

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Implementación de una Red Mesh con ESP32 para el monitoreo y control en
piscinas camaroneras

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Telecomunicaciones

Presentado por:

Xavier Alejandro Alvarado Barros

Giordan Daniel Maldonado González

Guayaquil - Ecuador

Año: 2023

Dedicatoria

El presente proyecto de grado se lo dedico a Dios, mis padres y hermanos por siempre brindarme de manera incondicional su amor, apoyo y comprensión a lo largo de toda mi travesía universitaria.

A don Fernando y la Sra. Yessenia, mis padrinos putativos, por siempre brindarme su gentil ayuda y aliento.

Al Ing. Miguel Yapur, mi “padrino”, por toda su sabiduría, paciencia y apoyo a lo largo de esta corrida académica.

Al Ing. Aristóteles Amat, por ser como un hermano mayor, brindándome su apoyo, aliento, guía y enseñanzas a lo largo de la carrera.

Xavier Alvarado, Jr.

Dedicatoria

Quiero dedicar este proyecto de grado a mi familia, por su inquebrantable fe en mí y por ser mi refugio en momentos de incertidumbre. Su amor y aliento me han sostenido a lo largo de los años y me han llevado hasta este punto.

A la ESPOL, por brindarme la plataforma y los recursos necesarios para perseguir mis sueños académicos y alcanzar esta meta.

A todos aquellos que han contribuido, en grande o pequeña medida, a mi formación y crecimiento, les dedico este logro con profundo agradecimiento.

Giordan Maldonado

Agradecimientos

Mi más sincero agradecimiento al Ing. Félix Moncayo, Ing. Tonny Toscano y la Ing. Antonieta Álvarez por su guía y retroalimentación durante el desarrollo del presente proyecto de titulación.

A todos mis maestros por compartirme sus conocimientos, por ser una guía académica y ética a través de estos 5 años para poder ser un profesional y brindar excelente aporte a la sociedad.

Quisiera agradecerles también a todos mis amigos por enriquecerme con todas las experiencias y recuerdos que me hemos vivido, en especial a Cristopher.

Xavier Alvarado, Jr.

Agradecimientos

En primer lugar, quiero expresar mi profundo agradecimiento a Ing. Félix Moncayo y la Ing. Antonieta Álvarez por su guía, paciencia y apoyo incondicional a lo largo de este proceso de investigación. Sus conocimientos y orientación fueron fundamentales para el desarrollo y culminación de este trabajo.

Mis padres, merecen un reconocimiento especial por su amor incondicional y su apoyo constante a lo largo de mi educación. Su sacrificio y aliento han sido el motor que me impulsó a alcanzar este hito.

Finalmente, quiero expresar mi reconocimiento a mi alma mater, ESPOL, por brindarme los recursos y el entorno propicio para llevar a cabo este proyecto académico.

Giordan Maldonado

Declaración Expresa

Nosotros, Xavier Alejandro Alvarado Barros y Giordan Daniel Maldonado González acordamos y reconocemos que la titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, información no divulgada y cualquier otro derecho o tipo de Propiedad Intelectual que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada durante el desarrollo de su trabajo de titulación, incluyendo cualquier derecho de participación de beneficios o de valor sobre titularidad de derechos, pertenecerán de forma total, perpetua, exclusiva e indivisible a LA ESPOL, sin limitación de ningún tipo. Se deja además expresa constancia de que lo aquí establecido constituye un “previo acuerdo”, así como de ser posible bajo la normativa vigente de transferencia o cesión a favor de la ESPOL de todo derecho o porcentaje de titularidad que pueda existir.

Sin perjuicio de lo anterior los alumnos firmantes de la presente declaración reciben en este acto una licencia de uso gratuita e intransferible de plazo indefinido para el uso no comercial de cualquier investigación, desarrollo tecnológico o invención realizada durante el desarrollo de su trabajo de titulación, sin perjuicio de lo cual deberán contar con una autorización previa expresa de la ESPOL para difundir públicamente el contenido de la investigación, desarrollo tecnológico o invención.

