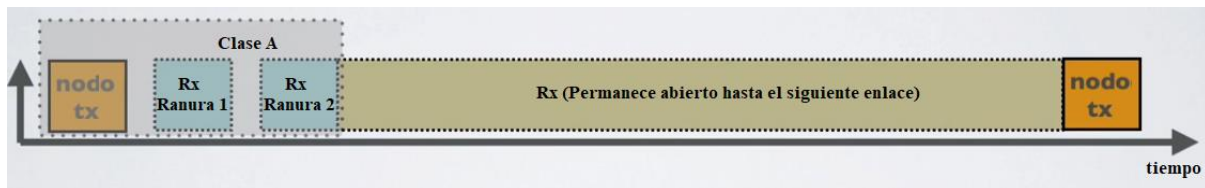


Figura 34. Representación de Clase C en LoRaWAN.

Fuente: (Noguera, 2020)

Conclusión de la Clase para utilizar en la comunicación LoRa. Gracias al proceso realizado donde se especifica las 3 clases de transmisión de los módulos LoRa, se procede a tomar la Clase C como referencia para la realización del sistema, ya que los nodos recolectores deben estar siempre activos y transmitiendo datos, tomando en cuenta que de igual forma es compatible con la Clase A, haciéndolo más versátil.

3.6.4. Pérdidas en el espacio libre.

La pérdida de energía entre dos antenas al atravesar una atmosfera se llama perdida de espacio libre, el modelo de perdida de trayectoria en el espacio libre es usado para predecir la intensidad del nivel de recepción cuando el transmisor y el receptor tiene una trayectoria de línea de vista como se establece entre el nodo recolector y la puerta de enlace.

La ecuación para el cálculo de las perdidas en el espacio libre es:

$$L(fs) = 32.45 + 20 \log(D) + 20 \log(f) \quad (\text{Ecuación 3})$$

Donde:

- $L(fs)$ = pérdida de espacio libre en dB.
- D = Distancia entre el nodo final y la puerta de enlace en KM.
- f = frecuencia en Mhz.

Con la formula establecida se hace el cálculo de la perdida de espacio libre con una la distancia máxima de transmisión ideal del módulo LoRa, con un valor de 1.2km en zona rural y 800m en zona urbana y una frecuencia de 915Mhz para el continente americano.

- Zona Urbana.

$$L(fs) = 32.45 + 20 \log(1.2) + 20 \log(915)$$

$$L(fs) = 93.26dB.$$

- Zona Rural.

$$L(fs) = 32.45 + 20 \log(0.8) + 20 \log(915)$$

$$L(fs) = 89.74dB.$$

3.6.5. Ubicación de los nodos.

El área de estudio a tratar está compuesta por 300 metros a la redonda correspondientes a un invernadero de pimientos, el presente proyecto está desarrollado para cubrir en su totalidad el invernadero. La topología es tipo estrella, es decir que cada nodo en teoría debería estar a un máximo de 1,2 km de distancia del nodo central o puerta de enlace, en la Figura 35 se puede apreciar las distancias de cada nodo.



Figura 35. Línea de vista entre el nodo servidor y nodos sensores.

Fuente: *Obtenido de Google Earth.*

Para el despliegue de la red se han planteado 2 nodos sensores y un nodo central, los cuales se encuentran distribuidos a una distancia que no excede la capacidad del nodo central. En el caso del nodo central se lo ha ubicado dentro de la parte residencial de la parroquia, así se facilita para que los datos censados puedan ser subidos a la plataforma en tiempo real. Para dejar estableciendo la ubicación exacta de los nodos se utilizó la herramienta de Google Earth para establecer la latitud y longitud de cada uno, como se especifica en la Tabla 22.

Tabla 22. Coordenadas geográficas de cada nodo.

Dispositivo	Distancia hacia la puerta de enlace	Coordenadas geográficas		Coordenadas Angulares	
		Latitud	Longitud	Latitud	Longitud
Puerta de enlace	-	0°26'41.44"N	77°54'58.01"O	0,2685	-77,542949
Nodo 1	118.11	0°26'45.18"N	77°55'1.85"O	0,2677305	-77,559179
Nodo 2	234.03	0°26'42.42"N	77°55'0.79"O	0,2699805	-77,550791

Fuente: Google Earth.

Para comprobar que existe línea de vista se procede a simular el escenario con el fin de confirmar la zona de Fresnel de cada nodo, mediante el uso del Radio Mobile.

Se crean los puntos que conforman el enlace, haciendo uso de las coordenadas obtenidas en Google Earth presentados en la Tabla 22. En la Figura 36 se puede apreciar la ventana de Radio Mobile donde se puede expresar los valores de las coordenadas de cada nodo a ingresar.

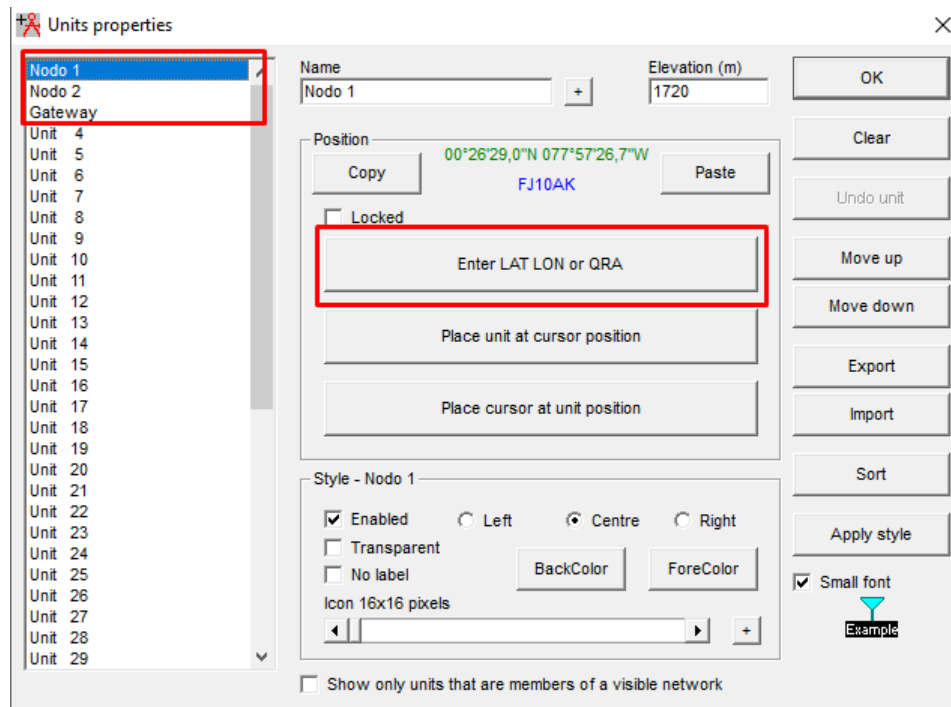


Figura 36. Puntos de referencia de cada nodo en Radio Mobile.

Al establecer la ubicación de cada nodo en la topología de red, se debe configurar los parámetros principales con los que trabajan los equipos, a continuación, se presentan los valores a tomar en cuenta. En la Figura 37 se puede apreciar los parámetros principales de la red.

- Frecuencia máxima y mínima: 915Mhz.
- Polarización: Horizontal.

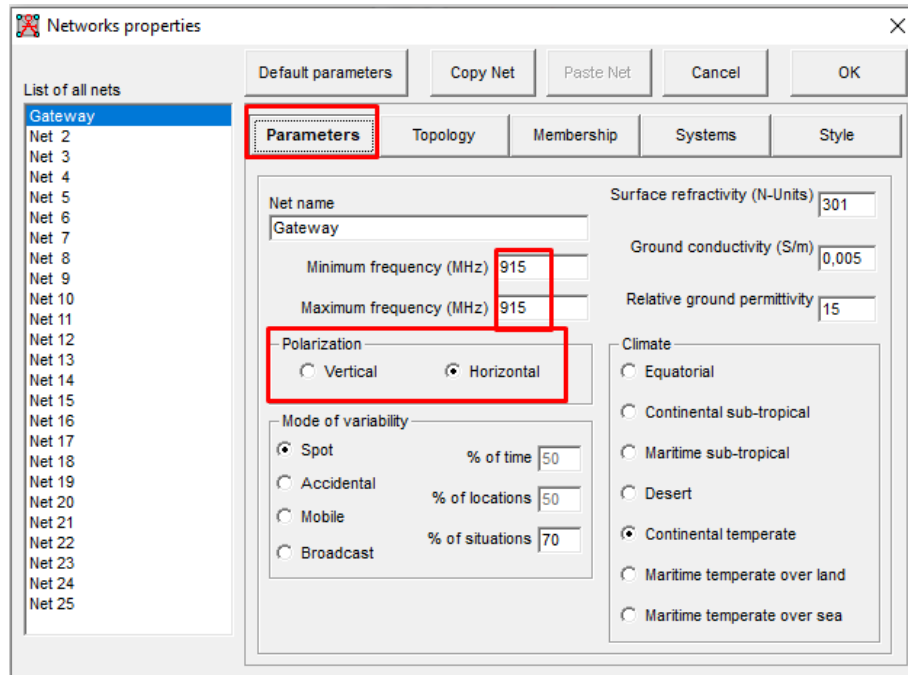


Figura 37. Parámetros principales de la red.

Del mismo modo se debe establecer los parámetros del sistema de cada comunicación, al ser el mismo tipo de enlace entre cada nodo y el nodo central, se entiende que todos los enlaces deben tener las mismas características. En la Figura 38 se puede apreciar los valores establecidos a continuación:

- Potencia de Transmisión (dBm): 20dBm.
- Umbral de Recepción (dBm): -148dBm.
- Pérdida de línea (dB): 0.5dB.
- Tipo de antena: Omnidireccional.
- Ganancia de la Antena (dBi): 3dBi.
- Altura de la Antena (m): 10 metros.

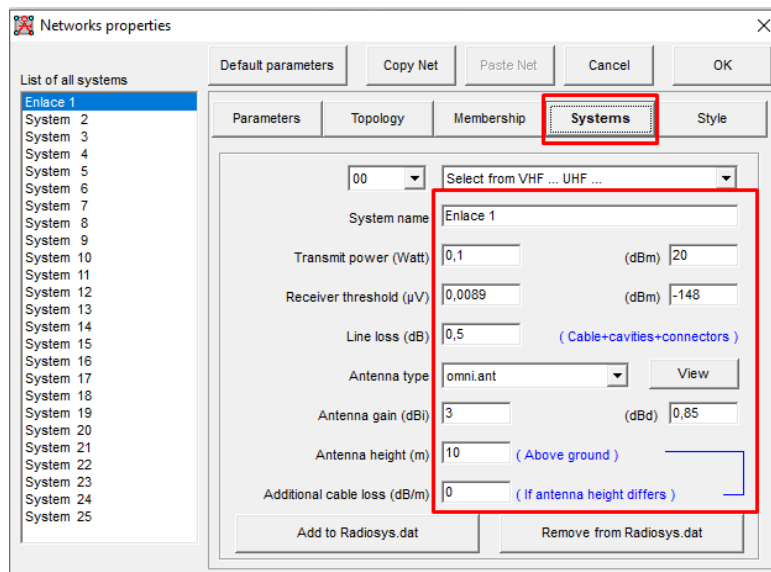


Figura 38. Parámetros principales del sistema de enlaces.

Finalmente, el resultado obtenido de la simulación en Radio Mobile es la topología en estrella en el cultivo de pimiento como muestra la Figura 39.

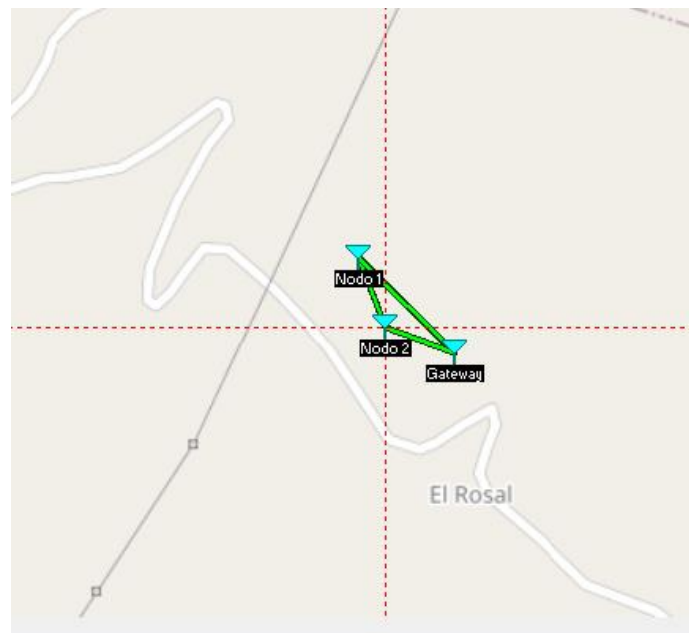


Figura 39. Resultado de la simulación del cultivo de pimiento.

3.6.6. Zona de Fresnel.

La zona Fresnel es un cuerpo de forma elíptica alrededor de la línea directa de la ruta de visión entre el nodo recolector y la puerta de enlace. La fórmula establecida para el cálculo de la zona de Fresnel con módulos LoRa es la siguiente:

$$r = 8.657 \times \sqrt{D/f} \quad (\text{Ecuación 6})$$

(Perez, 2018) indica que:

- r = radio de la zona de fresnel en metros
- D = distancia en Km.
- f = frecuencia en Ghz.

3.6.6.1. Zona de Fresnel entre nodo 1 y puerta de enlace.

En la Figura 40 se puede observar la línea de vista entre el nodo recolector 1 que se encuentra a 20 centímetros del suelo y la puerta de enlace con una altura de 10 metros.

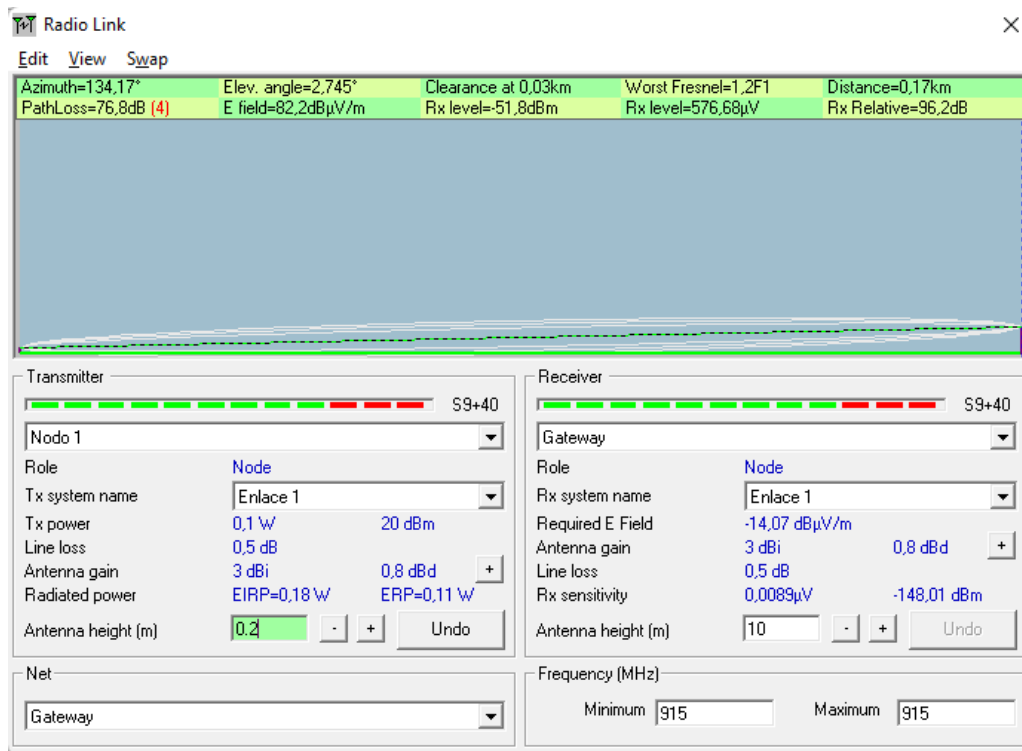


Figura 40. Zona de Fresnel e Intensidad de señal-nodo 1.

Se realiza el cálculo de la zona de Fresnel en base a la distancia entre el nodo 1 y la puerta de enlace.

$$r = 8.657 \times \sqrt{0.118/0.915}$$

$$r = 3.108\text{m}$$

3.6.6.2. Zona de Fresnel entre nodo 2 y puerta de enlace.

En la Figura 41 se puede observar la línea de vista entre el nodo recolector 2 ubicado a 20 centímetros del suelo y la puerta de enlace con una altura de 10 metros.

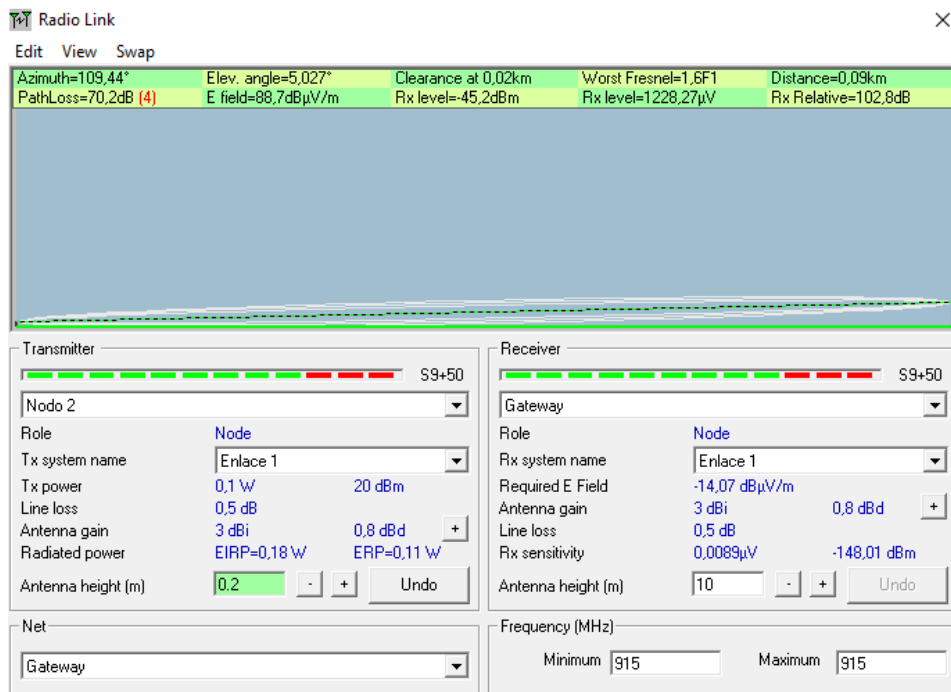


Figura 41. Zona de Fresnel e Intensidad de señal-nodo 2.

Se realiza el cálculo de la zona de Fresnel en base a la distancia entre el nodo 2 y la puerta de enlace.

$$r = 8.657 \times \sqrt{0.513/0.915}$$

$$r = 6.48\text{m}$$

3.6.7. Potencia radiada isotrópica efectiva (PIRE)

La potencia radiada isotrópica efectiva es la potencia total radiada por una antena isotrópica en una dirección establecida, cabe recalcar que las antenas isotrópicas emiten su intensidad de radiación en todas las direcciones del espacio. El cálculo del PIRE se realiza para establecer un valor nominal que no se debe exceder basándose en los organismos reguladores como el ISM.

La fórmula para calcular la potencia radiada isotrópica efectiva es:

$$P_{ire} = \text{Potencia de transmisión}(dBm) + \text{Ganancia de antena } (dBi) \quad (\text{Ecuación 4})$$

$$- \text{perdida de cable}(dBm)$$

$$P_{ire} = 20(dBm) + 10(dBi) - 5(dBm)$$

$$P_{ire} = 25dBm$$

3.6.8. Indicación de intensidad de señal recibida (RSSI).

La indicación de intensidad de señal recibida o (RSSI) es un valor negativo que se mide en dBm e indica la intensidad en base a la distancia entre un enlace de dos nodos, el RSSI indica que mientras el valor sea más cercano a 0, la señal es más potente y mejor establecida y si el valor negativo es alto significa que la señal es más débil.

- Si RSSI = -30dBm: la señal es fuerte.
- Si RSSI=-120dBm: la señal es débil.

3.6.9. Factor de Propagación y sensibilidad con relación al receptor.

Como se puede ver la tecnología *LoRaWAN* utiliza una modulación basada en técnicas de espectro ensanchando y una variación de espectro de propagación.

LoRaWAN dispone de diferentes valores de velocidades en función de la señal, con velocidades comprendidas entre 0.3 Kbps y 22 Kbps basado en el factor de expansión.