Así también autorizamos expresamente a que la ESPOL realice la comunicación pública de la obra o invento, por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual.

Guayaquil, 1 de septiembre del 2023.


Xavier Alvarado


Giordan Maldonado

Evaluadores

María Antonieta Álvarez, PhD

Profesor de Materia

Félix Moncayo R. M.Sc.

Tutor de proyecto

Resumen

La industria acuícola ecuatoriana se enfrenta al reto de mantener la calidad de sus productos ante el incremento de las exportaciones del camarón, para ello, es necesario implementar soluciones tecnológicas que permitan el monitoreo y control en tiempo real de los cultivos. En este contexto, el presente proyecto busca diseñar una red Mesh de máxima cobertura utilizando sistemas programables de bajo costo que permitan la conexión e intercambio de datos entre dispositivos IoT. La red se implementó con 6 microcontroladores ESP32 y 5 módulos RF24 en tres escenarios diferentes: un polideportivo, un parque público y un lago, estableciendo las redes en sus distancias máximas de conexión entre nodos y se analizaron los niveles de latencia en dichas distancias. La red implementada con `painlessMesh` logró una distancia máxima de 441.86m con latencias promedio máximas de 906 ms; por otra parte, RF24 Mesh una distancia de 843.19 m con latencia promedio hasta 258.02 ms. RF24 Mesh es óptima para largas distancias con un mejor control en su transmisión y recepción, pero `painlessMesh` adapta de mejor manera los nodos a la red. El aumento de dispositivos incrementa gradualmente los niveles de latencia, ya que, cada dispositivo envía constantemente datos a la red.

Palabras Clave: `painlessMesh`, RF24, latencia, distancia

Abstract

The Ecuadorian aquaculture industry faces the challenge of maintaining the quality of its products given the increase in shrimp exports. For this, it is necessary to implement technological solutions that allow real-time monitoring and control of crops. In this context, this project seeks to design a Mesh network with maximum coverage using low-cost programmable systems that allow the connection and data exchange of IoT devices. The network was implemented with 6 ESP32 devices and 5 RF24 modules in three different scenarios: a sports center, a public park and a lake, establishing the networks at their maximum connection distances between nodes and latency levels at these distances were analyzed. The network implemented with `painlessMesh` achieved a maximum distance of 441.86m with maximum average latencies of 906ms; on the other hand, RF24 Mesh is 843.19 m with average latency levels up to 258.02 ms. RF24 Mesh is optimal for long distances with better control in its transmission and reception, but `painlessMesh` better adapts the nodes to the network. The increase in devices gradually increases latency levels, since each device constantly sends data to the network.

Keywords:

Índice general

Resumen	I
Abstract	II
Índice general	III
Abreviaturas	V
Simbología	VII
Índice de figuras	VIII
Índice de tablas	X
Capítulo 1	1
1.1 Introducción	2
1.2 Descripción del problema	3
1.3 Justificación del problema	5
1.4 Objetivos	5
1.4.1 <i>Objetivo general</i>	5
1.4.2 <i>Objetivos específicos</i>	6
1.5 Marco teórico	6
1.5.1 <i>¿Qué es una Red Mesh?</i>	6
1.5.2 <i>¿Qué es el ESP32 Device Kit?</i>	8
1.5.4 <i>painlessMesh</i>	10
1.5.5 <i>nRF24L01+PA+LNA</i>	11
1.5.6 <i>RF24</i>	12
1.5.7 <i>RF24 Network</i>	12
1.5.8 <i>RF24 Mesh</i>	14
1.6 Antecedentes	14
Capítulo 2	47
2.1 Estructura de investigación	48
2.1.1 <i>Librería painlessMesh</i>	50

2.1.2	<i>Librería RF24 Mesh</i>	54
2.2	Descripción de escenarios	57
2.2.1	<i>Escenario 1: Canchas de FIEC - ESPOL</i>	57
2.2.2	<i>Escenario 2: Parque Samanes</i>	59
2.2.3	<i>Escenario 3: Lago de ESPOL</i>	61
2.3	Diseño de la red Mesh	63
2.3.1	<i>Topología</i>	63
2.3.2	<i>Equipos de red</i>	67
2.3.3	<i>Software</i>	68
2.3.4	<i>Librerías</i>	68
2.4	Implementación	69
2.4.1	<i>Librería painlessMesh</i>	69
2.4.2	<i>Librería RF24 Mesh</i>	71
Capítulo 3	73
3.1	Resultados y análisis.....	74
3.2.1	<i>Pruebas de enlace</i>	74
3.2.2	<i>Pruebas en Complejo deportivo de ESPOL</i>	77
3.2.3	<i>Pruebas en Parque Samanes</i>	80
3.2.4	<i>Pruebas en Lago ESPOL</i>	84
3.2.5	Recopilación de datos.....	90
3.2	Costos de implementación.....	92
Capítulo 4	71
4.1	Conclusiones y recomendaciones	72
4.1.1	<i>Conclusiones</i>	72
4.1.2	<i>Recomendaciones</i>	73
Referencias	74

Abreviaturas

ESPOL	Escuela Superior Politécnica del Litoral
FIEC	Facultad de Ingeniería en Electricidad y Computación
FADCOM	Facultad de Arte, Diseño y Comunicación Audiovisual
UID	Identificador único
M2M	Máquina a máquina (<i>Machine to Machine</i>)
WMN	Redes Mesh inalámbricas
VoIP	Voz sobre IP
SIP	Protocolo de inicio de sesión
CNA	Cámara Nacional de Acuacultura
FCA	Factor de Conversión Alimenticia
CPU	Unidad Central de Procesamiento
ID	Identificador
MID	Identificador de Mesh
NID	Identificador de Red
IoT	Internet de las Cosas
UART	Receptor/ Transmisor Asíncrono Universal
SPI	Interfaz Periférica en Serie
I2C	Circuitos Inter Integrados
ADC	Convertor Analógico-Digital
DAC	Convertor Digital-Analógico
GPIO	Entrada/Salida de Propósito General
E/S	Entrada/Salida
SDIO	Entrada/Salida Digital Segura
SDK	Kit de Desarrollo de Software
PCB	Placa de Circuito Impreso
RF	Radio Frecuencias
PA	Amplificador de Potencia
LNA	Amplificador de Bajo Ruido
OSI	Interconexión de Sistemas Abiertos
WLAN	Red Inalámbrica de Área Local
MQTT	Message Queuing Telemetry Transport

HFC Híbrido de Fibra Coaxial
CONECEL Consorcio Ecuatoriano de Telecomunicaciones
USB Bus Universal en Serie

Simbología

m ²	metros cuadrados
Hz	Hercios
GHz	Giga hercios
MHz	Megahercios
V	Voltios
A	Amperios
μA	microamperios
°C	Grados centígrados
kbps	kilobits por segundo
Mbps	Megabits por segundo
dBm	decibeles por milivatio
mAh	miliamperios hora