La red LoRaWAN será la encargada de administrar el factor de ensanchamiento utilizado y la potencia de señal de transmisión para optimizar el rendimiento y la escalabilidad. En la Tabla 23 se puede apreciar la relación mediante el factor de propagación.

Tabla 23. Relación de Factor de Propagación.

Factor de Propagación	Tasa de bits equivalente(kb/s)	Sensibilidad(dBm)
12	0.293	137
11	0.537	134.5
10	0.976	132
9	1.757	129
8	3.125	126
7	5.468	123

Se debe tener en cuenta que a mayor distancia entre el punto emisor y el nodo receptor, el factor de propagación será mayor, disminuyendo la velocidad de transmisión, así como una sensibilidad por parte del receptor mucho más alta.

3.7. Nodo servidor.

El nodo servidor es el encargado de comunicarse con cada uno de los nodos, además, es el responsable de realizar el procesamiento mediante el Meta Sistema Operativo, almacenamiento y visualización de datos. En la Figura 42, se muestra el diagrama circuital, integrado por una placa de procesamiento Raspberry Pi Zero y un módulo Sparkfun LoRa Gateway y una fuente de energía.

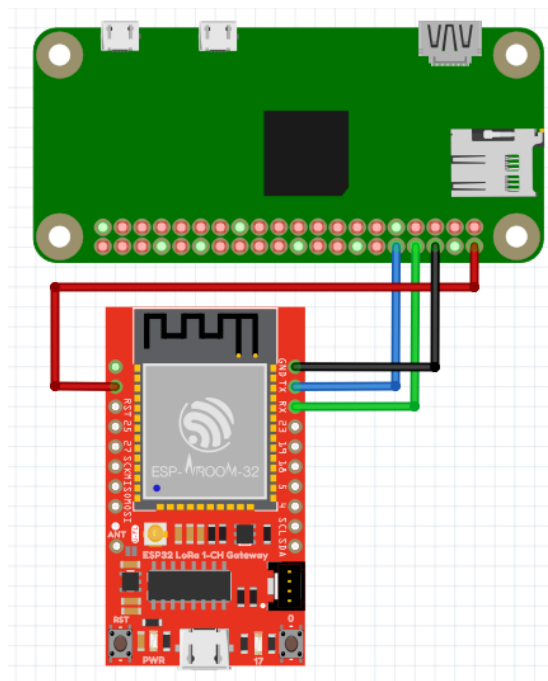


Figura 42. Diagrama Circuital del Nodo Servidor.

El módulo LoRa RF96, es el encargado de comunicarse inalámbricamente con los nodos 1 y 2 mediante tecnología LPWAN (Low Power Wide Area Network), la placa de procesamiento Raspberry Pi Zero se conecta al módulo Sparkfun LoRa Gateway por medio del puerto serial, además se encarga de procesar los datos monitoreados y almacenarlos en una base de datos

para su posterior visualización en una interfaz local, la fuente de alimentación va a estar conectada permanentemente a un banco de baterías para evitar la pérdida de información.

3.7.1. Diagrama de flujo nodo servidor.

Este apartado pretende ilustrar de manera general el funcionamiento del nodo servidor, acorde al diagrama de flujo de la Figura 42. Al iniciar, el nodo inicializa las variables y la comunicación con los nodos 1 y 2, posteriormente envía peticiones a los nodos sensores para que puedan tener una comunicación y receptor los datos del monitoreo para almacenarlos en la base de datos.

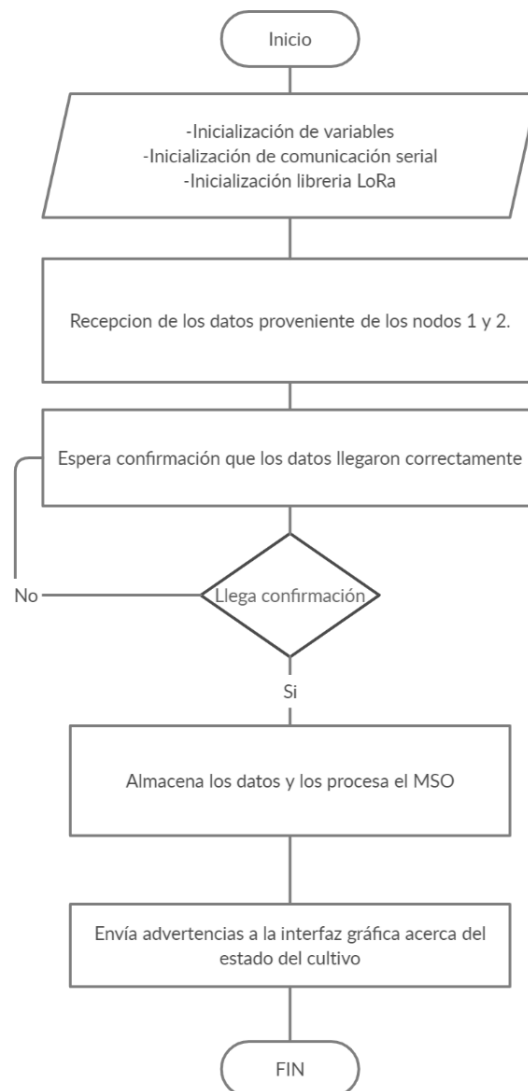


Figura 43. Diagrama de Flujo del Nodo Servidor.

3.8. Diagrama de bloques de funcionamiento

En la Figura 44 se detalla el diagrama de bloques general de funcionamiento del sistema, en donde cada una de las etapas viene a ser una parte fundamental para el funcionamiento y desempeño total del proyecto, a continuación, se detallan brevemente las etapas en base a sus funciones y objetivos.

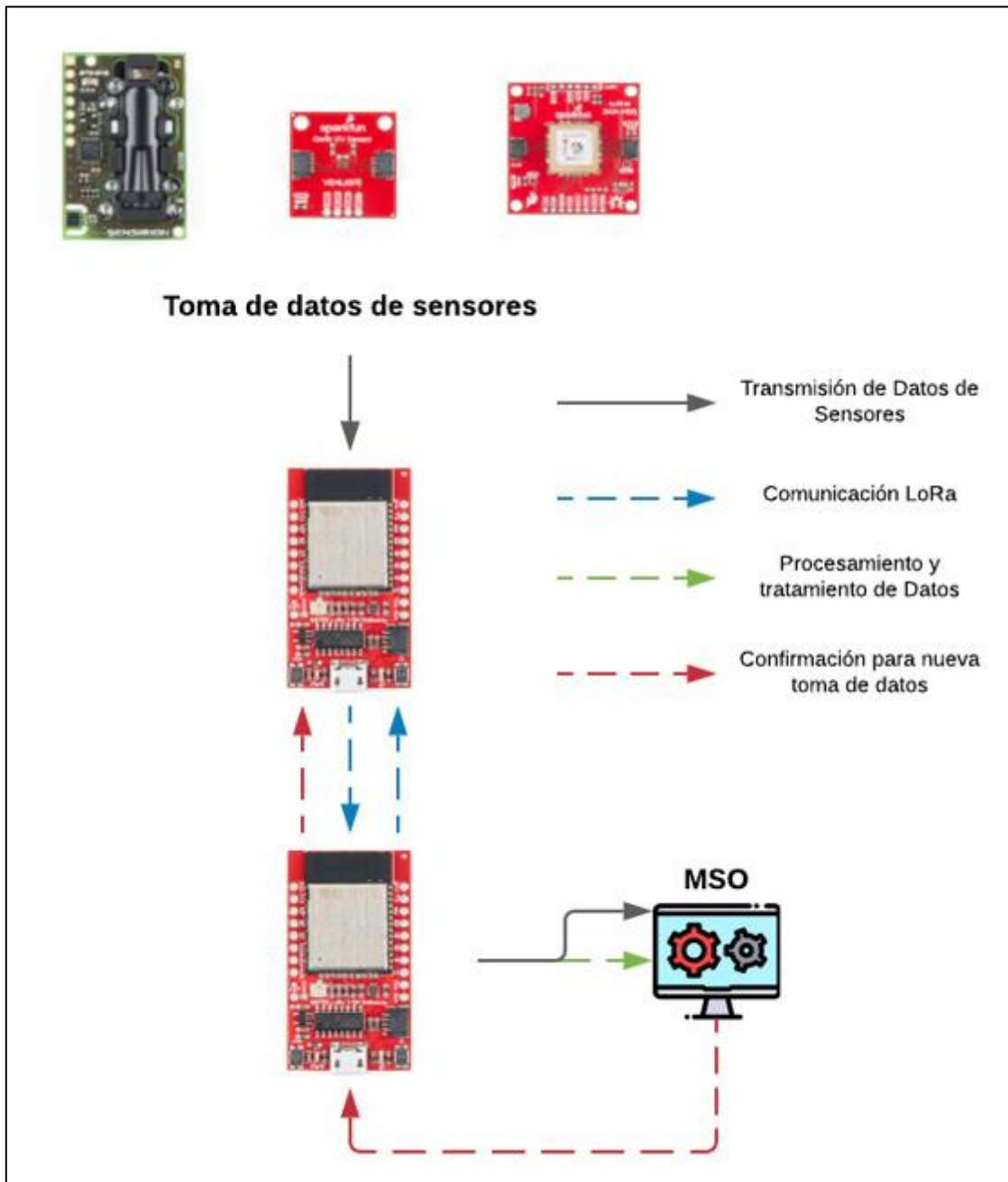


Figura 44. Diagrama de Bloques Funcionamiento

- **Etapa 1-Toma de datos de los sensores:**

La red de sensores es la que permite que se puedan monitorear las variables dentro del invernadero, por lo que deben tener la configuración adecuada para obtener una toma de datos sumamente precisa, el encargado de recopilar los datos de los sensores es el módulo LoRa-Gateway, el mismo que utilizando los pines I2C adquiere las variables tomadas por los sensores como los rayos uv, temperatura, humedad, CO2, los cuales serán enviados mediante la tecnología inalámbrica LoRa posteriormente. En la figura 45 se muestra el diagrama de bloques del funcionamiento de la toma de datos de los sensores.

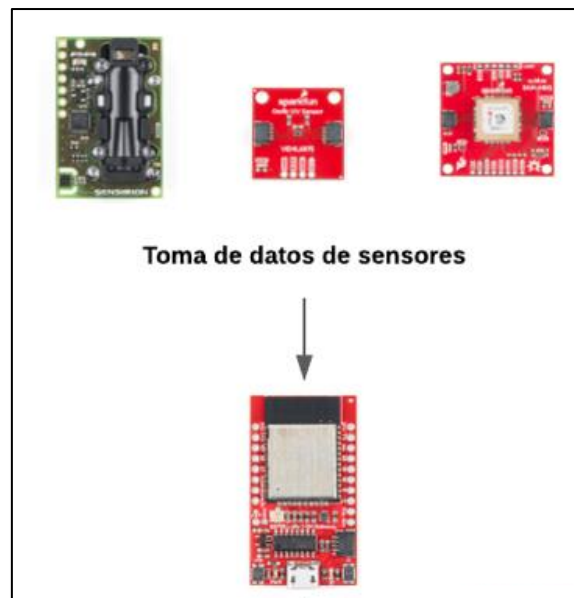


Figura 45. Diagrama de Bloques Toma de datos de sensores.

- **Etapa 2- Comunicación Inalámbrica:**

En la figura 46, una vez obtenidos los datos de los sensores, se tiene la trama de datos que van a ser enviados mediante tecnología inalámbrica LoRa hacia el nodo servidor para que sean procesados directamente en el Meta Sistema Operativo y una vez se procese la información, retorne la confirmación igualmente mediante tecnología LoRa para realizar una nueva toma de datos.

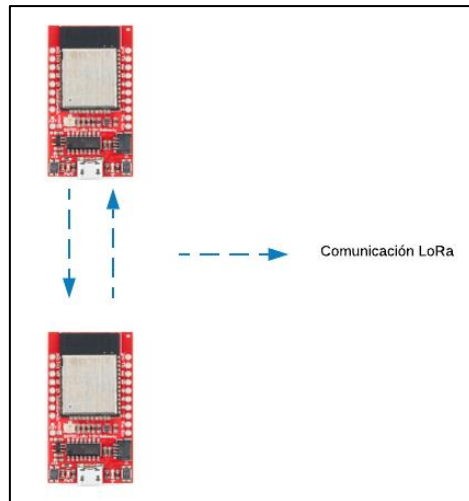


Figura 46. Diagrama de Bloques Comunicación Inalámbrica.

- **Etapa 3-Procesamiento y almacenamiento:**

Como se detalla en la figura 47, Una vez se obtengan los datos inalámbricamente mediante el módulo LoRa, se enviará mediante Comunicación Serial hacia el MSO quien se encargará de procesar la información obtenida de los sensores y posteriormente ejecutará las acciones necesarias con respecto a las condiciones dentro del invernadero, las cuales nos indica el estado en el que se encuentran los cultivos, en el que una vez finalizado el procesamiento, se almacena la información en una base de datos y se envía un dato de retorno hacia el módulo LoRa-Gateway para así confirmar la recepción y procesamientos de datos y de esta manera se pueda enviar nuevos datos para su debido tratamiento.

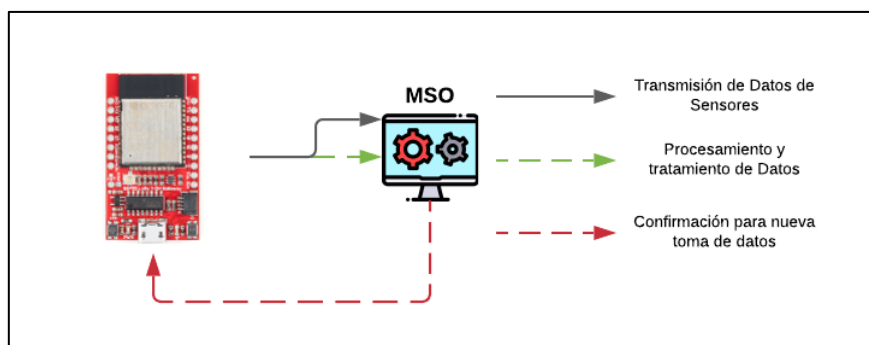


Figura 47. Diagrama de Bloques de Procesamiento y almacenamiento.

CAPITULO IV. IMPLEMENTACIÓN Y PRUEBAS

En este capítulo se desarrolla la implementación de la red, iniciando con los pasos a seguir para la toma de datos de cada sensor y su comunicación mediante I2C, seguido de la configuración básica de los módulos LoRaWAN; además se detalla todo el proceso para la ejecución del hardware, con pruebas de cada nodo del cultivo de pimiento correspondiente a la WSN para finalmente visualizar los datos monitoreados del sistema.

4.1. FASE 3: Implementación de la red

Al terminar con el diseño de la red, se procede a realizar la implementación de esta, iniciando por la toma de datos y calibración de los sensores junto con el diagrama de flujo para conocer su funcionamiento en este sistema, posteriormente se establece la comunicación inalámbrica entre nodos y la puerta de enlace, los datos serán procesados en el Meta Sistema Operativo y la información será mostrada en una interfaz gráfica.

4.1.1. Configuración placas SparkFun

Se obtiene como resultado que el SparkFun LoRa Gateway 1-Channel es reconocible en el computador a través del puerto COM3. En la Figura 48, se muestra el reconocimiento del módulo en la PC a través del IDE de Arduino, se aprecian sus características como: el tipo de placa, el tamaño de la memoria, la frecuencia, la velocidad de subida y el puerto de comunicación.

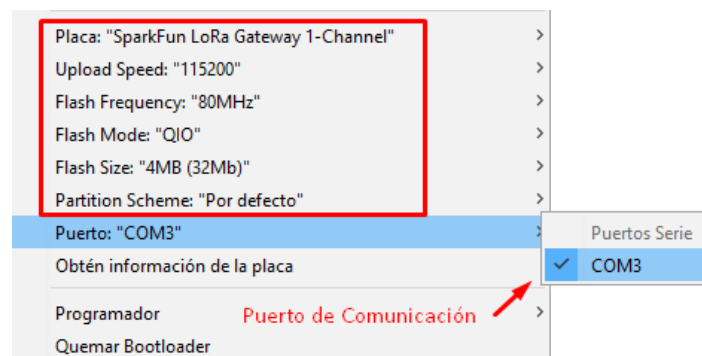
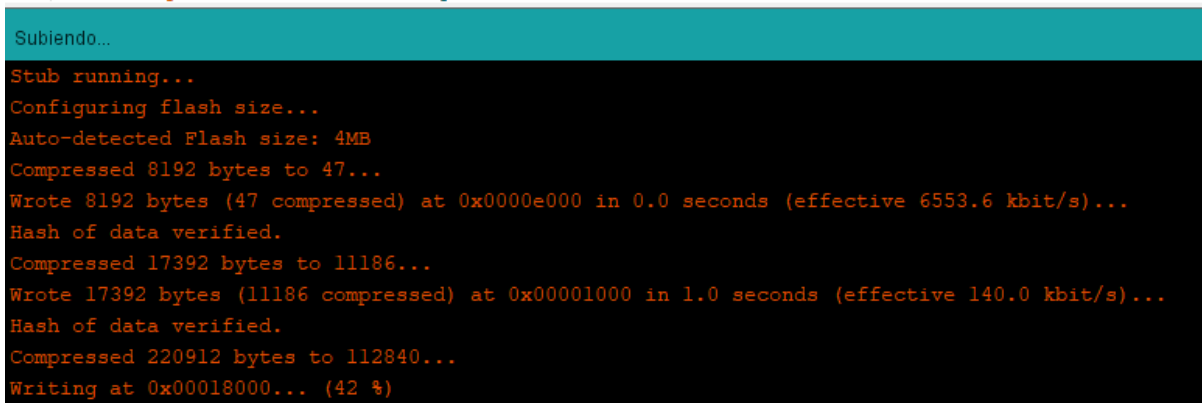


Figura 48. Reconocimiento del SparkFun LoRa Gateway 1-Channel en la PC

Se verifica la carga de datos en el SparkFun LoRa Gateway 1-Channel así como se muestra en la Figura 49 la cual es diferente a la carga de un Sketch en Arduino ya que en este caso se muestra el progreso de carga en porcentajes además de mostrar los bytes escritos en el controlador.

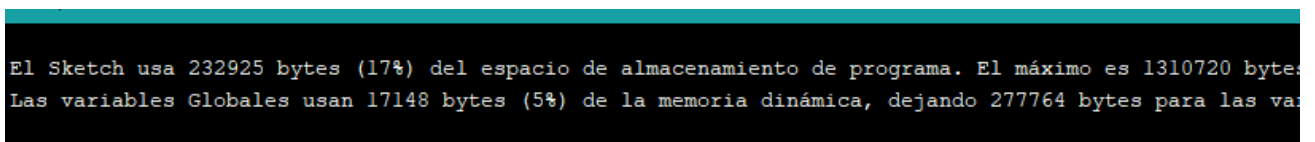


```

Subiendo...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 6553.6 kbit/s)...
Hash of data verified.
Compressed 17392 bytes to 11186...
Wrote 17392 bytes (11186 compressed) at 0x00001000 in 1.0 seconds (effective 140.0 kbit/s)...
Hash of data verified.
Compressed 220912 bytes to 112840...
Writing at 0x00018000... (42 %)
  
```

Figura 49. Carga exitosa del Sketch al SparkFun LoRa Gateway 1-Channel

Se observa en la Figura 50 el sketch usa tan solo el 17% de memoria y el 5% de la memoria dinámica rangos aceptables por lo que se concluye que el sistema funcionará sin ningún inconveniente. El *sketch* no ocupa demasiada memoria así que su rendimiento es satisfactorio.



```

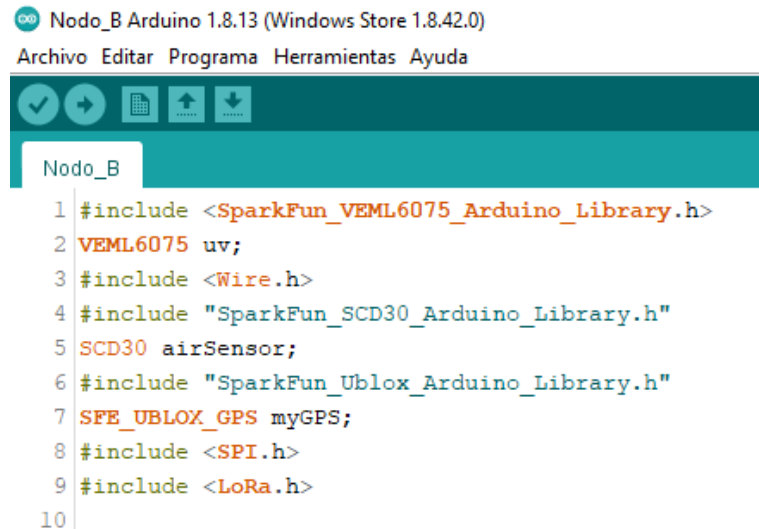
El Sketch usa 232925 bytes (17%) del espacio de almacenamiento de programa. El máximo es 1310720 bytes.
Las variables Globales usan 17148 bytes (5%) de la memoria dinámica, dejando 277764 bytes para las variables
  
```

Figura 50. Uso de memoria del SparkFun LoRa Gateway 1-Channel

4.1.2. Configuración de sensores

Los sensores son los encargados de medir las variables presentes en el cultivo de la planta de pimiento, por lo que deben contar con librerías especiales y compatibles con la IDE Arduino para que de esta manera se muestren los datos correctamente.