Índice de figuras

FIGURA 1. ARQUITECTURA DE UNA RED WI-FI MESH	7
FIGURA 2. TOPOLOGÍA DE MALLA COMPLETA Y PARCIAL	8
FIGURA 3. PINES DEL ESP32	9
FIGURA 4. PINES DE SALIDA DEL MÓDULO NRF24L01+PA+LNA	11
FIGURA 5. MODELO NRF24L01+PA+LNA.....	12
FIGURA 6. ESTRUCTURA DE NODOS EN RF24 NETWORK	13
FIGURA 7. FOTOGRAFÍA SATELITAL DE LA PISCINA CAMARONERA	48
FIGURA 8. DIAGRAMA DE BLOQUES DE LA RED	50
FIGURA 9. DIAGRAMA DE BLOQUES DEL NODO EMISOR	51
FIGURA 10. DIAGRAMA DE BLOQUES DEL NODO REPETIDOR.....	52
FIGURA 11. DIAGRAMA DE BLOQUES DEL NODO CABECERA/MAESTRO	53
FIGURA 12. DIAGRAMA DE BLOQUES DEL NODO ESCLAVO	54
FIGURA 13. DIAGRAMA DE BLOQUES DEL NODO EMISOR	55
FIGURA 14. DIAGRAMA DE BLOQUES DEL NODO REPETIDOR.....	56
FIGURA 15. DIAGRAMA DE BLOQUES DEL NODO CABECERA/MAESTRO.....	57
FIGURA 16. MEDIDAS DEL COMPLEJO DEPORTIVO DE FIEC.....	58
FIGURA 17. FOTOGRAFÍA DIAGONAL HACIA EL COMPLEJO DEPORTIVO.....	58
FIGURA 18. FOTOGRAFÍA LATERAL DEL COMPLEJO DEPORTIVO	59
FIGURA 19. MEDIDAS DEL PARQUE SAMANES	60
FIGURA 20. FOTOGRAFÍA 1 DE ZONA PRUEBA EN PARQUE SAMANES.....	60
FIGURA 21. FOTOGRAFÍA 2 DE ZONA DE PRUEBA.....	61
FIGURA 22. FOTOGRAFÍA SATELITAL DEL LAGO ESPOL	62
FIGURA 23. FOTOGRAFÍA DEL PERÍMETRO DEL LAGO ESPOL	62
FIGURA 24. TOPOLOGÍA DE CONDICIÓN IDEAL USANDO PAINLESSMESH	64
FIGURA 25. POSIBLE POSICIONAMIENTO DE NODOS EN CONDICIONES IDEALES CON PAINLESSMESH	64
FIGURA 26. TOPOLOGÍA DE CONDICIÓN IDEAL USANDO RF24 MESH.....	65
FIGURA 27. POSIBLE POSICIONAMIENTO DE NODOS EN CONDICIONES IDEALES CON RF24 MESH ...	66
FIGURA 28. TOPOLOGÍA PROPUESTA PARA EL ANÁLISIS DE LAS REDES	66
FIGURA 29. POSIBLE POSICIONAMIENTO DE NODOS EN LOS CASOS CRÍTICOS	67
FIGURA 30. MODELOS DE BATERÍAS USADAS	68
FIGURA 31. IMPLEMENTACIÓN DE NODOS EMISORES	70

FIGURA 32. IMPLEMENTACIÓN DE NODOS REPETIDORES.....	70
FIGURA 33. IMPLEMENTACIÓN DE NODOS CABECERA/MAESTRO Y ESCLAVO	71
FIGURA 34. IMPLEMENTACIÓN DE NODOS EMISORES	71
FIGURA 35. ESQUEMÁTICO DEL NODO INTERMEDIO USANDO RF24 MESH	72
FIGURA 36. ESQUEMÁTICO DEL NODO DE CABECERA/MAESTRO Y ESCLAVO USANDO RF24 MESH.	72
FIGURA 37. UBICACIÓN DE LOS NODOS PARA LA PRUEBA DE ENLACE.....	74
FIGURA 38. PRUEBAS DE ENLACE – RECEPCIÓN Y PUBLICACIÓN DE DATOS EN EL NODO ESCLAVO .	75
FIGURA 39. PRUEBAS DE ENLACE – PUBLICACIÓN DE DATOS EN THINGSPEAK.....	75
FIGURA 40. PRUEBAS DE ENLACE – RECEPCIÓN Y PUBLICACIÓN DE DATOS EN EL NODO ESCLAVO .	76
FIGURA 41. PRUEBAS DE ENLACE – PUBLICACIÓN DE DATOS EN THINGSPEAK.....	77
FIGURA 42. UBICACIÓN DE LOS NODOS PAINLESSMESH EN EL COMPLEJO DEPORTIVO DE ESPOL..	78
FIGURA 43. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 1 – PAINLESSMESH	79
FIGURA 44. DATOS DE LATENCIA OBTENIDOS DE ESCENARIO 1 – RF24	79
FIGURA 45. UBICACIÓN DE LOS NODOS RF24 EN EL COMPLEJO DEPORTIVO	80
FIGURA 46. UBICACIÓN DE LOS NODOS PAINLESSMESH EN EL PARQUE SAMANES.....	81
FIGURA 47. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 2 – PAINLESSMESH	82
FIGURA 48. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 2 – RF24.....	83
FIGURA 49. UBICACIÓN DE LOS NODOS RF24 EN PARQUE SAMANES	83
FIGURA 50. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 3: CASO 1 – PAINLESSMESH	85
FIGURA 51. UBICACIÓN DE LOS NODOS PAINLESSMESH EN EL LAGO ESPOL.....	85
FIGURA 52. DATOS OBTENIDOS EN THINGSPEAK - PAINLESSMESH	86
FIGURA 53. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 3: CASO 2 – PAINLESSMESH	87
FIGURA 54. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 3: CASO 1 – RF24.....	88
FIGURA 55. UBICACIÓN DE LOS NODOS RF24 EN EL LAGO ESPOL	88
FIGURA 56. UBICACIÓN DE LOS NODOS RF24 EN EL LAGO ESPOL	89
FIGURA 57. DATOS DE LATENCIA OBTENIDOS EN ESCENARIO 3: CASO 2 – RF24.....	90

Índice de tablas

TABLA 1. CARACTERÍSTICAS PRINCIPALES DEL MÓDULO ESP32.....	10
TABLA 2. RESUMEN DE DATOS RECOPIADOS DEL ESCENARIO 1	90
TABLA 3. RESUMEN DE DATOS RECOPIADOS DEL ESCENARIO 2	91
TABLA 4. RESUMEN DE DATOS RECOPIADOS DEL ESCENARIO 3 – CASO 1.....	91
TABLA 5. RESUMEN DE DATOS RECOPIADOS DEL ESCENARIO 3 – CASO 2.....	91
TABLA 6. DETALLE DE COSTOS USANDO PAINLESSMESH	92
TABLA 7. DETALLE DE COSTOS USANDO RF24 MESH	92

Capítulo 1

1.1 Introducción

La conectividad hacia la red se ha convertido en una parte esencial de la vida cotidiana, académica y en el mundo laboral; esto sigue creciendo exponencialmente con el transcurso del tiempo debido a la constante demanda de los servicios que se pueden ofrecer o por el simple hecho de compartir la información. Los diferentes entornos aplicables involucran no solo a las grandes ciudades, sino que poco a poco se han incorporado en sectores rurales y/o zonas remotas [1]. Este crecimiento exponencial se traduce en volumen de datos transferidos, crecimiento físico de las redes y tráfico interconectado, lo cual obliga a que los fabricantes innoven en productos más eficientes e inteligentes que utilicen nuevos métodos capaces de cubrir esta demanda. Por esta razón, los dispositivos electrónicos se encuentran en un gran auge y surge la importancia de los dispositivos de Internet de las Cosas (IoT) [2].

Para este año 2023, según el informe anual de internet de CISCO [3], la cantidad de dispositivos conectados a redes IP será tres veces la población mundial, y las conexiones Machine to Machine (M2M), que es referenciada a las aplicaciones de dispositivos IoT, alcanzará los 14.7 billones de conexiones.

El término Internet de las Cosas (IoT) se refiere a un conjunto de dispositivos electrónicos interconectados que poseen identificadores únicos (UID) y tienen la habilidad de intercambiar información a través de una red. Un entorno de IoT está conformado por dispositivos inteligentes enlazados a la web que emplean sistemas embebidos, como chips, detectores y componentes de comunicación, para recolectar, transmitir y responder a los datos obtenidos de su entorno [4]. Estas plataformas de IoT han sido ampliamente utilizadas con éxito por las Wireless Mesh Networks (WMN) [5], que consisten en una red inalámbrica conformada de múltiples saltos y establece una estructura en forma de malla. Puede ser implementada sin necesidad de cables o una infraestructura principal de soporte, lo cual la convierte en una opción de bajo costo.