En la Figura 51 se muestra la declaración de variables para que cada uno de los sensores pueda ser reconocido por la placa y posteriormente logre tomar los datos del cultivo.



```

Nodo_B
1 #include <SparkFun_VEML6075_Arduino_Library.h>
2 VEML6075 uv;
3 #include <Wire.h>
4 #include "SparkFun_SCD30_Arduino_Library.h"
5 SCD30 airSensor;
6 #include "SparkFun_Ublox_Arduino_Library.h"
7 SFE_UBLOX_GPS myGPS;
8 #include <SPI.h>
9 #include <LoRa.h>
10

```

Figura 51. Declaración de librerías de los sensores

Como resultado se obtiene que todos los sensores funcionan correctamente, se evidencia su encendido por el led indicador del VEML6075, el SCD30 y el SAM-M8Q ya que los sensores cuentan con leds indicadores que nos muestran si están tomando correctamente los datos.

En la Figura 52 se aprecia la lectura exitosa de los sensores de rayos uv, de temperatura, de humedad, de CO2 y las coordenadas GPS, verificándose el correcto funcionamiento de cada sensor.

```

Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2 (ppm):383 temp (C):23.1 humidity(%):60.5
Lat: 3593222 Long: -781241154 (degrees * 10^-7) Alt: 2229156 (mm) SIV: 11
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2 (ppm):382 temp (C):23.2 humidity(%):60.6
Lat: 3593244 Long: -781241095 (degrees * 10^-7) Alt: 2228207 (mm) SIV: 10
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2 (ppm):381 temp (C):23.2 humidity(%):60.5
Lat: 3593263 Long: -781241040 (degrees * 10^-7) Alt: 2228072 (mm) SIV: 10

```

Figura 52. Lectura de sensores

Se verificó la funcionalidad del sistema, se comprobó que el SparkFun LoRa Gateway 1-Channel puede alimentar perfectamente a los sensores y la corriente es suficiente para este

propósito. Los sensores de rayos uv, de humedad, de temperatura, el sensor de CO2 y el módulo GPS miden adecuadamente las variables. La velocidad de transmisión es de 115000 baudios.

Luego se procede a verificar la calibración de los sensores para lo cual se debe acceder a sitios web certificados que muestran los valores tomados por los sensores y se realiza una comparativa de estos datos con los valores que toman los sensores. En la Figura 53 se verifica que la medición del sensor SCD30 es correcta ya que la aplicación marca 23°C y la medición del sensor es 23°C.

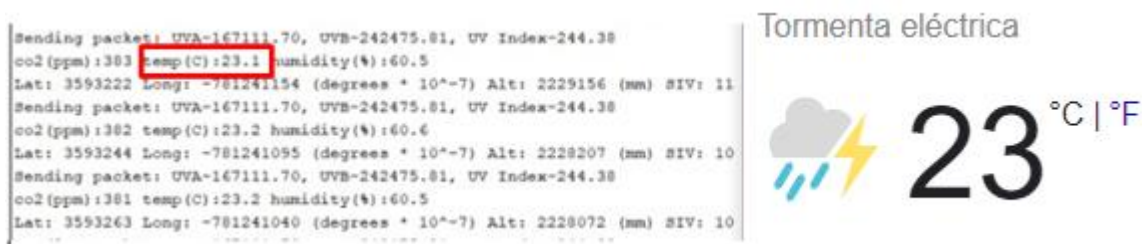


Figura 53. Verificación de Temperatura con la página web www.wheater.com

Para calibrar el sensor GPS, se usa la página <https://www.coordenadas-gps.com/>; se verifica que el valor medido por el sensor es bastante aproximado a la posición que indica la página, así como se muestra en la Figura 54.

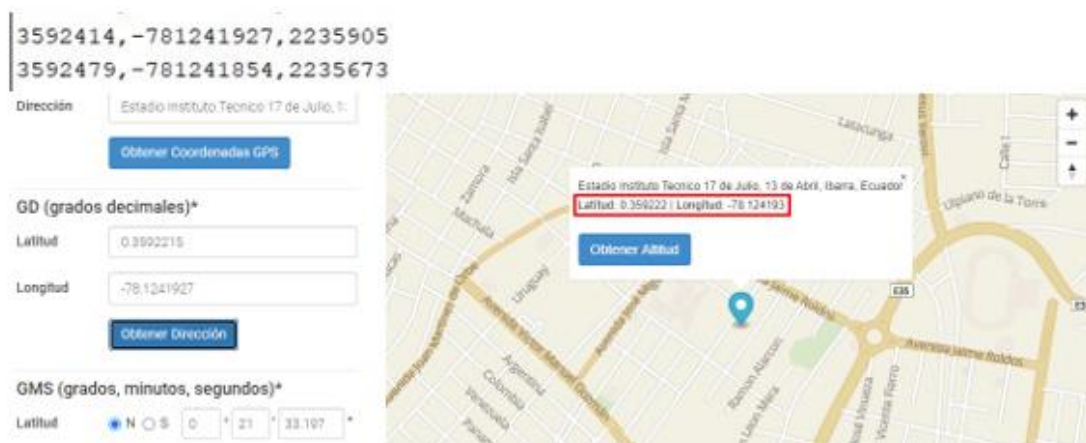


Figura 54. Verificación de coordenadas GPS con la página web [www.https://www.coordenadas-gps.com](https://www.coordenadas-gps.com)

Para calibrar el sensor de luz uv es simple, lo que se hace es medir la intensidad lumínica cuando no hay incidencia de luz midiendo un valor de cero y un valor alto cuando incida luz sobre el dispositivo, el sensor si trabaja según las características mencionadas verificándose así su correcto funcionamiento.

Para calibrar el sensor de humedad del suelo, se toma sus valores analógicos siendo 1023 el máximo y 0 el mínimo; entonces para detectar el porcentaje de humedad del suelo se aplica la siguiente fórmula correspondiente a la Ecuación 5.

$$Humedad = -1 * (100 * \left(\frac{\text{valor sensor}}{1023}\right) - 100) \quad (\text{Ecuación 5})$$

Para calibrar el sensor de CO2, se toma sus valores analógicos y se compara con la curva característica del sensor que se detalla su Datasheet, se lo puede apreciar en la Ecuación 6.

$$ppm = a * x^b \quad (\text{Ecuación 6})$$

- En donde: a y b constantes
- x = valor medido

Se compara la ecuación con la gráfica y se obtiene el valor de las constantes: $a = 100,0482$ $b = -3.1041$ entonces la fórmula para calcular el valor de CO2 en ppm está dada por la expresión de la Ecuación 7.

$$ppm = 100.0482 * \sqrt[-3.1041]{CO2/383} \quad (\text{Ecuación 7})$$

4.1.3. Configuración de módulos LoRaWAN RF95

La Figura 55, muestra los comandos de configuración básica con los que se pudo establecer la comunicación inalámbrica entre los nodos del sistema con el uso de módulos LoRaWAN

RF95, lo cual permitió establecer una comunicación entre los nodos y la puerta de enlace. A continuación, se explica la manera en la que se procede a conectar y configurar los módulos RF95, primero, el módulo debe tener una dirección local y posteriormente se ingresa las direcciones de destino, una vez configuradas las direcciones en las que se van a enviar los mensajes se procede a configurar la frecuencia con la que se va a trabajar la cual es de 915 MHz.

Además, si se desea conocer más acerca de los comandos y su función en los módulos RF95, el ANEXO 1, muestra un manual de configuraciones.

```

byte localAddress = 0xAA; ← Dirección Puerta de enlace
byte destinationAddress = 0xBB; ← Dirección Nodo 1
byte destination2Address = 0xCC; ← Dirección Nodo 2
long lastSendTime = 0;
int interval = 2000;
int count = 0;

void setup() {
  Serial.begin(115200);
  Serial.println("Start LoRa duplex");

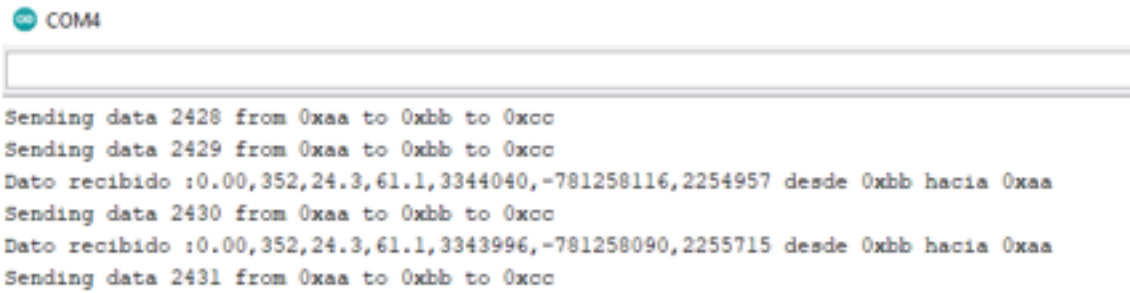
  LoRa.setPins(csPin, resetPin, irqPin);

  if (!LoRa.begin(915E6)) { ← Frecuencia módulo LoRa
    Serial.println("LoRa init failed. Check your connections.");
    while (true) {}
  }
}

```

Figura 55. Configuración de Direcciones y Frecuencia de operación

Al utilizar un método de acceso al medio el cual nos permite que los nodos envíen los datos hacia la puerta de enlace, se obtiene una comunicación con mensajes que nos garantiza que todos los datos enviados son recibidos correctamente y si algún dato se llega a perder, el método automáticamente solicita nuevamente se envíe la información, hasta garantizar que todos los datos enviados son correctamente recibidos y no exista pérdida de información. En la Figura 56. Se muestra el intercambio de mensajes entre los nodos y la puerta de enlace.



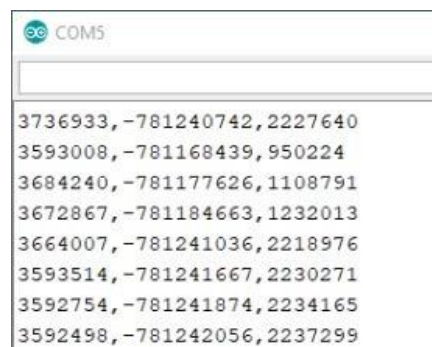
```

COM4
Sending data 2428 from 0xaa to 0xbb to 0xcc
Sending data 2429 from 0xaa to 0xbb to 0xcc
Dato recibido :0.00,352,24.3,61.1,3344040,-781258116,2254957 desde 0xbb hacia 0xaa
Sending data 2430 from 0xaa to 0xbb to 0xcc
Dato recibido :0.00,352,24.3,61.1,3343996,-781258090,2255715 desde 0xbb hacia 0xaa
Sending data 2431 from 0xaa to 0xbb to 0xcc

```

Figura 56. Intercambio de mensajes entre nodos

Una vez establecido el intercambio de mensajes inalámbricos entre los nodos y la puerta de enlace, la Figura 57. Muestra el envío de las coordenadas GPS mediante comunicación Serial entre el módulo SparkFun LoRa Gateway 1-Channel y el Arduino Nano correspondiente al Roster, para que de esta manera el vehículo autónomo logre ubicarse al interior del invernadero y se logre la toma de datos.



```

COM5
3736933,-781240742,2227640
3593008,-781168439,950224
3684240,-781177626,1108791
3672867,-781184663,1232013
3664007,-781241036,2218976
3593514,-781241667,2230271
3592754,-781241874,2234165
3592498,-781242056,2237299

```

Figura 57. Envío de coordenadas GPS

La Figura 58. muestra la recepción de los datos en la puerta de enlace utilizando la comunicación inalámbrica de LoRaWAN teniendo resultados satisfactorios.

```

COM9
Received packet 'valores sensores'
UVA-167111.70, UVB25.6, UV Index58.4
co2 (ppm):685 temp (C):25.6 humidity(%):58.4

' with RSSI -103
Received packet 'Valores sensores'
UVA-167111.70, UVB25.7, UV Index58.4
co2 (ppm):686 temp (C):25.7 humidity(%):58.4

' with RSSI -103
Received packet 'Valores sensores'
UVA-167111.70, UVB25.6, UV Index58.3
co2 (ppm):686 temp (C):25.6 humidity(%):58.3

```

Figura 58. Recepción de información mediante LoRa

4.1.4. Configuración Meta Sistema Operativo

Se procede a realizar el almacenamiento y visualización de los datos a través de la plataforma Anaconda Navigator que es una interfaz gráfica de usuario GUI bastante sencilla y con un gran potencial. Puede gestionar de manera avanzada paquetes relacionados a la ciencia de datos con Python desde la terminal, permite compilar Python en código de máquina para una ejecución rápida. Ahora, ya estando dentro de la plataforma se procede a realizar las configuraciones básicas para visualizar los datos provenientes del nodo central. Como se muestra en la Figura 59.

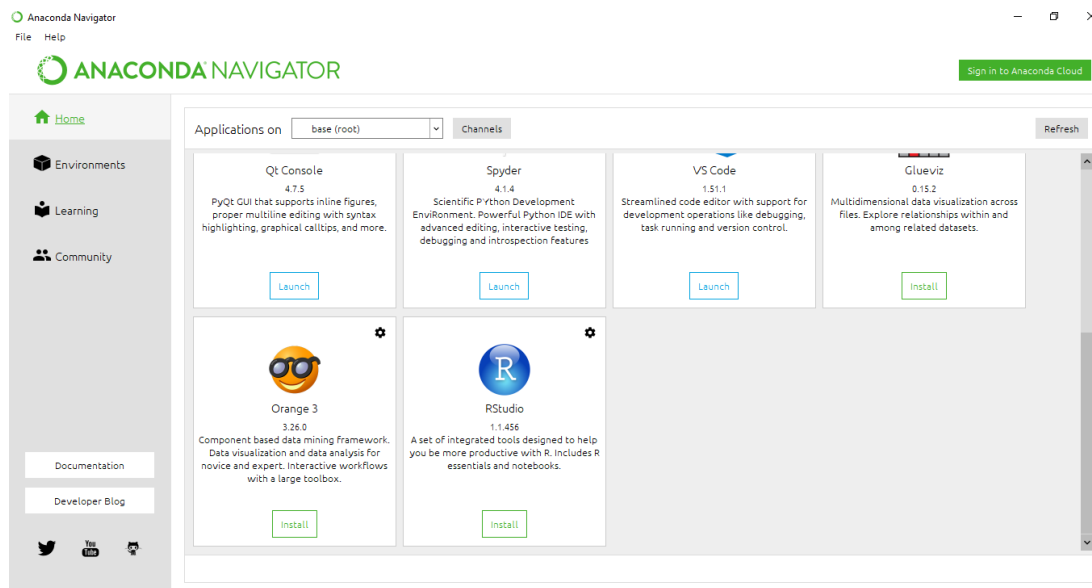


Figura 59. Interfaz Anaconda Navigator

Una vez instalado *Anaconda Navigator*, la programación de la interfaz gráfica se la realiza en el lenguaje *Python* la cual nos va a permitir recibir los mensajes, imprimirlos y mostrarlos, así se garantizará que el administrador del invernadero pueda conocer las variables obtenidas en el cultivo, en la Figura 60 se puede observar un fragmento de la programación dentro de *Python*.

```

28 #Titulo
29 label1=Label(root,text="UTN-CIERCOM",font=("Times New Roman",16)).pack()
30 label11=Label(root,text="Monitoreo de cultivos",font=("Agency FB",14)).pack()
31 label12=Label(root,text="Interfaz de control",font=("Agency FB",14)).pack()
32 ##Comunicacion
33 def encender():
34     labelEncender=Label(root,text="Sistema encendido",background = '#4FF719',font=("Times New Roman",12)).place(
35
36 def prender():
37     labelPrender=Label(root,text="Si envía datos",background = '#4FF719',font=("Times New Roman",12)).place(x=200,y=100)
38 label14=Label(root,text="Conexión",font=("Agency FB",14)).place(x=170,y=100)
39 btnCx=Button(root,text="Envío",command=encender,font=("Times New Roman",12)).place(x=90,y=150)
40 btnEstado=Button(root,text="Estado WSN",command=prender,font=("Times New Roman",12),width=10).place(x=90,y=200)
41 ##Control
42 label15=Label(root,text="Control",font=("Agency FB",14)).place(x=600,y=90)
43
44 #menu
45 n = tk.StringVar()
46 menu1=tk.Combobox(root, width = 20,textvariable = n)
47 menu1['values'] = ('UV',

```

Figura 60. Programación interfaz gráfica en Visual Studio Code

4.1.5. Integración del Nodo 1 en un vehículo

Se ha integrado el nodo 1 en un vehículo autónomo preprogramado que se desplaza en un área de terreno en base al sistema global de posicionamiento GPS, las coordenadas son enviadas hacia un Arduino uno quien procesa el movimiento de esta manera se logra contar con un nodo móvil que se desplace por un terreno de acuerdo con coordenadas geográficas de un sector determinado. La investigación que se desarrolló para que el vehículo se desplace fue realizada por Cristian Padilla, estudiante de ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte y las pruebas de funcionamiento se han desarrollado de manera experimental en uno de los invernaderos ubicados en el sector de Piquiucho.

En la Figura 61 se aprecia el Nodo sobre el vehículo autónomo antes mencionado donde se observa los sensores de Uv, CO2, Temperatura, Humedad y el módulo GPS.



Figura 61. Nodo 1 sobre vehículo autónomo.

4.2. FASE 4: Verificación

Una vez culminada la implementación de los nodos, ahora se muestra las pruebas realizadas en el invernadero. Las pruebas presentadas a continuación fueron realizadas en el día, en la tarde y en la noche para verificar si el proceso de sensado del sistema es el adecuado para esas condiciones.

4.2.1. Pruebas en el invernadero

El cultivo de pimiento como se muestra en la Figura 62, se encuentra en el invernadero “El Rosal”, con una extensión de 300 m², únicamente de dicho cultivo. Como se investigó en el capítulo II, se debe cumplir con los valores óptimos para el desarrollo adecuado del cultivo de pimiento, siendo estos los siguientes:

- Rayos UV: 440 y 680 lux.
- Temperatura: Superior a los 15°C.

- Humedad: Entre el 50 % y el 70 %.
- CO2: 800 ppm.



Figura 62. Vehículo autónomo dentro del invernadero.

4.2.2. Pruebas comunicación inalámbrica

Utilizando un analizador de espectro se evidencia la frecuencia en la que están operando los módulos LoRa RMF95, tal como se muestra en la Figura 63, la frecuencia en la que está trabajando el módulo que es 915 MHz.

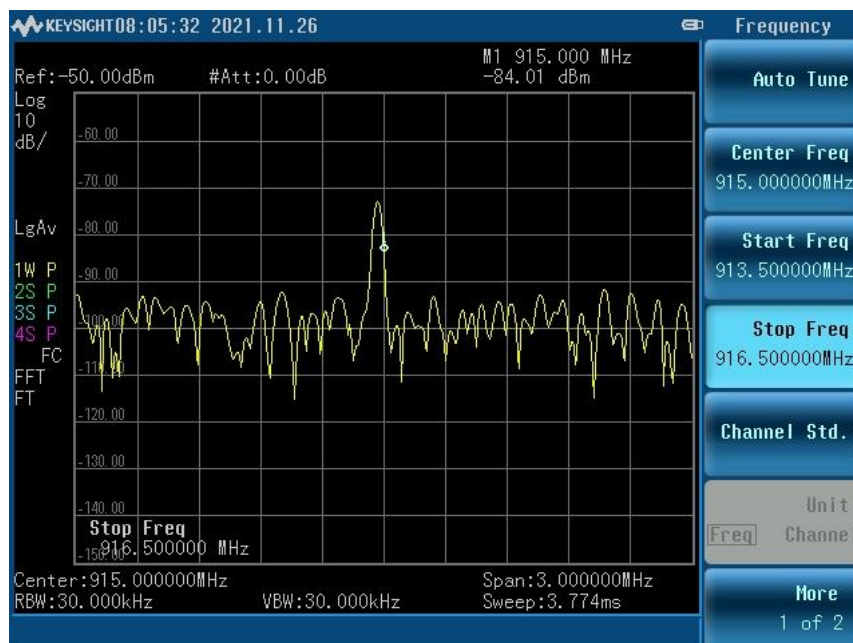


Figura 63. Frecuencia LoRa.

En la Figura 64, se indica el muestreo del espectro, en donde las zonas con color dentro del recuadro verde, muestran en que frecuencia estuvo la mayor parte la potencia de la señal. Como se observa, ésta se desplaza y brinca dentro de la zona limitada por el ancho de banda.

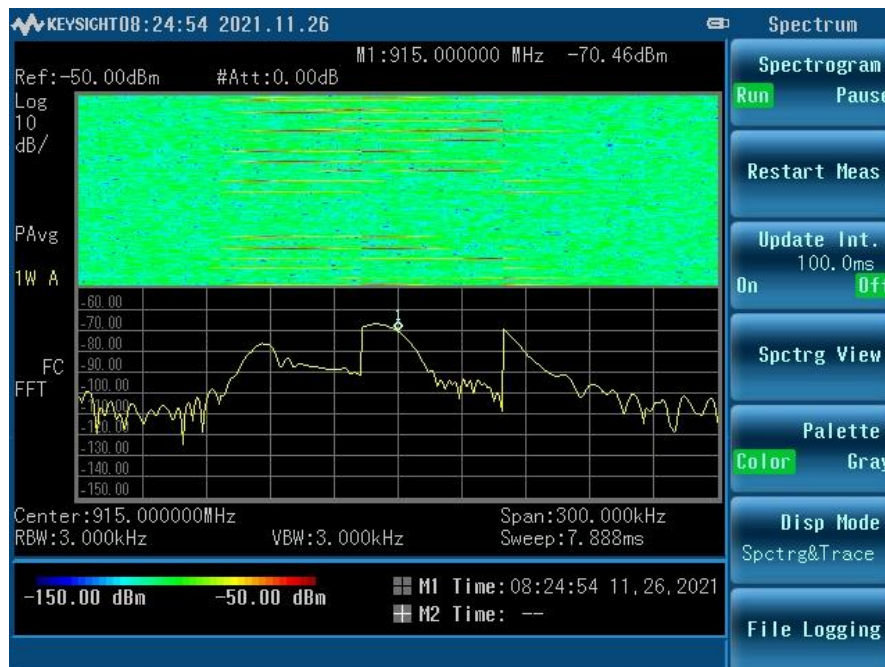


Figura 64. Muestreo espectro LoRa.

Se realizaron pruebas de alcance de los módulos LoRa modificando el SF (Factor de dispersión), con el cual se determina el número de símbolos utilizados para codificar los datos de los sensores. Como se muestra en la Figura 65, la prueba de distancia fue realizada en el Autódromo Internacional de Yahuarcocha, en donde se realizaron un total de 6 pruebas modificando la distancia y el factor de dispersión.



Figura 65. Distancias de toma de datos.

En la tabla 24, se muestran las distancias de la toma de datos de los sensores y el factor de dispersión empleado, obteniendo como resultado que con el SF7 se logra más transacciones por segundo posibles con un menor consumo de energía aplicándolo para cortas distancias. Por otra parte, con el SF11 se apreció un largo tiempo de transmisión lo que implica una sola transacción cada segundo, lo que conlleva a un mayor consumo de energía, pero se obtiene un mayor alcance.

Tabla 24. Factor de Dispersión empleado

SF	Distancia (Nodo)	Tiempo de Transmisión
SF7	80 m	120 ms
SF11		1320 ms
SF7	150m	120 ms
SF11		1320 ms
SF7	280 m	120 ms
SF11		1320 ms

En conclusión, para trabajar en estos invernaderos se debe utilizar el SF7, debido a que el alcance probado es de 100 m a 500 m, además se tiene un corto tiempo de transmisión con respecto al otro SF y con esto se logra más tomas de datos por segundo, lo que implica que se puede obtener mayor número de muestras de datos de los sensores por cada segundo, con lo que se obtiene una mayor cobertura que la tecnología Wi-Fi que comprende 90 m aproximadamente, la distancia entre los nodos y puerta de enlace comprende una distancia de 250m con lo que no hay inconvenientes al aplicar el factor de dispersión seleccionado.

4.2.3. Lectura del Sensado

Para el monitoreo de los patrones dentro del cultivo se ha diseñado una interfaz gráfica en la que se puede acceder a toda la información acerca del cultivo, la cual consta de tres secciones: en la primera sección denominada “Conexión”, la cual consta de dos botones. El primero al presionarlo indica si el sistema esta correctamente encendido y al presionar el segundo botón tenemos la información acerca del estado de la WSN el cual informa si se estableció correctamente el envío de datos. En la segunda sección denominada “Control” se tiene un menú el cual nos permite seleccionar un determinado sensor para de esta manera al presionar el botón graficar los datos sensados. Finalmente, en la sección “Análisis” se muestra una gráfica del proceso de aprendizaje de máquina denominado clustering, el cual por medio de agrupamiento permite obtener los patrones adecuados de crecimiento del cultivo. La Figura 66 muestra la interfaz gráfica final.

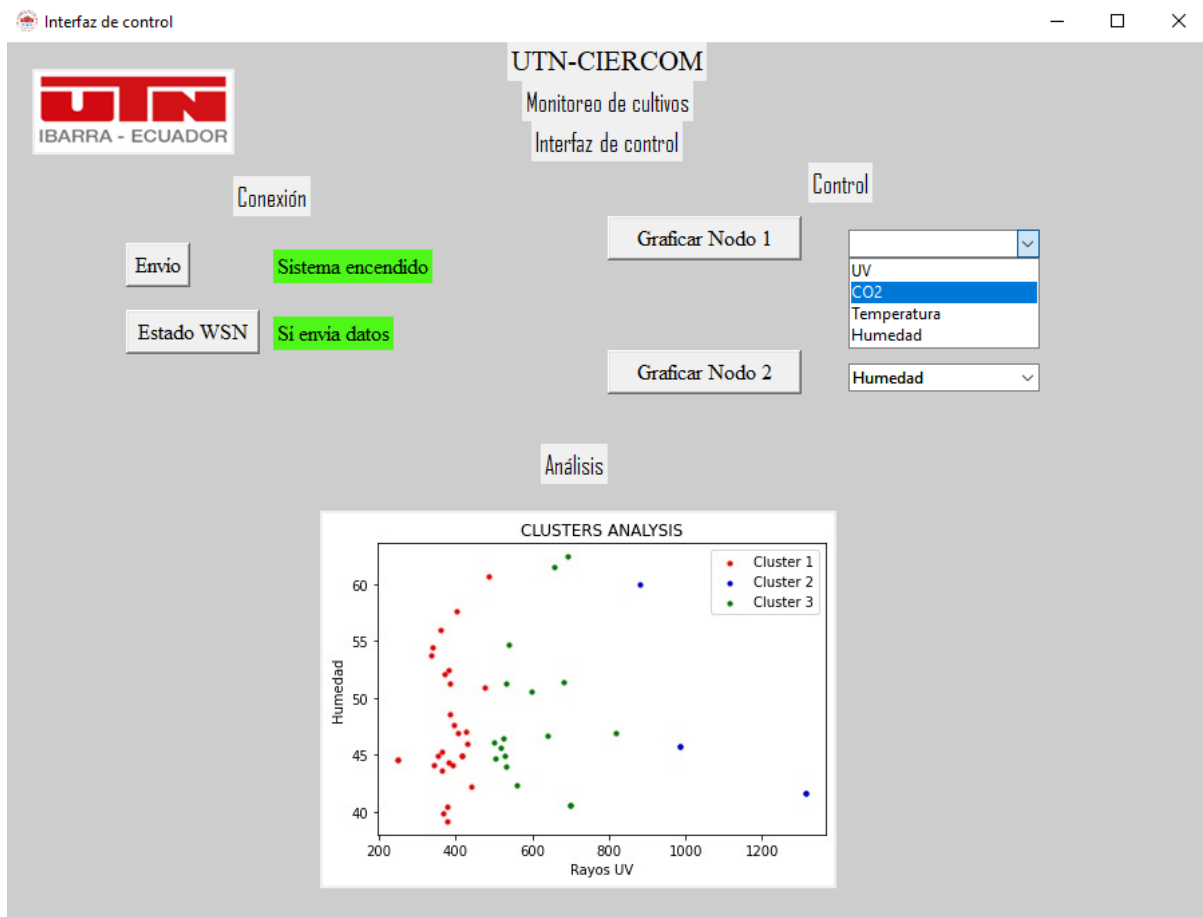


Figura 66. Interfaz gráfica.

Los sensores debidamente calibrados recolectan los datos de cada parámetro a monitorear, los módulos LoRaWAN permiten la comunicación con el nodo central, el cual se encarga de enviar los datos al MSO para visualizarlos, identificando cada uno de los sensores, por su nombre como se puede mostrar en la Figura 67.

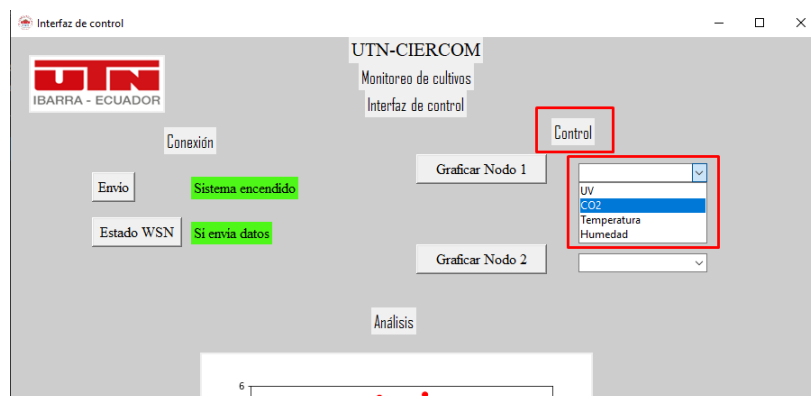


Figura 67. Datos de las variables de los nodos.

Del mismo modo, para una mejor visualización se realizan graficas de cada parámetro sentido en el cultivo. Para acceder a la gráfica se debe dirigir al menú y seleccionar el sensor del cual se necesita la información y se presiona el botón. La Figura 68, muestra la gráfica del índice UV en los cultivos.

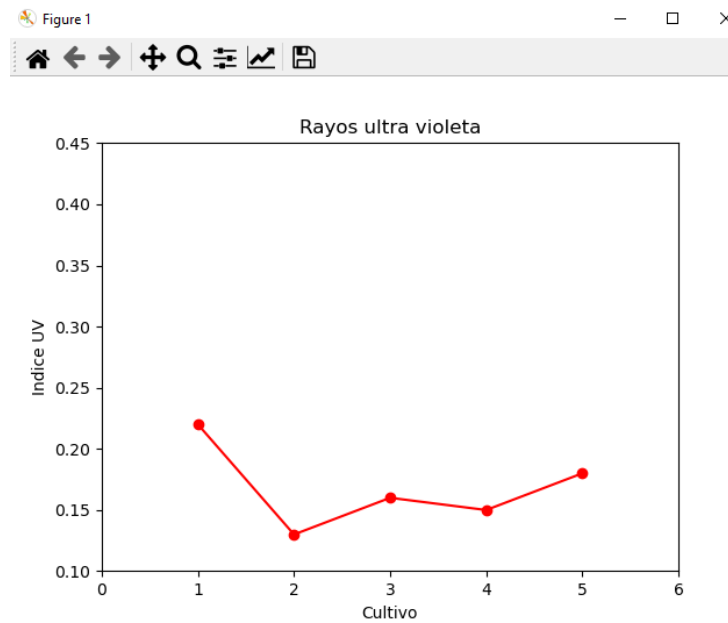


Figura 68. Datos de la variable UV.

En la Figura 69, se muestra el sensado de “CO2” en los cultivos que se encuentran dentro del invernadero.

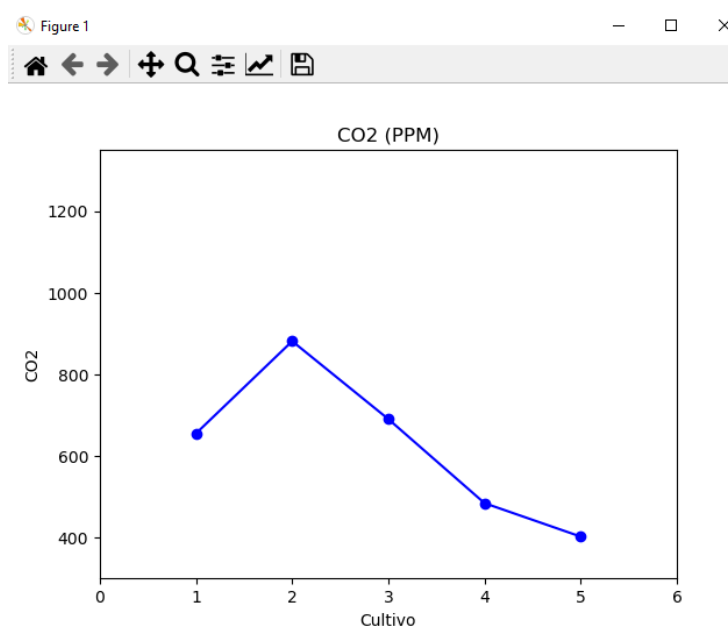


Figura 69. Datos de la variable CO2.

En la Figura 70, se muestra la gráfica correspondiente a la “Temperatura” en los cultivos que se encuentran dentro del invernadero.

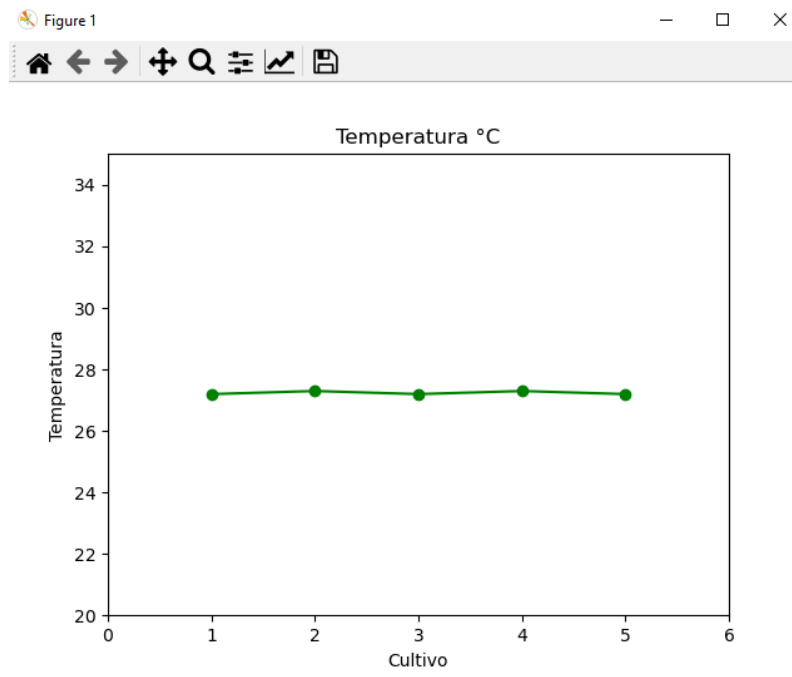


Figura 70. Datos de la variable Temperatura.

En la Figura 71, se muestra el monitoreo de la “Humedad” en los cultivos que se encuentran dentro del invernadero.

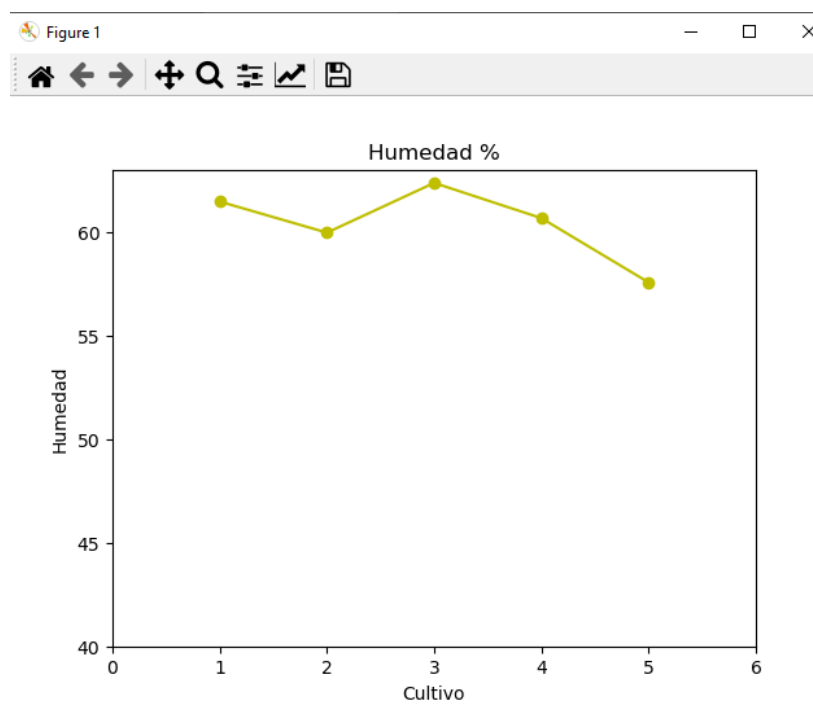


Figura 71. Datos de la variable Humedad.

4.2.4. Patrones de crecimiento

Utilizando el análisis de los clusters se procede a agrupar todos los datos de los sensores en una sola matriz, la cual será analizada por este algoritmo de aprendizaje de máquina para poder obtener los parámetros necesarios para el desarrollo del cultivo. La Figura 72 muestra el análisis que se realizó a los datos del cultivo durante la mañana en el que reconoce como patrón para su correcto desarrollo la relación entre ciertos niveles CO₂ y la humedad presente en el cultivo.

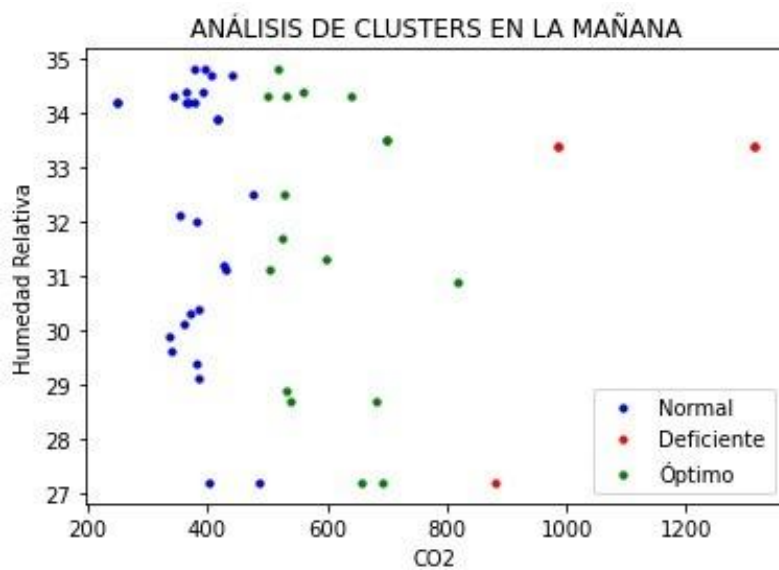


Figura 72. Relación entre CO₂ y humedad.

De la misma manera se analizó en base a la matriz agrupada de valores de los sensores que existe una relación entre el CO₂ y la temperatura en el interior del invernadero para lograr un correcto desarrollo del cultivo como se muestra en la Figura 73.

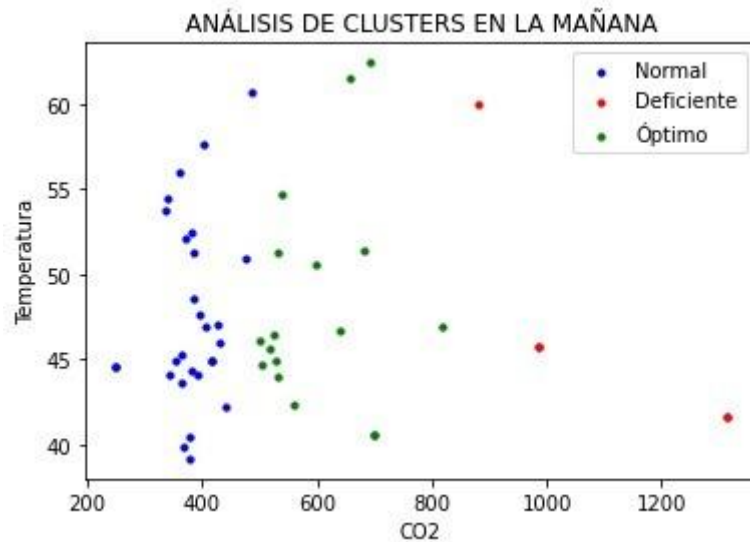


Figura 73. Relación entre CO2 y temperatura.