Los bajos precios de implementación y solucionar de manera eficiente los problemas de cobertura, ha llamado la atención del sector empresarial/industrial que realiza sus actividades en zonas remotas, tal como el sector minero. Por ejemplo, en “*An Emergency Communication System Based on WMN in Underground Mine*” [5] se muestra el desarrollo de un sistema de comunicación de emergencia basado en WMN en una mina subterránea, capaz de realizar la comunicación inalámbrica basada en voz sobre IP (VoIP) y el protocolo de inicio de sesión (SIP), además de aplicaciones de video y sensores de monitoreo. Por este motivo, el presente proyecto propone el diseño de una WMN de bajo costo y de amplia cobertura, enfocada en la conexión dispositivos IoT para aplicaciones en zonas remotas como sería el sector camaronero.

1.2 Descripción del problema

Según los registros de la Cámara Nacional de Acuicultura (CNA), desde octubre de 2017, el camarón se posicionó como el principal producto de exportación no petrolera de nuestra nación. La creciente demanda experimentada generó una mayor carga de trabajo en los laboratorios de cría de larvas, los cuales se vieron forzados a redoblar sus esfuerzos para aumentar la producción, sacrificando algunos procesos y controles indispensables para hacer frente a la vibriosis que había estado afectando a este producto desde 2016 [6].

En la industria acuícola, el cultivo de camarones está condicionado por factores ambientales y químicos presentes en el agua; dicha actividad se ve frecuentemente perturbada por la aparición de brotes de diversas enfermedades, convirtiéndose en uno de los principales desafíos a abordar. Aunque resulta difícil determinar las causas exactas de estos brotes, las pruebas indican que su desarrollo puede estar relacionado con factores ambientales. Ya que la susceptibilidad a través de factores de estrés del entorno afecta la capacidad de respuesta de los organismos frente a cualquier patógeno o factor no biológico presente. Comprender los mecanismos de interacción de estos agentes puede ayudar a identificar las principales causas de enfermedades en sistemas de cultivo [8]. Por lo general, la recolección y registro de información relacionada con estos factores

se lleva a cabo de forma tradicional, mediante métodos manuales, los que no están exentos del error humano [9].

Como indican en *Design and Deploy a Wireless Sensor Network for Precision Agriculture* [10], las mejoras con aplicaciones de sensores inalámbricos impactan positivamente en la eficiencia, productividad y rentabilidad de diversos sistemas de producción, al mismo tiempo que reducen los posibles errores humanos y efectos indeseados. La recopilación de información en tiempo real permite a los productores adaptar sus estrategias de manera oportuna y fundamentada en las mediciones automatizadas de estos sensores.

El tema de cobertura para una red WiFi siempre será un punto importante para tomar en cuenta, y en una red mallada no es excepción. En “*Banco de pruebas de una Red Inalámbrica Mesh basada en el protocolo IEEE 802.11s*” [11], mencionan la gran acogida que están tomando las WMN en estos últimos años y de cómo han sido direccionadas en aplicaciones IoT de pequeña cobertura como; redes domésticas de banda ancha, redes de una comunidad de vecinos, redes empresariales, redes públicas, o redes en lugares donde es muy difícil y costosa la implantación de una red cableada. Las distancias entre nodos usualmente son pequeñas, alcanzando menos de 50 metros entre ellos.

Para una industria en general, el procesamiento de información es siempre un tema de vital discusión. En “*The Network as a Computer: A Framework for Distributed Computing Over IoT Mesh Networks*” [13], se habla de la importancia de que un sistema IoT responda rápidamente a eventos externos, debido a restricciones de tiempo estrictas impuestas. Esto puede deberse a la naturaleza crítica del proceso de monitoreo o para minimizar los costos derivados de demorar innecesariamente las acciones de control adecuadas en una industria. Por consiguiente, resulta esencial reducir al mínimo la latencia de estas aplicaciones, es decir, el tiempo requerido para transformar las lecturas de entrada sin procesar de los sensores en la señal de salida final en los nodos de actuación.