Con los datos finales de los clusters seleccionados, se determinó tres etiquetas para los datos entre los cuales tenemos: etiqueta “2” representada por el color verde la cual nos indica que los niveles de los sensores son los adecuados, la etiqueta “0” que se muestra en color amarillo representa niveles normales y finalmente la etiqueta “1” en color rojo la cual indica que está mal. La Figura 74 indica el mapa del invernadero con los datos tomados en la mañana.

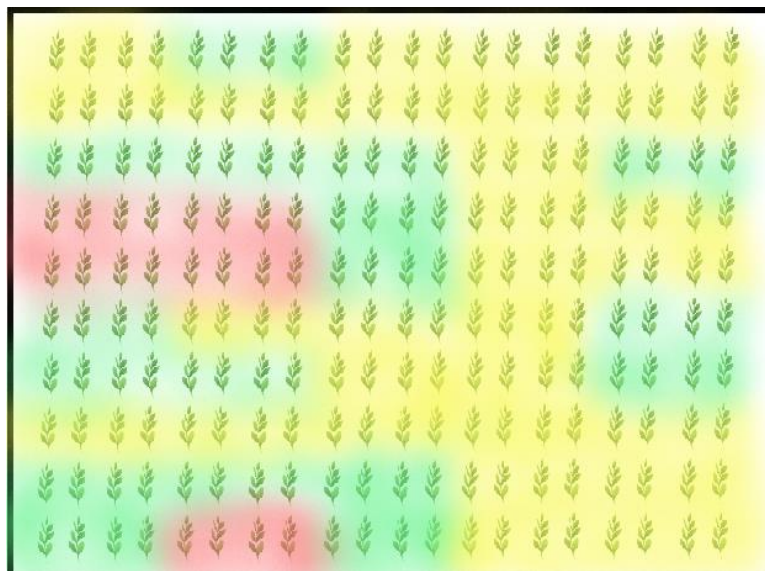


Figura 74. Datos en la mañana.

La Figura 75 muestra el mapa de colores del cultivo dentro del invernadero con las muestras tomadas en la tarde.

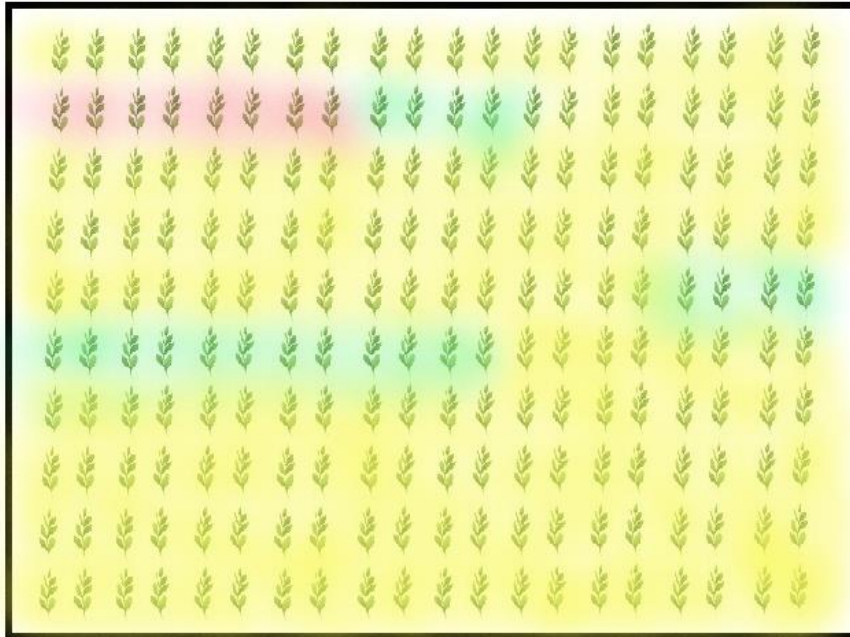


Figura 75. Datos en la tarde.

4.3. Costo del sistema

En esta sección se presenta el costo de cada uno de los componentes y elementos que conforman el sistema de monitoreo por medio de una red inalámbrica de sensores. El costo se refiere a la suma económica de cada uno de los componentes del sistema, teniendo en cuenta los componentes de hardware, software, infraestructura y trabajo de ingeniería como se detallan a continuación.

4.3.1. Costo de Hardware

En la Tabla 25, se muestra el costo de los elementos del hardware que constituyen el sistema de monitoreo, como son los nodos sensores, nodo central y la alimentación eléctrica.

Tabla 25. Costos de Hardware

Equipos	Cantidad	Precio Unitario (USD)	Subtotal (USD)
Sparkfun LoRa Gateway	3	\$50.00	\$150.00
Sensor VEML6075	2	\$10.00	\$20.00
Sensor SCD30	2	\$80.00	\$160.00
GPS SAM-M8Q	2	\$50.00	\$100.00
Raspberry Pi Zero	2	\$40.00	\$80.00
Batería de litio	2	\$30.00	\$60.00
Cables	4	\$3	\$12.00
Elementos electrónicos varios	1	\$30.00	\$30.00
Total			\$612.00

4.3.2. Costo de infraestructura

En este apartado se toma en cuenta los costos de infraestructura como son: las bases de los nodos sensores y la puerta de enlace como se muestra en la Tabla 26.

Tabla 26. Costos de Infraestructura

Infraestructura	Cantidad	Precio Unitario (USD)	Subtotal
Soportes acrílico	2	\$10.00	\$20.00
Tornillos y extras	1	\$30.00	\$30.00
Total			\$50.00

4.3.3. Costo de Ingeniería

Para los costos de ingeniería se toman en cuenta los honorarios de la persona encargada del diseño del sistema, estudio de campo y documentación. La Tabla 27 muestra los honorarios del diseñador del sistema.

Tabla 27. Costos de Ingeniería

Ingeniería	Cantidad	Precio Unitario (USD)	Subtotal
Estudio de campo, diseño y revisión de la infraestructura	1	\$650.00	\$650.00
Documentación	1	\$250.00	\$250.00
Total			\$900.00

4.3.4. Costo de Software

En la Tabla 28, se muestra el costo de software equivalente a 0.00 USD, debido a que los programas usados en todo el sistema se basan en Arquitectura Open Source de acceso libre.

Tabla 28. Costos de Software

Software	Cantidad	Precio (USD)	Unitario	Subtotal
IDE de Arduino	1	\$0.00		\$0.00
Raspbian	1	\$0.00		\$0.00
Python	1	\$0.00		\$0.00
Total				\$0.00

4.3.5. Costo de implementación del sistema general

En la Tabla 29, se muestra los costos de inversión total del sistema, el cual se obtiene de la suma de los costos que interceden en la implementación

Tabla 29. Costos de implementación del sistema

Descripción	Subtotal
Costos de Hardware	\$612.00
Costos de Infraestructura	\$50.00
Costos de Ingeniería	\$900.00
Costos de Software	\$0.00
Total	\$1562.00

4.4. Beneficios del sistema

A continuación, se definen los beneficios que resultan de la implementación del sistema de monitoreo de cultivos para de esta manera obtener patrones adecuados de crecimiento.

- La implementación de este tipo de sistema en el invernadero hace que aumente considerablemente la producción del cultivo, además de aportar con mayores beneficios económicos para el dueño del invernadero.
- El sistema en su totalidad se lo realizó con elementos aptos para el monitoreo en la agricultura y de bajo costo, haciéndolo un sistema eficiente.
- La utilización del sistema permite a la persona que está encargada del invernadero obtener mejoras al momento de tomar decisiones basándose en los datos monitoreados por parte de los nodos con sensores.
- El sistema cuenta con una interfaz gráfica para la visualización de los datos de los sensores que son mostrados en graficas fáciles de interpretar por parte de la persona que lo use.
- Al obtener parámetros adecuados de crecimiento que mejoran los cultivos dentro de un invernadero implementando este tipo de monitoreo, la tecnología LoRaWAN podrá ser

utilizada por parte de las personas que se dedican a la agricultura ya que su cobertura es sumamente amplia.

4.5. Conclusiones

- Con la revisión bibliográfica se pudo establecer las condiciones ambientales que intervienen en el crecimiento del cultivo de pimiento, dando de esta manera parámetros que influyeron en el desarrollo del sistema, así como varios requerimientos técnicos del mismo.
- El desarrollo de sistemas electrónicos utilizando hardware y software libre permite crear dispositivos que ayuden al sector de la agricultura, promoviendo de esta manera la agricultura inteligente en el Ecuador, beneficiando de esta manera a los propietarios de invernaderos y lo más conveniente es que se obtiene dispositivos de costo muy reducido.
- El desarrollo de este proyecto se lo hizo con tecnología LoRaWAN que tradicionalmente es costosa, pero con el estudio de mercado realizado, se optó por adquirir los productos de Sparkfun ya que tienen dispositivos con todos los parámetros requeridos y con un menor costo.
- El Sparkfun LoRa Gateway es un poderoso dispositivo capaz de enviar datos mediante LoRaWAN, logra conectarse a una red wifi, posee comunicación I2C con la cual se permite integrar múltiples sensores sin ocupar demasiados pines lo que lo hace ideal para desarrollar aplicaciones relacionadas al internet de las cosas.
- Los nodos distribuidos en el invernadero son imperceptibles para los agricultores permitiendo de esta manera el desarrollo de otras actividades sin ningún problema, lo que convierte al sistema en un dispositivo factible para implementar.

- Al realizar las pruebas de funcionamiento en el invernadero utilizando el monitoreo de variables, se obtuvo patrones de crecimiento de las plantas, lo cual permitirá tener una base para el desarrollo de los nuevos cultivos.
- Se logró cumplir con todos los objetivos de este proyecto de titulación de manera satisfactoria, ya que se cuenta con un sistema montado sobre un vehículo autónomo formando un Nodo móvil que recorrerá en su totalidad el invernadero monitoreando y recolectando datos de los cultivos para enviarlos mediante tecnología LoRaWAN al Meta Sistema Operativo el cual procesará la información y la mostrará a través de la interfaz gráfica.

4.6. Recomendaciones

- Los módulos LoRaWAN poseen un alcance teórico de 10km, al realizar las pruebas de cobertura se logró una distancia sumamente menor al hacerlo en un ambiente urbano, se podría lograr tener más cobertura si se reemplaza las antenas normales por unas con mucho más ganancia.
- Si se desea tener mucho más rapidez en el procesamiento de la información es recomendable adquirir un Raspberry más potente, con las implicaciones de costo que esto demanda.
- Se recomienda fabricar cases de materiales resistentes y económicos, para abaratar costos, y porque al ser un sistema expuesto a un ambiente externo, mantendrá a los dispositivos a una temperatura adecuada para un correcto funcionamiento, además de protegerlos de cualquier fenómeno que pueda llegar a dañarlos.
- Se debe verificar el estado de la batería para evitar de esta manera que el sistema se apague, es recomendable tener un sistema de batería de respaldo para alimentar el sistema en caso de alguna emergencia.

- Para el uso y calibración de los sensores se debe tener en cuenta los Datasheet de cada uno, reconocer sus valores de medición máximo y mínimo, conocer adecuadamente sus pines de conexión para de esta manera lograr obtener datos lo más reales posibles.
- Antes de realizar la manipulación de los componentes electrónicos, realizar la carga de nuevo código en las placas, es recomendable hacerlo cuando el sistema esté apagado, esto evitara las variaciones de voltaje y corriente y evitará que se dañen los componentes electrónicos.

REFERENCIAS

- Aguado da Costa, J. (2012). *Desarrollo de un sistema automatizado para un invernadero*. <http://uvadoc.uva.es/handle/10324/2930>
- Allience, L. (2015). *A technical overview of LoRa and LoRaWAN*.
- Arduino, S. A. (2015). Arduino. *Arduino LLC*.
- Bor, M. C., Roedig, U., Voigt, T., & Alonso, J. M. (2016). Do LoRa low-power wide-area networks scale? *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 59–67.
- Bottrighi, A., & Terenziani, P. (2016). META-GLARE: A meta-system for defining your own computer interpretable guideline system—Architecture and acquisition. *Artificial Intelligence in Medicine*, 72, 22–41. <https://doi.org/10.1016/J.ARTMED.2016.07.002>
- Bustos, A. (2018). *Diseño e implementación de una placa de desarrollo para aplicaciones de Nodos LoRaWAN*. 68. <https://www.researchgate.net/publication/327118250%0Ahttp://hdl.handle.net/11349/13875>
- Camargo Bareño, C. I. (2011). *Transferencia tecnológica y de conocimientos en el diseño de sistemas embebidos*. Universidad Nacional de Colombia.
- Canamero, M. ; Helfgott, S. ; & Laguna, T. (n.d.). *Aplicaciones de la agricultura de precisión*. Retrieved September 16, 2020, from <http://kgoteo.com/ap3.htm>
- Castells, D., Galin, A., & Carrabina, J. (2018, November 30). *The Regulation of Unlicensed Sub-GHz bands: Are Stronger Restrictions Required for LPWAN-based IoT Success?* . <https://arxiv.org/ftp/arxiv/papers/1812/1812.00031.pdf>
- César, A., & Álvarez, B. (2006). *Pimiento para pimentón en Santa María: Alternativas de Riego*. http://www.cappama.org.ar/descargas/estudioRIEGO_pimenton_CATA%0Amrca.pdf
- Córdoba, D. M. A., & Buitrago, F. A. S. (2013). Estado del arte de las redes de sensores inalámbricos. *Tecnología Investigación y Academia*, 1(2).
- De Poorter, E., Hoebeke, J., Strobbe, M., Moerman, I., Latré, S., Weyn, M., Lannoo, B., & Famaey, J. (2017). Sub-GHz LPWAN Network Coexistence, Management and Virtualization: An Overview and Open Research Challenges. *Wireless Personal Communications*, 95(1), 187–213. <https://doi.org/10.1007/s11277-017-4419-5>
- Diosdado, R. (2018). *Zona Maker - Introducción a Raspberry*. <https://www.zonamaker.com/raspberry/raspberry-pi/intro-raspberry>
- Domínguez, A. (2020). *Diseño de una red de sensores inalámbricos LPWAN para el monitoreo de cultivos y materia orgánica en la granja experimental La Pradera de la Universidad Técnica del Norte*. <http://repositorio.utn.edu.ec/handle/123456789/10297>
- ECOAGRICULTOR. (2014). *El cultivo del pimiento*. <https://www.ecoagricultor.com/el-cultivo-del-pimiento/>
- Ertürk, M. A., Aydın, M. A., Büyükakkaşlar, M. T., & Evirgen, H. (2019). A Survey on LoRaWAN Architecture, Protocol and Technologies. *Future Internet*, 11(10), 216.

<https://doi.org/10.3390/fi11100216>

- Fao. (2012). *ORAWRACION DEUS IUMEIHAMIAS liARMAALMENTACIONYLAMINCULTURA DIPECHO BINACIONAL PERO-BOLIVIA VII PLAN DE ACCIÓN.*
- FAO. (n.d.). *BUENAS PRÁCTICAS AGRÍCOLAS-BPA-EN LA PRODUCCIÓN DE TOMATE BAJO CONDICIONES PROTEGIDAS.*
- FAO. (2017, June 19). *BLOG: Smart Farming is key for the future of agriculture - Schuttelaar & Partners.* <https://www.schuttelaar-partners.com/news/2017/smart-farming-is-key-for-the-future-of-agriculture>
- Garber, L. (2013). Robot OS: A New Day for Robot Design. *Computer*, 46(12), 16–20. <https://doi.org/10.1109/MC.2013.434>
- García-Cervigón, D., & José, J. (2015). *Estudio de Índices de vegetación a partir de imágenes aéreas tomadas desde UAS/RPAS y aplicaciones de estos a la agricultura de precisión.*
- García, E., & Flego, F. (n.d.). *Agricultura de Precisión.*
- Gaybor Murillo, M. A., & Maridueña Chunga, M. D. (2018). *Diseño de un sistema de adquisición de datos de una red de sensores inalámbricos que miden variables oceanográficas en el perfil costanero de Santa Elena, usando tecnología LoRa.* Espol.
- Hernandez, S. ;, & Alvarez, A. (2012). *Cultivos Bajo Cubierta.* <https://agroinvernaderos.webnode.es/tipos-de-invernaderos/invernadero-dientes-de-sierra/>
- HortiCultivos. (2017). *La elección del invernadero depende del cultivo.* <https://www.horticultivos.com/agricultura-protegida/la-eleccion-del-invernadero-depende-del-cultivo/>
- Iacono, L., Godoy, P., Marianetti, O., Garino, C. G., & Párraga, C. (2012). Estudio de la Integración entre WSN y redes TCP/IP. *Memoria Investigaciones En Ingeniería*, 10, 57–68.
- Iglesias, A. (2018, October 8). *La Agricultura Inteligente basada en IoT es el futuro de la industria.* <https://bitcoin.es/actualidad/la-agricultura-inteligente-basada-en-iot-es-el-futuro-de-la-industria/>
- Infoagro. (2010, June 11). *Fotografía de invernadero plano.* <https://www.infoagro.com/galeria/imprimir.asp?id=355>
- InfoAgro. (n.d.). *Control climático en invernaderos.* http://www.infoagro.com/industria_auxiliar/control_climatico.htm
- Kodali, R. K., Soratkal, S., & Boppana, L. (2017). WSN in coffee cultivation. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, 661–666. <https://doi.org/10.1109/CCAA.2016.7813804>
- LoRa Alliance. (2015). *A technical overview of LoRa® and LoRaWAN™ What is it?*
- Ma, X., Fang, F., Qian, K., & Liang, C. (2018). Networked robot systems for indoor service enhanced via ROS middleware. *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 852–857. <https://doi.org/10.1109/ICIEA.2018.8397832>

- Mantovani, E., & Magdalena, C. (2014). *Manual de agricultura de precisión*.
- Martínez, D. (2011). *Efecto de cuatro bioestimulantes en el crecimiento y productividad del cultivo de pimiento (Capsicum annuum L.) variedad cacique en la zona de Chaltura provincia de Imbabura*. Universidad Técnica de Babahoyo.
- Mora, D. (2019). *Desempeño de una WSN y simulación de la pérdida de paquetes y latencia*. <http://repositorio.ucsg.edu.ec/bitstream/3317/12735/1/T-UCSG-PRE-TEC-ITEL-340.pdf>
- Moya, M. (2018). Evaluación de pasarela LoRa/LoRaWAN en entornos urbanos. *Máster Universitario En Tecnologías, Sistemas y Redes de Comunicaciones*.
- National Instruments. (n.d.). *¿Qué es Adquisición de Datos?* Retrieved March 22, 2020, from <https://www.ni.com/data-acquisition/what-is/esa/>
- Nayibe, C. C., Alexander, T. B. D., Cristina, A. Z. L., & Miguel, C. O. L. (2014). *Redes de Sensores Inalámbricos*. Universidad Nacional de Colombia, Bogotá.
- Noguera, L. (2020). *Sistema de alerta temprana de inundaciones mediante una red de sensores inalámbricos LPWAN dirigido a la parroquia de Andrade Marín perteneciente al cantón Antonio Ante*. <http://repositorio.utn.edu.ec/handle/123456789/10518>
- Novagric. (2015, June 22). *Comprar un Invernadero Túnel. Aspectos a tener en cuenta*. <https://www.novagric.com/es/blog/articulos/comprar-invernadero-tunel>
- Novagric. (2016). *Invernaderos Multitúnel Capilla - Multicapilla*. <https://www.novagric.com/es/venta-invernaderos-novedades/tipos-de-invernaderos/invernadero-capilla>
- Pérez, R. (2017). *Evaluación de LoRa/LoRaWAN para escenarios de Smart City*.
- Pinto, M. (2013). *El cultivo del pimiento y el clima en el Ecuador*. [http://186.42.174.231/meteorologia/articulos/agrometeorologia/El cultivo %0Adel pimiento y el clima en el Ecuador.pdf](http://186.42.174.231/meteorologia/articulos/agrometeorologia/El_cultivo_%0Adel_pimiento_y_el_clima_en_el_Ecuador.pdf)
- Plataforma de desarrollo en hardware libre. (2012). *Hacia una definición de Hardware Libre, Centro Nacional de Desarrollo e Investigación en Tecnologías Libres*. <http://hl.cenditel.gob.ve/intro/definicion/>
- Productores de Hortalizas. (2004). *Plagas y enfermedades de chiles y pimientos*. www.hortalizas.com
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., & Ng, A. (2009). *ROS: an open-source Robot Operating System*. <http://stair.stanford.edu>
- Ramírez, L. G. C., Jiménez, G. S. A., & Carreño, J. M. (2014). *Sensores y actuadores*. Grupo Editorial Patria.
- Raspberry. (n.d.). *Raspberry Pi Foundation - About Us*. Retrieved March 26, 2020, from <https://www.raspberrypi.org/about/>
- Reche, J. (2010). *Cultivo de pimiento dulce en invernadero*. http://www.juntadeandalucia.es/export/drupaljda/1337160265Cultivo_Pimiento_Inver%0Anadero.pdf
- Robyns, P., Quax, P., Lamotte, W., & Thenaers, W. (2018). A Multi-Channel Software

Decoder for the LoRa Modulation Scheme. *IoTBDs*, 41–51.

- Rufepa. (2020). *INVERNADERO LLAVE EN MANO*. <http://www.rufepa.com/invernadero/familia-productos/1/invernaderos/1/invernaderos-de-plastico/81/invernadero-tipo-parral-o-almeria.html>
- Runberg, D. (2015). *The SparkFun Guide to Processing: Create Interactive Art with Code*. No Starch Press.
- Sacaleanu, D. I., Popescu, R., Manciu, I. P., & Perisoara, L. A. (2019, April 1). Data Compression in Wireless Sensor Nodes with LoRa. *Proceedings of the 10th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2018*. <https://doi.org/10.1109/ECAI.2018.8679003>
- SEMTECH. (n.d.). *Semtech LoRa Technology Overview*. Retrieved April 4, 2020, from <https://www.semtech.com/lora>
- Serrano, Z. (2011). *Prontuario del cultivo de pimiento*. [http://www.zoiloserrano.com/wpcontent/uploads/2012/03/Tráiler del %0Aibro Prontuario del cultivo de pimiento..pdf](http://www.zoiloserrano.com/wpcontent/uploads/2012/03/Tráiler%20del%20Prontuario%20del%20cultivo%20de%20pimiento.pdf)
- Sparkfun. (n.d.-a). *CO₂ Humidity and Temperature Sensor - SCD30*. Retrieved October 5, 2020, from <https://www.sparkfun.com/products/15112>
- Sparkfun. (n.d.-b). *SparkFun GPS Breakout - Chip Antenna, SAM-M8Q (Qwiic)*. Retrieved October 5, 2020, from <https://www.sparkfun.com/products/15210>
- Sparkfun. (n.d.-c). *SparkFun LoRa Gateway - 1-Channel (ESP32)*. Retrieved March 26, 2020, from <https://www.sparkfun.com/products/15006>
- Sparkfun. (n.d.-d). *SparkFun UV Light Sensor Breakout - VEML6075 (Qwiic)*. Retrieved October 5, 2020, from <https://www.sparkfun.com/products/15089>
- Técnico, I., & Agrónomo, A. I. (2010). *José Reche Mármol Cultivo del Pimiento dulce en invernadero*.
- Tzivaras, V. (2017). *Raspberry Pi Zero W Wireless Projects*. Packt Publishing Ltd.
- UNICEN. (n.d.). *Pimiento: Un cultivo muy productivo bajo invernadero*. <http://www.faa.unicen.edu.ar/crescaa/Pimiento.pdf>
- Vega, A. M. (2014). *Mantenimiento y manejo de invernaderos*. Ediciones Paraninfo, SA.
- Vu, V. A., Cong Trinh, D., Truvant, T. C., & Dang Bui, T. (2018). Design of automatic irrigation system for greenhouse based on LoRa technology. *International Conference on Advanced Technologies for Communications, 2018-October*, 72–77. <https://doi.org/10.1109/ATC.2018.8587487>
- Wordpress. (2017). *Topología estrella – Clasificación De Las Redes*. <https://clasificaciondelasredesblog.wordpress.com/2017/05/09/topologia-estrella/>

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

LPWAN: Low Power Wide Area Network (Red de área amplia y baja potencia). Es una red óptima para dispositivos de conexión que necesitan enviar pequeñas cantidades de datos a largas distancias (10 a 15km en zonas rurales y 1 a 5km en zonas urbanas), con la ventaja que la duración de la batería es larga (hasta 15 años).

LoRa: Long Range (Largo Alcance). Es una tecnología de comunicación inalámbrica diseñada para interconectar dispositivos que requieren bajo consumo de energía a velocidades muy bajas (hasta 50 Kbps).

WSN: Wireless Sensor Network (Redes de sensores inalámbricos). Realiza la adquisición y tratamiento de datos con múltiples aplicaciones en distintos campos tales como entornos industriales, domótica, entornos militares y detección ambiental.

IoT: Internet of Things (Internet de las Cosas). Es un concepto que se refiere a la interconexión digital de objetos con internet. Alternativamente, IoT es el punto en el tiempo en el que se conectarían a internet más “objetos” que personas.

UART: Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal). Por medio de este dispositivo se realiza la comunicación serial entre dispositivos PC con alguna placa en particular.

CSS: Chirp Spread Spectrum (Espectro Expandido Chirp). Es una técnica que usa chirp (pulsos) modulados de frecuencia lineal de banda ancha para codificar información.

SF: Spreading Factor (Factor de Dispersión). Es la cantidad de chirps por símbolo utilizados en el tratamiento de la información antes de la transmisión de la señal.

PCB: Printed Circuit Board (Placa de circuito impreso). Es una zona constituida por caminos o pistas de material conductor laminadas sobre una base no conductora.

ANEXOS

ANEXO 1: Datasheet Modulo Rf95



RFM95/96/97/98 (W)

RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0

GENERAL DESCRIPTION

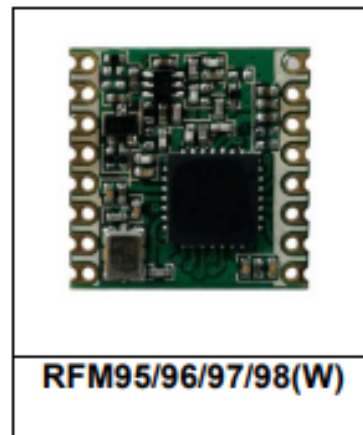
The RFM95/96/97/98(W) transceivers feature the LoRa™ long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

Using Hope RF's patented LoRa™ modulation technique RFM95/96/97/98(W) can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness. LoRa™ also provides significant advantages in both blocking and selectivity over conventional modulation techniques, solving the traditional design compromise between range, interference immunity and energy consumption.

These devices also support high performance (G)FSK modes for systems including WMBus, IEEE802.15.4g. The RFM95/96/97/98(W) deliver exceptional phase noise, selectivity, receiver linearity and IIP3 for significantly lower current consumption than competing devices.

KEY PRODUCT FEATURES

- ◆ LoRa™ Modem.
- ◆ 168 dB maximum link budget.
- ◆ +20 dBm - 100 mW constant RF output vs. V supply.
- ◆ +14 dBm high efficiency PA.
- ◆ Programmable bit rate up to 300 kbps.
- ◆ High sensitivity: down to -148 dBm.
- ◆ Bullet-proof front end: IIP3 = -12.5 dBm.
- ◆ Excellent blocking immunity.
- ◆ Low RX current of 10.3 mA, 200 nA register retention.
- ◆ Fully integrated synthesizer with a resolution of 61 Hz.
- ◆ FSK, GFSK, MSK, GMSK, LoRa™ and OOK modulation.
- ◆ Built-in bit synchronizer for clock recovery.
- ◆ Preamble detection.
- ◆ 127 dB Dynamic Range RSSI.
- ◆ Automatic RF Sense and CAD with ultra-fast AFC.
- ◆ Packet engine up to 256 bytes with CRC.
- ◆ Built-in temperature sensor and low battery indicator.
- ◆ Module Size: 16*16mm



APPLICATIONS

- ◆ Automated Meter Reading.
- ◆ Home and Building Automation.
- ◆ Wireless Alarm and Security Systems.
- ◆ Industrial Monitoring and Control
- ◆ Long range Irrigation Systems

1.2. Product Versions

The features of the three product variants are detailed in the following table.

Table 48 RFM95/96/97/98(W) Device Variants and Key Parameters

Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
RFM95W	868/915 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
RFM97W	868/915 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
RFM96W/RFM98W	433/470MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

1.3. Pin Diagram

The following diagram shows the pin arrangement , top view.

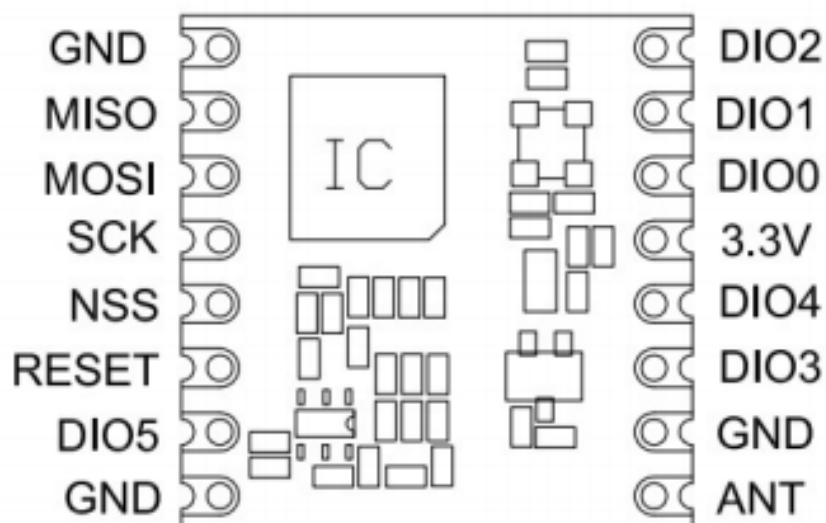


Figure 2. Pin Diagrams

1.4. Pin Description

Number	Name	Type	Description Description Stand Alone Mode
1	GND	-	Ground
2	MISO	I	SPI Data output
3	MOSI	O	SPI Data input
4	SCK	I	SPI Clock input
5	NSS	I	SPI Chip select input
6	RESET	I/O	Reset trigger input
7	DIO5	I/O	Digital I/O, software configured
8	GND	-	Ground
9	ANT	-	RF signal output/input.
10	GND	-	Ground
11	DIO3	I/O	Digital I/O, software configured
12	DIO4	I/O	Digital I/O, software configured
13	3.3V	-	Supply voltage
14	DIO0	I/O	Digital I/O, software configured
15	DIO1	I/O	Digital I/O, software configured
16	DIO2	I/O	Digital I/O, software configured

Anexo 2: Datasheet sensor UV VEML6075

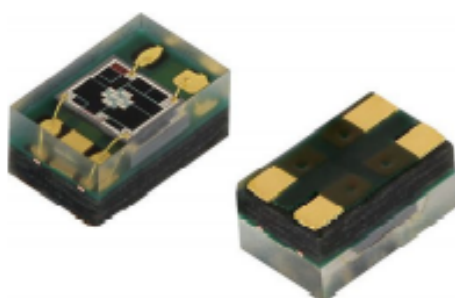


www.vishay.com

VEML6075

Vishay Semiconductors

UVA and UVB Light Sensor with I²C Interface



DESCRIPTION

The VEML6075 senses UVA and UVB light and incorporates photodiode, amplifiers, and analog / digital circuits into a single chip using a CMOS process. When the UV sensor is applied, it is able to detect UVA and UVB intensity to provide a measure of the signal strength as well as allowing for UVI measurement.

The VEML6075 provides excellent temperature compensation capability for keeping the output stable under changing temperature. VEML6075's functionality is easily operated via the simple command format of I²C (SMBus compatible) interface protocol. VEML6075's operating voltage ranges from 1.7 V to 3.6 V. VEML6075 is packaged in a lead (Pb)-free 4 pin OPLGA package which offers the best market-proven reliability.

FEATURES

- Package type: surface mount
- Dimensions (L x W x H in mm): 2.0 x 1.25 x 1.0
- Integrated modules: ultraviolet sensor (UV), and signal conditioning IC
- Converts solar UV light intensity to digital data
- Excellent UVA and UVB sensitivity
- Reliable performance of UV radiation measurement under long time solar UV exposure
- 16-bit resolution per channel
- UVA and UVB individual channel solution
- Low power consumption I²C protocol (SMBus compatible) interface
- Package: OPLGA
- Temperature compensation: -40 °C to +85 °C
- Output type: I²C bus
- Operation voltage: 1.7 V to 3.6 V
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912



RoHS
COMPLIANT
HALOGEN
FREE
GREEN
05-20090

APPLICATIONS

- Handheld device
- Notebook
- Consumer device
- Industrial and medical application

PRODUCT SUMMARY					
PART NUMBER	OPERATING VOLTAGE RANGE (V)	I ² C BUS VOLTAGE RANGE (V)	PEAK SENSITIVITY UVA, UVB (nm)	RANGE OF SPECTRAL BANDWIDTH $\lambda_{0.5}$ (nm)	OUTPUT CODE
VEML6075	1.7 to 3.6	1.7 to 3.6	385, 330	± 10	16 bit, I ² C

Note

⁽¹⁾ Adjustable through I²C interface

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	VOLUME ⁽¹⁾	REMARKS
VEML6075	Tape and reel	MOQ: 2500 pcs	2.0 mm x 1.25 mm x 1.0 mm

Note

⁽¹⁾ MOQ: minimum order quantity



www.vishay.com

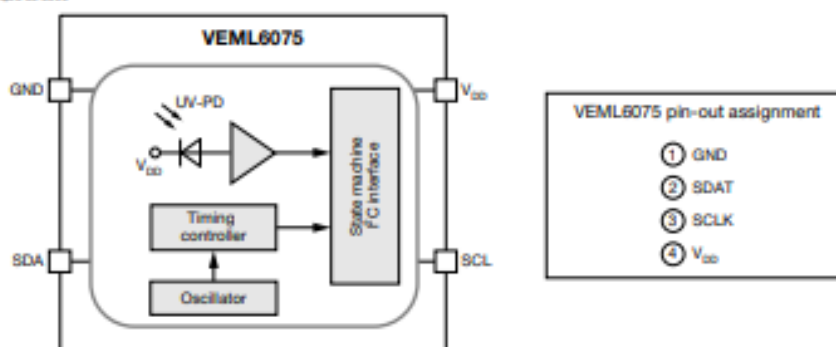
VEML6075

Vishay Semiconductors

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)					
PARAMETER	TEST CONDITION	SYMBOL	MIN.	MAX.	UNIT
Supply voltage		V_{DD}	0	3.6	V
Operation temperature range		T_{amb}	-40	+85	$^{\circ}\text{C}$
Storage temperature range		T_{stg}	-40	+85	$^{\circ}\text{C}$

RECOMMENDED OPERATING CONDITIONS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)					
PARAMETER	TEST CONDITION	SYMBOL	MIN.	MAX.	UNIT
Supply voltage		V_{DD}	1.7	3.6	V
Operation temperature range		T_{amb}	-40	+85	$^{\circ}\text{C}$
I ² C bus operating frequency		f_{I2CCLK}	10	400	kHz

PIN DESCRIPTIONS			
PIN ASSIGNMENT	SYMBOL	TYPE	FUNCTION
1	GND	I	Ground
2	SDAT	I / O (open drain)	I ² C data bus data input / output
3	SCLK	I	I ² C digital bus clock input
4	V_{DD}	I	Power supply input

BLOCK DIAGRAM

BASIC CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Supply operation voltage		V_{DD}	1.7	-	3.6	V
Supply current	$V_{DD} = 1.8\text{ V}$	I_{DD}	-	480	-	μA
I ² C signal input	Logic high	V_{IH}	1.5	-	-	V
	Logic low	V_{IL}	-	-	0.8	
I ² C signal output	Logic high	V_{OH}	1.4	-	-	V
	Logic low	V_{OL}	-	-	0.6	
Operating temperature		T_{amb}	-40	-	+85	$^{\circ}\text{C}$
Shutdown current	Light condition = dark; $V_{DD} = 1.8\text{ V}$, $T_{amb} = 25\text{ }^{\circ}\text{C}$	$I_{DD}(\text{SD})$	-	800	-	nA
UVA responsivity	$t_f = 50\text{ ms}^{(1)}$		-	0.93	-	counts/ $\mu\text{W}/\text{cm}^2$
UVB responsivity	$t_f = 50\text{ ms}^{(2)}$		-	2.1	-	counts/ $\mu\text{W}/\text{cm}^2$
Visible / infrared response	W-LED / IRED (940 nm)		-	0.5	< 1	%

Notes⁽¹⁾ Nichia NCSU033X (365 nm)⁽²⁾ LVTOP310TO39HS (315 nm)

Anexo 3: Datasheet sensor CO₂, Temperatura, Humedad SCD30



Datasheet Sensirion SCD30 Sensor Module CO₂, humidity, and temperature sensor

- NDIR CO₂ sensor technology
- Integrated temperature and humidity sensor
- Best performance-to-price ratio
- Dual-channel detection for superior stability
- Small form factor: 35 mm x 23 mm x 7 mm
- Measurement range: 400 ppm – 10.000 ppm
- Accuracy: $\pm(30 \text{ ppm} + 3\%)$
- Current consumption: 19 mA @ 1 meas. per 2 s.
- Energy consumption: 120 mJ @ 1 measurement
- Fully calibrated and linearized
- Digital interface UART or I²C



Product Summary

CMOSens® Technology for IR detection enables carbon dioxide measurements of the highest accuracy at a competitive price.

Along with the NDIR measurement technology for detecting CO₂ comes a best-in-class Sensirion humidity and temperature sensor integrated on the very same sensor module. Ambient humidity and temperature can be measured by Sensirion's algorithm expertise through modelling and compensating of external heat sources without the need of any additional components. The very small module height allows easy integration into different applications.

Carbon Dioxide is a key indicator for indoor air quality. Thanks to new energy standards and better insulation, houses have become increasingly energy-efficient, but the air quality can deteriorate rapidly. Active ventilation is needed to maintain a comfortable and healthy indoor environment, and improve the well-being and productivity of the inhabitants. Sensirion sensor solutions offer an accurate and stable monitoring of CO₂ in the air, as well as temperature and humidity. This enables our customers to develop new solutions that increase energy efficiency and simultaneously support the well-being of everyone.

Final Qualification results of SCD30 are still pending. Thus any specification is subject to change without prior notice.

Benefits of Sensirion's CMOSens® Technology

- High reliability and long-term stability
- Industry-proven technology with a track record of more than 10 years
- Designed for mass production
- Optimized for lowest cost
- Low signal noise

1 Sensor Specifications¹

Preliminary information. Subject to change without notice.

CO₂ Sensor Specifications

Parameter	Conditions	Value
CO ₂ measurement range	-	0 – 40'000 ppm
Accuracy ²	400 ppm – 10'000 ppm	± (30 ppm + 3%)
Repeatability ³	400 ppm – 10'000 ppm	± 10 ppm
Temperature stability ⁴	T = 0 ... 50°C, 400 ppm – 10'000 ppm	± 2.5 ppm / °C
Response time ⁵	T _{63%}	20 s
Accuracy drift over lifetime ⁶	400 ppm – 10'000 ppm ASC field-calibration algorithm activated	± 100 ppm

Table 1: SCD30 CO₂ sensor specifications

Humidity Sensor Specifications⁷

Parameter	Conditions	Value
Humidity measurement range	-	0 %RH – 100 %RH
Accuracy ⁸	25°C, 0 – 100 %RH	± 2 %RH
Repeatability ³	-	0.1 %RH
Response time ⁵	T _{63%}	8 s
Accuracy drift	-	< 0.25 %RH / year

Table 2: SCD30 humidity sensor specifications

Temperature Sensor Specifications⁷

Parameter	Conditions	Value
Temperature measurement range ⁹	-	- 40°C – 70°C
Accuracy ⁸	0 – 50°C	± 0.3°C
Repeatability ³	-	0.1°C
Response time ⁵	T _{63%}	> 2 s
Accuracy drift	-	< 0.03 °C / year

Table 3: SCD30 temperature sensor specifications

¹ Default conditions of T = 25°C, p = 1013 mbar, V_{CC} = 3.3 V, continuous measurement mode with measurement rate = 2 s apply to values listed in the tables, unless otherwise stated.

² Deviation to a high-precision reference. Accuracy is fulfilled by > 90% of the sensors after calibration. Rough handling, shipping and soldering can falsify the measurements. Accuracy is based on tests with gas mixtures having a tolerance of ± 1.5%.

³ RMS error of consecutive measurements at constant conditions. Repeatability is fulfilled by > 90% of the sensors.

⁴ Average slope of CO₂ accuracy when changing temperature, valid at 400 ppm. Fulfilled by > 90% of the sensors after calibration.

⁵ Time for achieving 63% of a respective step function. Response time depends on design-in, heat exchange and environment of the sensor in the final application.

⁶ CO₂ concentrations < 400 ppm may result in sensor drifts when ASC is activated. For proper function of ASC field-calibration algorithm SCD30 has to be exposed to air with CO₂ concentration 400 ppm regularly.

⁷ Design-in of the SCD30 in final application and the environment impacts the accuracy of the RH/T sensor. Heat sources have to be considered for optimal performance. Please use integrated on-board RH/T compensation algorithm to account for the actual design-in.

⁸ Deviation to a high-precision reference. Accuracy is fulfilled by > 90% of the sensors after calibration.

⁹ RH/T sensor component is capable of measuring up to T = 120°C. Measuring at T > 70°C might result in permanent damage of the sensor.

2 Package Outline Drawing

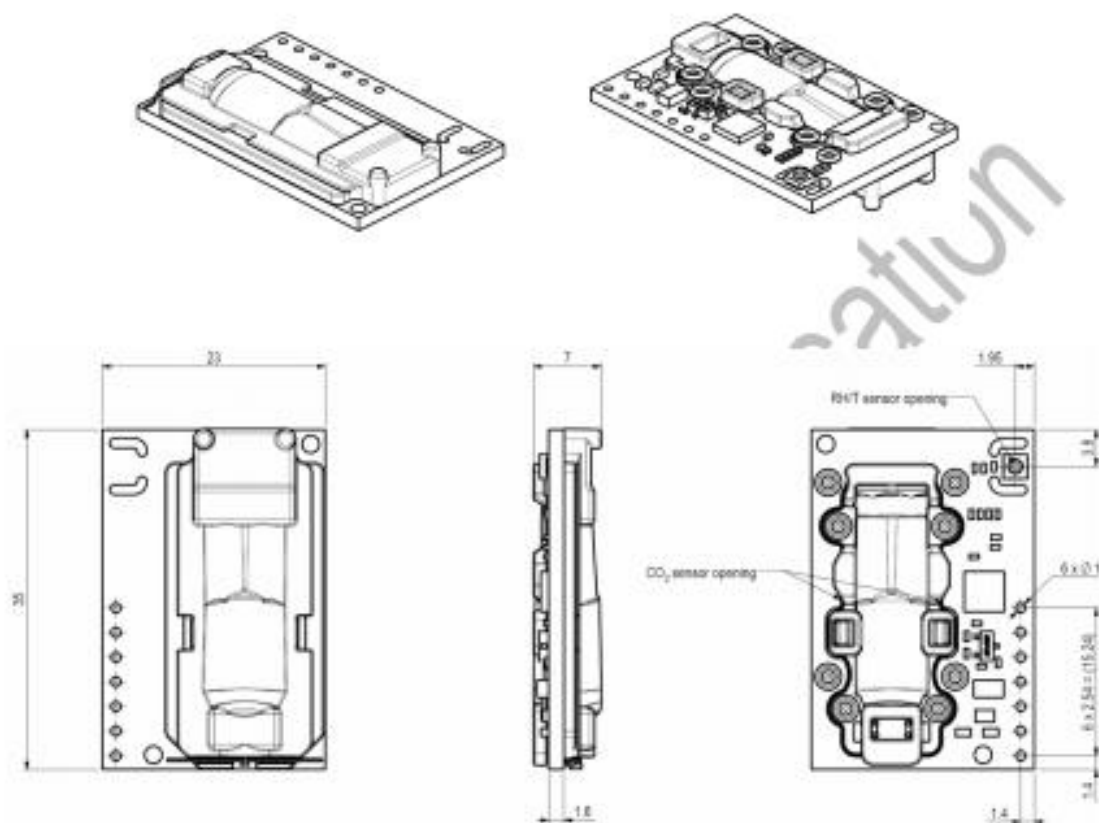


Figure 1 Product outline drawing of SCD30. Pictures on the left show top-view, pictures on the right bottom-view.

Sensor height is 7 mm at the thickest part of SCD30. The weight of one SCD30 sensor is 3.4 g.

3 Pin-Out Diagram

Preliminary information. Subject to change without notice.

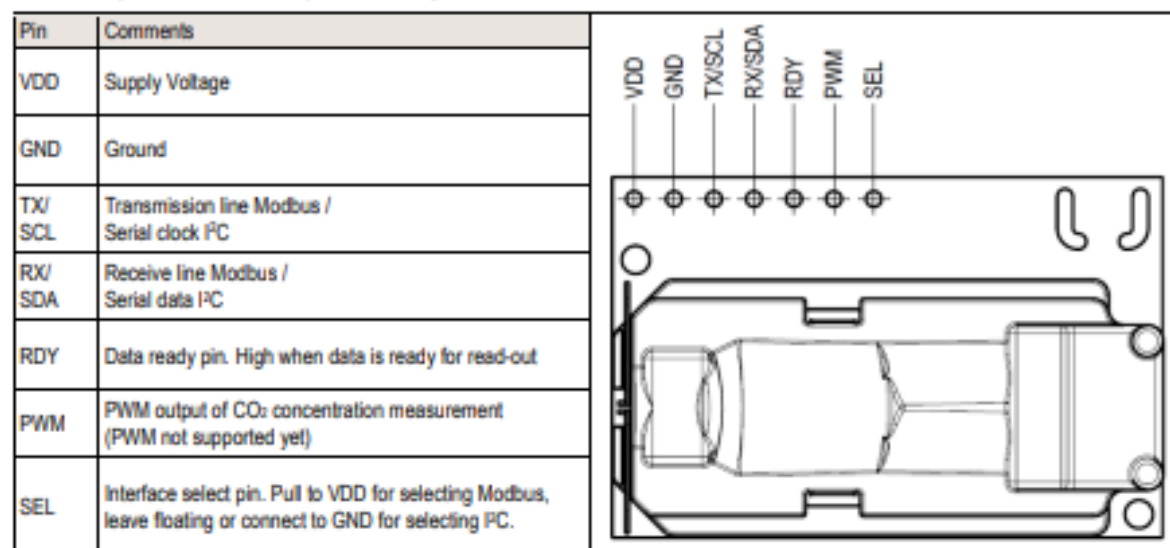


Figure 2: Pin-out of the SCD30.

4 Operation and Communication

Please contact your local Sensirion representative for separate document.

5 Shipping Package

SCD30 sensor is shipped in stackable trays with 40 pieces each. The tray dimension is 363 mm x 257 mm x 19 mm. Stacking of trays results in an effective tray height of 13 mm.

6 Ordering Information

SCD30 and accessory can be ordered via the following article numbers. Please accept longer lead times until official start of production.

Product	Description	Article Number
SCD30 sensor	CO ₂ , RH and T sensor module	1-101625-01
SCD30 evaluation kit	SCD30 sensor, SEK sensor bridge and cables.	3.000.055

Anexo 4: Datasheet modulo GPS SAM-M8Q

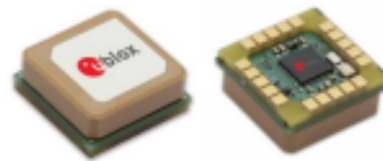
SAM-M8Q

Easy-to-use u-blox M8 GNSS antenna module

Data Sheet

Smart antenna module for easy and reliable integration

- Easy to design-in with no RF expertise required
- Consistently strong performance regardless of installation
- High accuracy thanks to concurrent reception of up to 3 GNSS (GPS, Galileo, GLONASS)
- Tiny form factor 15.5 x 15.5 x 6.3 mm
- Embedded wide-band patch antenna
- Surface-mount device, enabling simple and automated manufacturing



Top and bottom view

www.u-blox.com

UBX-16012619 - R04

1 Functional description

1.1 Overview

The u-blox concurrent SAM-M8Q GNSS patch antenna module benefits from the exceptional performance of the u-blox M8 multi-GNSS engine. The SAM-M8Q module offers high sensitivity and minimal acquisition times in an ultra compact form factor.

The SAM-M8Q module utilizes concurrent reception of up to three GNSS systems (GPS/Galileo and GLONASS), recognizes multiple constellations simultaneously and provides outstanding positioning accuracy in scenarios where urban canyon or weak signals are involved. For even better and faster positioning improvement, the SAM-M8Q supports augmentation of QZSS, GAGAN and IMES together with WAAS, EGNOS, MSAS. The SAM-M8Q also supports message integrity protection, geofencing, and spoofing detection with configurable interface settings to easily fit to customer applications.

Incorporating the SAM-M8Q module into customer designs is simple and straightforward, thanks to the embedded GNSS patch antenna, low power consumption, a small footprint of 15.5 x 15.5 x 6.3 mm, simple interface, and sophisticated interference suppression that ensures maximum performance even in GNSS-hostile environments.

The 15 x 15 mm patch antenna provides the best compromise between the performance of a Right Hand Circular Polarized (RHCP) antenna and a small size to be integrated in any design. The omni-directional radiation pattern increases flexibility for device installation. The simple SMD design and easy interfacing keeps design and manufacturing costs to a minimum. The SAM-M8Q features an additional front-end LNA for optimized performance and a front-end SAW filter for increased jamming immunity. The SAM-M8Q targets industrial and consumer applications that require small, cost efficient, and ready-to-use GNSS solutions.

The SAM-M8Q module uses AEC-Q100 qualified GNSS chips and is fully tested at the system level. Qualification is done according to the ISO 16750 standard.

The u-blox SAM-M8Q module can also benefit from the u-blox AssistNow assistance service. The Online service provides GNSS broadcast parameters, e.g. ephemeris, almanac plus time or rough position to reduce the receiver's time to first fix significantly and improve acquisition sensitivity. The extended validity of AssistNow Offline data (up to 35 days) and AssistNow Autonomous data (up to 3 days) provide faster acquisition after a long off time.



See section 1.6 for more information concerning the SAM-M8Q related AssistNow Assistance.

1.2 Product features

Model	Category	GNSS		Supply	Interfaces	Features	Grade
	Standard Precision GNSS High Precision GNSS Dead Reckoning Tearing	GPS / QZSS GLONASS Galileo BeiDou	Number of Concurrent GNSS	2.7 V – 3.6 V	UART USB SPI DDC (I ² C compliant)	Programmable (Flash) Data logging Additional SAW Additional LNA RTC crystal Oscillator Built-in antenna Built-in antenna supply and supervisor Timepulse	Standard Professional Automotive
SAM-M8Q	•	• • •	3	•	• • •	• • • • • • • • •	•

C = Crystal / T = TCXO



1.3 Performance

Parameter	Specification				
Receiver type	72-channel u-blox M8 engine GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1 SAIF, GLONASS L1OF, Galileo E1B/C				
Accuracy of time pulse signal	RMS	30 ns			
	99%	60 ns			
Frequency of time pulse signal	0.25 Hz...10 MHz (configurable)				
Operational limits ¹	Dynamics	≤ 4 g			
	Altitude	50,000 m			
	Velocity	500 m/s			
Velocity accuracy ²	0.05m/s				
Heading accuracy ²	0.3 degrees				
	GNSS	GPS & GLONASS	GPS	GLONASS	Galileo
Horizontal position accuracy ³		2.5 m	2.5 m	8.0 m	TBC ⁴
Max navigation update rate ⁴		10 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix ⁵	Cold start	26 s	29 s	30 s	TBC ⁴
	Hot start	1 s	1 s	1 s	TBC ⁴
	Aided starts ⁷	2 s	2 s	3 s	TBC ⁴
Sensitivity ⁶	Tracking & Navigation	-165 dBm	-164 dBm	-164 dBm	-157 dBm
	Reacquisition	-158 dBm	-158 dBm	-154 dBm	-151 dBm
	Cold start	-146 dBm	-146 dBm	-143 dBm	-136 dBm
	Hot start	-155 dBm	-155 dBm	-154 dBm	-149 dBm

Table 1: SAM-M8Q (on 50 mm x 50 mm GND plane) performance in different GNSS modes (default: concurrent reception of GPS and GLONASS incl. QZSS, SBAS)

1.4 Block diagram

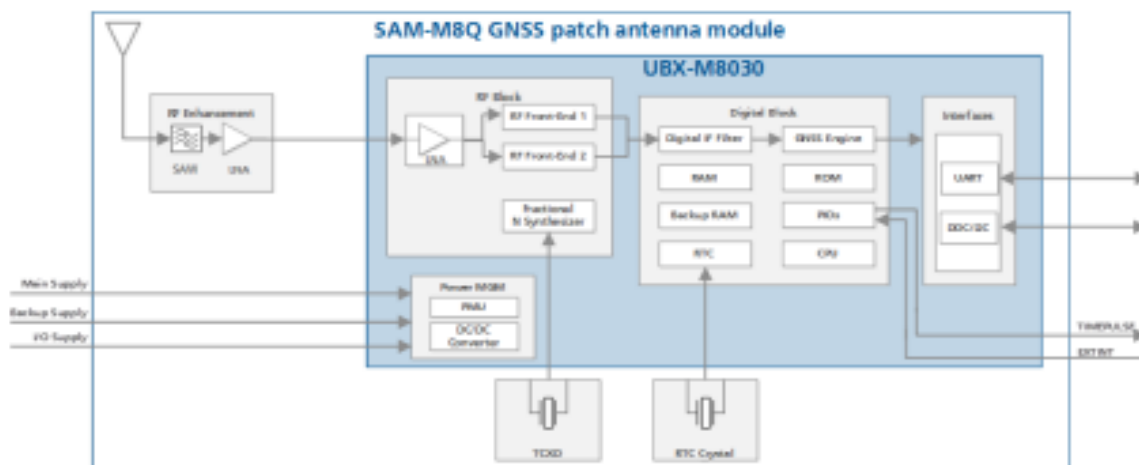


Figure 1: SAM-M8Q block diagram

1.5 Supported GNSS Constellations

The SAM-M8Q GNSS module is a concurrent GNSS receiver which can receive and track multiple GNSS systems: GPS, Galileo and GLONASS. Owing to the dual-frequency RF front-end architecture, GLONASS can be processed concurrently with GPS and Galileo signals, thus providing reception of three GNSS systems. By default, the M8 receivers are configured for concurrent GPS and GLONASS, including SBAS and QZSS reception. If power consumption is a key factor, then the receiver should be configured for a single GNSS operation using GPS, Galileo or GLONASS and disabling QZSS and SBAS. The module can also be configured to receive any single GNSS constellation.

GPS	Galileo	GLONASS
•	•	–
•	•	•
•	–	•
•	–	–
–	•	•
–	•	–
–	–	•

Table 2: Permissible GNSS combinations (• = enabled)

- The augmentation systems: SBAS and QZSS can be enabled only if GPS operation is also configured.
- Galileo is not enabled as the default configuration.
- BeiDou reception is not supported by SAM-M8Q module.

1.5.1 GPS

The SAM-M8Q GNSS patch antenna module is designed to receive and track the L1C/A signals provided at 1575.42 MHz by the Global Positioning System (GPS).

1.5.2 GLONASS

The SAM-M8Q GNSS patch antenna module can receive and process the GLONASS satellite system as an alternative to the US-based Global Positioning System (GPS). The SAM-M8Q is designed to receive and track the L1OF signals which GLONASS provides at $1602 \text{ MHz} + k \cdot 562.5 \text{ kHz}$, where k is the satellite's frequency channel

Anexo 5: Código puerta de enlace

/ Trabajo desarrollado como parte del Proyecto de Titulación:*

"RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS

**PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO
PARA MONITOREO AMBIENTAL EN INVERNADEROS."**

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

AUTOR: CARLOS ALBERTO GORDILLO GORDILLO

El trabajo de titulación con tema antes mencionado fue Aprobado mediante Resolución HCD Nro. UTN-FICA-2020-016 de fecha 12 de marzo de 2020 en la Universidad Técnica del Norte ubicada en la ciudad de Ibarra-Ecuador.

**/*

```
#include <SPI.h> // libreria para comunicar multiples sensores
#include <LoRa.h> // libreria LoRa
const int csPin = 16; // pines activacion modulo LoRa
const int resetPin = 27;
const int irqPin = 16;

byte localAddress = 0xAA; // Direccion MAC puerta de enlace
byte destinationAddress = 0xBB; // Direccion MAC Nodo 1
byte destination2Address = 0xCC; // Direccion MAC Nodo 2
long lastSendTime = 0; // Tiempo de la función millis
int interval = 3000; // Tiempo
int count = 0; // Mensaje de respuesta

void setup() {
  Serial.begin(9600); // Comunicación en baudrate
```

```

pinMode(17,OUTPUT); // Led de confirmacion de llegada
Serial.println("Start LoRa duplex");

LoRa.setPins(esPin, resetPin, irqPin); //Inicializa pines modulo LoRa

if (!LoRa.begin(915E6)) { //Establece frecuencia de trabajo LoRa
  Serial.println("LoRa init failed. Check your connections.");
  while (true) {}
}
}

void loop() {
  if (millis() - lastSendTime > interval) {
    String sensor = String(count++); //Mensajes de respuesta
    sendMessage(sensor); // Respuesta Nodo 1
    sendMessage2(sensor); // Respuesta Nodo 2

    lastSendTime = millis();
    interval = random(3000) + 1000;
  }

  receiveMessage(LoRa.parsePacket()); //Resuesta llegada de mensajes
}

void sendMessage(String outgoing) { // Respuesta nodo 1
  LoRa.beginPacket(); //Inicia Cx LoRa
  LoRa.write(destinationAddress); //Describe el destino
  LoRa.write(localAddress); //Describe el origen
  LoRa.write(outgoing.length()); //Describe el mensaje
  LoRa.print(outgoing); //Envia mensaje
  LoRa.endPacket(); //fin de Cx LoRa
}

```

```

void sendMessage2(String outgoing) { //Respuesta nodo 2
    LoRa.beginPacket();          //Inicia Cx LoRa
    LoRa.write(destination2Address); //Describe el destino
    LoRa.write(localAddress);     //Describe el origen
    LoRa.write(outgoing.length()); //Describe el mensaje
    LoRa.print(outgoing);        //Envía mensaje
    LoRa.endPacket();           //fin de Cx LoRa
}

void receiveMessage(int packetSize) { //Recepcion Mensajes
    if (packetSize == 0) return; //Respuesta de confirmacion
    int recipient = LoRa.read(); //Lee los datos de LoRa
    byte sender = LoRa.read(); //Lee el origen
    byte incomingLength = LoRa.read(); //Lee el tamaño del mensaje
    String incoming = "";
    while (LoRa.available()) {
        incoming += (char)LoRa.read();
    }
    if (incomingLength != incoming.length()) {
        return;
    }
    if (recipient != localAddress) {
        return;
    }
    digitalWrite(17,HIGH); //Enciende Led cuando llega un mensaje
    delay(200);
    digitalWrite(17,LOW);
    delay(200);
    Serial.println(incoming); //Imprime el mensaje Recibido
}

```

Anexo 6: Código Nodos 1 y 2

/* Trabajo desarrollado como parte del Proyecto de Titulación:

"RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS

**PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO
PARA MONITOREO AMBIENTAL EN INVERNADEROS.**

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

AUTOR: CARLOS ALBERTO GORDILLO GORDILLO

El trabajo de titulación con tema antes mencionado fue Aprobado mediante Resolución HCD Nro. UTN-FICA-2020-016 de fecha 12 de marzo de 2020 en la Universidad Técnica del Norte ubicada en la ciudad de Ibarra-Ecuador.

*/

```
#include <SparkFun_VEML6075_Arduino_Library.h> //Libreria Sensor UV
VEML6075 uv; // inicializa el sensor

#include <Wire.h> //Libreria comunicacion I2C

#include "SparkFun_SCD30_Arduino_Library.h" //Libreria sensor
CO2, Temperatura, Humedad

SCD30 airSensor; //Inicializa el sensor

#include "SparkFun_Ublox_Arduino_Library.h" //Libreria Modulo GPS
SFE_UBLOX_GPS myGPS; //Inicializa sensor GPS

#include <SPI.h> //Libreria para comunicar multiples sensores

#include <LoRa.h> //Libreria Lora

const int csPin = 16; //Pines activacion LoRa

const int resetPin = 27;

const int irqPin = 16;

byte localAddress = 0xBB; //Direccion Local

byte destinationAddress = 0xAA; //Direccion puerta de enlace

long lastSendTime = 0;
```



```

int interval = 2000;
int count = 0;
String coma=",";
String inputString = "";

void setup() {

  Serial.begin(9600); //Comunicacion Serial
  pinMode(17,OUTPUT); //Led de alerta
  Wire.begin();
  Serial.println("Start LoRa duplex");

  LoRa.setPins(csPin, resetPin, irqPin); //inicializa pines LoRa
  if (!LoRa.begin(915E6)) { //Establece frecuencia de trabajo LoRa
    Serial.println("LoRa init failed. Check your connections.");
    while (true) {}
  }
  if (myGPS.begin() == false){ //Inicia el modulo GPS
    Serial.println(F("Ublox GPS not detected at default I2C address. Please check wiring.
Freezing."));
    while (1);
  }
  if (airSensor.begin() == false){ //Inicializa sensor CO2, Temperatura, humedad
    Serial.println("Air sensor not detected. Please check wiring. Freezing...");
    while (1);
  }
  if (uv.begin() == false) { //Inicializa sensor UV
    Serial.println("Unable to communicate with VEML6075.");
    while (1) ;
  }
}

```

```

myGPS.setI2COutput(COM_TYPE_UBX);
myGPS.saveConfiguration();

}

void loop() {
  long latitude = myGPS.getLatitude(); //Obtiene latitud
  long longitude = myGPS.getLongitude(); //Obtiene Longitud
  long altitude = myGPS.getAltitude(); //Obtiene Altitud
  if (Serial.available() > 0) { //Condicion si recibe un mensaje
    char dato = Serial.read(); //Lee el mensale
    if(dato=='E'){ //Si recibe la E envia los datos
      String temperatura = String (airSensor.getTemperature(), 1);//Obtiene Temperatura
      String humedad = String (airSensor.getHumidity(),1); //Obtiene Humedad
      String sensorData =
(uv.index()+coma+airSensor.getCO2()+coma+temperatura+coma+humedad+coma+latitud
e+coma+longitude+coma+altitude+coma+"B");
      sendMessage(sensorData); // Envia por completo los datos
      lastSendTime = millis();
      interval = random(2000) + 1000;
      receiveMessage(LoRa.parsePacket());
    }
  }
}

void sendMessage(String outgoing) { //Envio de datos
  LoRa.beginPacket(); //Inicia Cx LoRa
  LoRa.write(destinationAddress); //Describe el destino
  LoRa.write(localAddress); //Describe el origen
  LoRa.write(outgoing.length()); //Describe el mensaje
  LoRa.print(outgoing); //Envia el mensaje
}

```

```

    LoRa.endPacket();      //Fin Cx loRa
}

void receiveMessage(int packetSize) { //Recepcion de mensajes
    if (packetSize == 0) return;     //Respuesta de confirmacion

    int recipient = LoRa.read();     //Lee los datos de LoRa
    byte sender = LoRa.read();      //Lee el origen
    byte incomingLength = LoRa.read(); //Lee el tamaño del mensaje

    String incoming = "";

    while (LoRa.available()) {
        incoming += (char)LoRa.read();
    }

    if (incomingLength != incoming.length()) {
        return;
    }

    if (recipient != localAddress) {
        return;
    }

    digitalWrite(17,HIGH);      //Enciende led cuando envia el mensaje
    delay(200);
    digitalWrite(17,LOW);
    delay(200);
    Serial.println("R");      //Cuando recibe confirmacion envia R
}

```

Anexo 7: Código MSO

#Trabajo desarrollado como parte del Proyecto de Titulación:

"RED DE SENSORES INALÁMBRICOS BAJO PROTOCOLO LORA Y GESTIÓN DE PROCESOS

PARA LA ANALÍTICA DE DATOS MEDIANTE META SISTEMA OPERATIVO

PARA MONITOREO AMBIENTAL EN INVERNADEROS."

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE

INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

#AUTOR: CARLOS ALBERTO GORDILLO GORDILLO

#El trabajo de titulación con tema antes mencionado fue Aprobado mediante

#Resolución HCD Nro. UTN-FICA-2020-016 de fecha 12 de marzo de 2020

#en la Universidad Técnica del Norte ubicada en la ciudad de Ibarra-Ecuador

```
from tkinter import *
```

```
from tkinter import ttk
```

```
from PIL import ImageTk,Image
```

```
import tkinter as tk
```

```
import pylab as pl
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
root=tk.Tk()
```

```
root.title("Interfaz de control")
```

```
root.resizable(True,True)
```

```
root.iconbitmap('C:/Users/carlo/Desktop/Interfaz/logo.ico')
```

```
imagenLogo=PhotoImage(file="C:/Users/carlo/Desktop/Interfaz/NUTN.jpg")
```

```
lblImagen=Label(root,image=imagenLogo).place(x=20,y=20)
```

```
root.geometry("900x700")
```

```
root['background']='#CDCDCD'
```

```
###Scrollbar
```

```
#main_frame = Frame(root)
```

```
#main_frame.pack(fill=BOTH,expand=1)
```

```

#my_canvas= Canvas(main_frame)
#my_canvas.pack(side=LEFT,fill=BOTH,expand=1)
#my_scroll=ttk.Scrollbar(main_frame,orient=VERTICAL,command=my_canvas.yview)
#my_scroll.pack(side=RIGHT,fill=Y)
#my_canvas.configure(yscrollcommand=my_scroll.set)
#my_canvas.bind('<Configure>',lambda
e:my_canvas.configure(scrollregion=my_canvas.bbox('all')))
##Second frame
#second_frame=Frame(my_canvas)
#my_canvas.create_window((0,0),window=second_frame,anchor="nw")
##Titulo
labelt=Label(root,text="UTN-CIERCOM",font=("Times New Roman",16)).pack()
labelt1=Label(root,text="Monitoreo de cultivos",font=("Agency FB",14)).pack()
labelt2=Label(root,text="Interfaz de control",font=("Agency FB",14)).pack()
##Comunicacion
def encender():
    labelEncender=Label(root,text="Sistema encendido",background =
'#4FF719',font=("Times New Roman",12)).place(x=200,y=155)
def prender():
    labelPrender=Label(root,text="Si envía datos",background = '#4FF719',font=("Times
New Roman",12)).place(x=200,y=205)
labelt4=Label(root,text="Conexión",font=("Agency FB",14)).place(x=170,y=100)
btnCx=Button(root,text="Envío",command=encender,font=("Times New
Roman",12)).place(x=90,y=150)
btnEstado=Button(root,text="Estado WSN",command=prender,font=("Times New
Roman",12),width=10).place(x=90,y=200)
##Control
labelt5=Label(root,text="Control",font=("Agency FB",14)).place(x=600,y=90)
#menu
n = tk.StringVar()
menu1=ttk.Combobox(root, width = 20,textvariable = n)
menu1['values'] = ('UV',

```

```

        'CO2',
        'Temperatura',
        'Humedad')
menu1.place(x=630,y=140)
n2 = tk.StringVar()
menu2=ttk.Combobox(root, width = 20,textvariable = n2)
menu2['values'] = ('UV',
                  'CO2',
                  'Temperatura',
                  'Humedad')
menu2.place(x=630,y=240)
#Graficos Nodo 1
def Nodo1():
    if n.get() == 'UV':
        plt.plot([1, 2, 3, 4, 5], [0.22,0.13,0.16,0.15,0.18], 'ro',linestyle="-")
        plt.axis([0, 6, 0.10, 0.45])
        plt.xlabel('Cultivo')
        plt.ylabel('Indice UV')
        plt.title('Rayos ultra violeta')
        plt.show()
    else:
        print("")
    if n.get() == 'CO2':
        plt.plot([1, 2, 3, 4, 5], [656,882,691,485,403], 'bo',linestyle="-")
        plt.axis([0, 6, 300, 1350])
        plt.xlabel('Cultivo')
        plt.ylabel('CO2')
        plt.title('CO2 (PPM)')
        plt.show()
    else:

```

```

    print("")
if n.get() == 'Temperatura':
    plt.plot([1, 2, 3, 4, 5], [27.2,27.3,27.2,27.3,27.2], 'go',linestyle="-")
    plt.axis([0, 6, 20, 35])
    plt.xlabel('Cultivo')
    plt.ylabel('Temperatura')
    plt.title('Temperatura °C')
    plt.show()
else:
    print("")
if n.get() == 'Humedad':
    plt.plot([1, 2, 3, 4, 5], [61.5,60,62.4,60.7,57.6], 'yo',linestyle="-")
    plt.axis([0, 6, 40, 63])
    plt.xlabel('Cultivo')
    plt.ylabel('Humedad')
    plt.title('Humedad %')
    plt.show()
else:
    print("")
btnN1=Button(root,text="Graficar  Nodo 1",command=Nodo1,font=("Times New
Roman",12),width=15).place(x=450,y=130)
#Graficos Nodo 2
def Nodo2():
    if n2.get() == 'UV':
        plt.plot([1, 2, 3, 4, 5], [0.18,0.20,0.23,0.20,0.34], 'ro',linestyle="-")
        plt.axis([0, 6, 0.10, 0.45])
        plt.xlabel('Cultivo')
        plt.ylabel('Indice UV')
        plt.title('Rayos ultra violeta')
        plt.show()
    else:

```

```

    print("")
if n2.get() == 'CO2':
    plt.plot([1, 2, 3, 4, 5], [540,680,530,385,382], 'bo',linestyle="-")
    plt.axis([0, 6, 300, 1350])
    plt.xlabel('Cultivo')
    plt.ylabel('CO2')
    plt.title('CO2 (PPM)')
    plt.show()
else:
    print("")
if n2.get() == 'Temperatura':
    plt.plot([1, 2, 3, 4, 5], [28.7,28.7,28.9,29.1,29.4], 'go',linestyle="-")
    plt.axis([0, 6, 20, 35])
    plt.xlabel('Cultivo')
    plt.ylabel('Temperatura')
    plt.title('Temperatura °C')
    plt.show()
else:
    print("")
if n2.get() == 'Humedad':
    plt.plot([1, 2, 3, 4, 5], [54.7,51.4,51.3,51.3,52.5], 'yo',linestyle="-")
    plt.axis([0, 6, 40, 63])
    plt.xlabel('Cultivo')
    plt.ylabel('Humedad')
    plt.title('Humedad %')
    plt.show()
else:
    print("")
btnN2=Button(root,text="Graficar  Nodo 2",command=Nodo2,font=("Times New
Roman",12),width=15).place(x=450,y=230)
##Análisis

```



```
label5=Label(root,text="Análisis",font=("Agency FB",14)).place(x=400,y=300)
#imagen
imagenAn=PhotoImage(file="C:/Users/carlo/Desktop/Interfaz/clusterp.png")
lblImagen=Label(root,image=imagenAn).place(x=235,y=370)
root.mainloop()
#pl.show()
```

Anexo 8: Evidencias fotográficas

- Pruebas finales toma de datos de sensores





- Toma de datos en cultivo temprano



- Toma de datos del cultivo



- Toma de datos en otra sección del invernadero





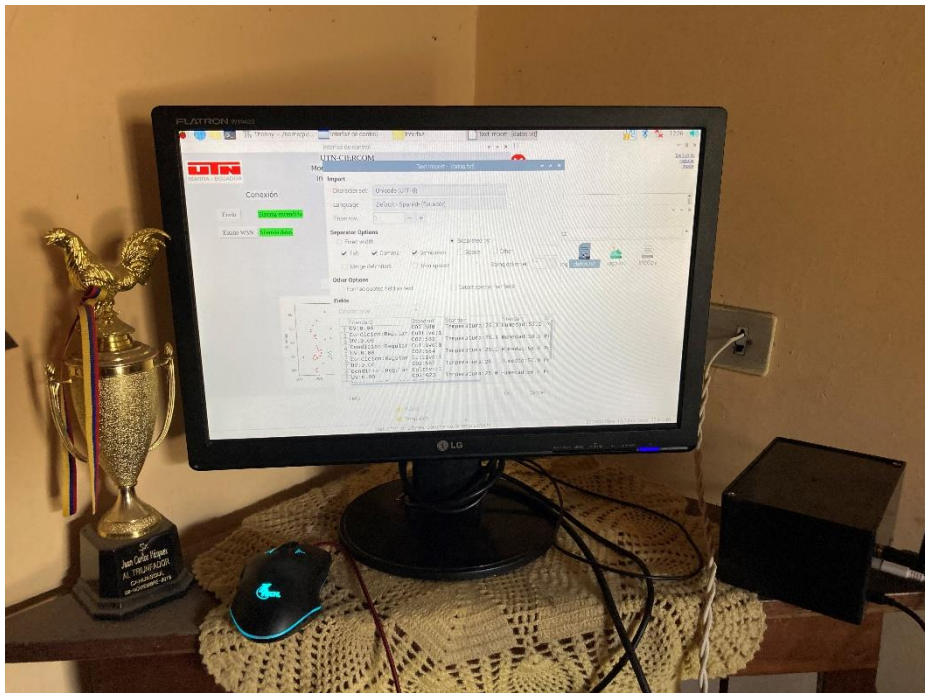
- Pruebas toma de datos de sensores



- Pruebas del sistema completo



- Interfaz gráfica en otra zona del invernadero



Anexo 9: Acciones monitor serial

- Emisión y recepción de mensajes vía LoRa

The image shows two serial terminal windows side-by-side. The left window, titled 'COM5', displays a list of 20 coordinate pairs (Latitude, Longitude, Altitude) for various sensor locations. The right window, titled 'COM9', shows the received data packets, which correspond to the coordinates in the left window. Each received packet includes the coordinates and an RSSI value.

```

COM5
-----
3736933,-781240742,2227640
3593008,-781168439,950224
3684240,-781177626,1108791
3672867,-781184663,1232013
3664007,-781241036,2218976
3593514,-781241667,2230271
3592754,-781241874,2234165
3592498,-781242056,2237299
3592318,-781242106,2238641
3592243,-781242104,2238783
3592251,-781242164,2239872
3592220,-781242249,2241504
3592130,-781242215,2241006
3592177,-781242160,2240356
3592208,-781242160,2240372
3592215,-781242114,2239741
3592251,-781241927,2237117
3592414,-781241927,2235905
3592479,-781241854,2235673

COM9
-----
Received packet ==> A
With RSSI -110
Received packet ==> UV:1.68
With RSSI -105
Received packet ==> Co2:381
With RSSI -107
Received packet ==> T°:26.0
With RSSI -106
Received packet ==> H:53.2
With RSSI -105
Received packet ==> La:3592215
With RSSI -106
Received packet ==> Lo:-781241927
With RSSI -105
Received packet ==> Al:2235905
With RSSI -106
Received packet ==> Z
With RSSI -113
  
```

- Envío de datos de sensores

The image shows two serial terminal windows side-by-side. The left window, titled 'COM5', displays sensor data (CO2, temperature, humidity) and coordinates for multiple sensor locations. The right window, titled 'COM9', shows the received data packets, which include the sensor readings and coordinates. The data is formatted as a structured text string for each packet.

```

COM5
-----
co2(ppm):383 temp(C):23.2 humidity(%):60.5
Lat: 3593186 Long: -781241233 (degrees * 10^-7) Alt: 2230273 (mm) SIV: 9
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):383 temp(C):23.1 humidity(%):60.5
Lat: 3593222 Long: -781241154 (degrees * 10^-7) Alt: 2229156 (mm) SIV: 11
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):382 temp(C):23.2 humidity(%):60.6
Lat: 3593244 Long: -781241095 (degrees * 10^-7) Alt: 2228207 (mm) SIV: 10
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):381 temp(C):23.2 humidity(%):60.5
Lat: 3593263 Long: -781241040 (degrees * 10^-7) Alt: 2228072 (mm) SIV: 10
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):381 temp(C):23.2 humidity(%):60.5
Lat: 3593252 Long: -781241013 (degrees * 10^-7) Alt: 2228167 (mm) SIV: 10
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):379 temp(C):23.2 humidity(%):60.5
Lat: 3593250 Long: -781241016 (degrees * 10^-7) Alt: 2227427 (mm) SIV: 11
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):379 temp(C):23.2 humidity(%):60.5
Lat: 3593270 Long: -781240966 (degrees * 10^-7) Alt: 2227727 (mm) SIV: 11
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):380 temp(C):23.2 humidity(%):60.5
Lat: 3593244 Long: -781240975 (degrees * 10^-7) Alt: 2227977 (mm) SIV: 10
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):380 temp(C):23.1 humidity(%):60.4
Lat: 3593210 Long: -781240983 (degrees * 10^-7) Alt: 2228674 (mm) SIV: 11
Sending packet: UVA-167111.70, UVB-242475.81, UV Index-244.38
co2(ppm):378 temp(C):23.1 humidity(%):60.5
Lat: 3593189 Long: -781241003 (degrees * 10^-7) Alt: 2229276 (mm) SIV: 10

COM9
-----
co2(ppm):381 temp(C):23.2 humidity(%):60.5
Lat: 3593252 Long: -781241013 Alt: 2228167 (mm) SIV: 10
' with RSSI -109
Received packet 'Valores sensores
UVA:-167111.70 UVB:-242475.81 UV Index:-244.38
co2(ppm):379 temp(C):23.2 humidity(%):60.5
Lat: 3593250 Long: -781241016 Alt: 2227427 (mm) SIV: 11
' with RSSI -108
Received packet 'Valores sensores
UVA:-167111.70 UVB:-242475.81 UV Index:-244.38
co2(ppm):380 temp(C):23.2 humidity(%):60.5
Lat: 3593270 Long: -781240966 Alt: 2227727 (mm) SIV: 11
' with RSSI -108
Received packet 'Valores sensores
UVA:-167111.70 UVB:-242475.81 UV Index:-244.38
co2(ppm):380 temp(C):23.1 humidity(%):60.4
Lat: 3593244 Long: -781240975 Alt: 2227977 (mm) SIV: 10
' with RSSI -108
Received packet 'Valores sensores
UVA:-167111.70 UVB:-242475.81 UV Index:-244.38
co2(ppm):378 temp(C):23.1 humidity(%):60.5
Lat: 3593210 Long: -781240983 Alt: 2228674 (mm) SIV: 11
' with RSSI -108
Received packet 'Valores sensores
UVA:-167111.70 UVB:-242475.81 UV Index:-244.38
co2(ppm):379 temp(C):23.2 humidity(%):60.5
Lat: 3593189 Long: -781241003 Alt: 2229276 (mm) SIV: 10
' with RSSI -110
  
```

- Comunicación con respuesta de recepción

```

COM3

0.01,357,23.5,63.2,3344175,-781258054,2255693 from 0xbb to 0xaa
0.01,356,23.5,63.0,3344129,-781258032,2255125 from 0xbb to 0xaa
Received data 2393 from 0xaa to 0xbb
0.01,354,23.5,63.0,3344131,-781258023,2255200 from 0xbb to 0xaa
0.01,354,23.6,62.8,3344086,-781257971,2254572 from 0xbb to 0xaa
0.01,354,23.6,62.8,3344076,-781257958,2254472 from 0xbb to 0xaa
0.01,355,23.6,62.7,3344068,-781257946,2254143 from 0xbb to 0xaa
0.01,354,23.6,62.6,3344034,-781257933,2253588 from 0xbb to 0xaa
0.01,353,23.6,62.6,3344029,-781257930,2253412 from 0xbb to 0xaa
0.01,353,23.6,62.5,3343996,-781257928,2253007 from 0xbb to 0xaa
0.01,353,23.7,62.5,3343899,-781257904,2253096 from 0xbb to 0xaa
0.01,352,23.7,62.4,3343824,-781257894,2252690 from 0xbb to 0xaa
0.01,353,23.8,62.3,3343774,-781257886,2252007 from 0xbb to 0xaa
0.01,352,23.8,62.3,3343786,-781257904,2251786 from 0xbb to 0xaa
0.01,352,23.8,62.1,3343833,-781257936,2252238 from 0xbb to 0xaa

 Autoscroll  Mostrar marca temporal

```

- Respuesta desde la Puerta de enlace

```

COM4

Sending data 2428 from 0xaa to 0xbb to 0xcc
Sending data 2429 from 0xaa to 0xbb to 0xcc
Dato recibido :0.00,352,24.3,61.1,3344040,-781258116,2254957 desde 0xbb hacia 0xaa
Sending data 2430 from 0xaa to 0xbb to 0xcc
Dato recibido :0.00,352,24.3,61.1,3343996,-781258090,2255715 desde 0xbb hacia 0xaa
Sending data 2431 from 0xaa to 0xbb to 0xcc
Sending data 2432 from 0xaa to 0xbb to 0xcc
Sending data 2433 from 0xaa to 0xbb to 0xcc

```