1.3 Justificación del problema

El monitoreo en tiempo real de dispositivos IoT es una parte vital para la industria, ya que con ellos se agiliza la producción, ahorra costos y mano de obra, eficiencia y precisión en estas áreas apartadas [6]. Tener una “piscina inteligente” es abastecerla con dispositivos y sensores IoT que brinden múltiples beneficios, tal como expresa [12], una empresa especializada en sistemas de monitoreo en tiempo real en el control automático de alimentos y de los parámetros de calidad del agua en las piscinas camaroneras. Los más destacables son los siguientes:

- Mejora la tasa de crecimiento
- Disminuye el factor de conversión alimenticia (FCA)
- Aumenta la tasa de supervivencia
- Reducción de costos en producción

La CNA se ha pronunciado respecto al alza de los costos de producción y de cómo la delincuencia organizada afecta directamente a esta industria. La problemática radica en que los pequeños productores de camarón no tienen el poder adquisitivo para llevar a cabo la implementación de un sistema de monitoreo de este calibre. Por ese motivo, se analiza la posibilidad de implementar una red Mesh con sistemas programables a bajo costo, con el objetivo de evaluar y analizar su funcionamiento adaptable hacia el entorno IoT o necesidades de la industria acuícola.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar una red Mesh de máxima cobertura utilizando sistemas programables de bajo costo que permitan la conexión y el intercambio de datos entre los dispositivos IoT dentro del sector acuícola.

1.4.2 Objetivos específicos

1. Analizar la cobertura de una red Mesh de microcontroladores ESP32 para la estimación de conexión de la red en relación con la cantidad y la distancia entre los nodos conectados a la misma.
2. Diseñar una red Mesh de arquitectura modular para que permita la inclusión de dispositivos IoT que lleven a cabo el monitoreo y control en tiempo real en una camaronera.
3. Implementar una red Mesh optimizada y resiliente que tenga la capacidad de comunicación constante y selección de la mejor ruta de transmisión entre los nodos para los cambios de la red en el tiempo, dándole flexibilidad y escalabilidad.

1.5 Marco teórico

En la actualidad, el empleo de redes de comunicación y el acceso a Internet se ha vuelto esencial tanto en el ámbito empresarial, instituciones educativas y dentro de los hogares. Esta creciente necesidad de conectividad y constante intercambio de información ha impulsado el desarrollo de una solución eficiente en las redes Mesh [1].

1.5.1 ¿Qué es una Red Mesh?

La tecnología de red Mesh presenta un enfoque revolucionario para la comunicación, donde cada nodo se enlaza de forma directa con múltiples dispositivos, generando una configuración en malla. A diferencia de los sistemas convencionales con un enrutador central, en una red Mesh, cada dispositivo posee la capacidad de transmitir datos tanto a dispositivos cercanos como de retransmitir información para alcanzar nodos más alejados en la red [1].

Este método de comunicación descentralizada facilita una transferencia de datos eficiente, evitando congestiones y aumentando la resistencia ante posibles fallas. Mediante la utilización de múltiples nodos fijos o móviles, la información puede encontrar la ruta óptima para llegar a su destino, lo cual resulta especialmente ventajoso en entornos extensos o de difícil acceso [14].

Por ejemplo, en edificaciones de grandes dimensiones, zonas urbanas densas o áreas al aire libre, las redes Mesh proporcionan una cobertura amplia y fiable. Una de las características más notable de las redes Mesh radica en su capacidad de autoorganización y autoconfiguración [5]. Esto implica que los dispositivos pueden unirse o abandonar la red sin requerir configuración manual, lo que brinda un sistema escalable y flexible. La topología de la red se ajusta automáticamente a los cambios, asegurando un funcionamiento eficiente y adaptado a las demandas del entorno [6]. Dicha adaptabilidad puede ser observada en *Figura 1*. Las topologías de malla pueden ser adaptada dependiendo de las necesidades del usuario: tomando forma de una malla completa, permitiendo que todos los dispositivos puedan estar enlazados entre todos o la forma de una malla parcial, esta última forma permite el enlace solo entre algunos de los dispositivos en referencia un nodo. En la gran mayoría de casos se utiliza solo la malla parcial [15], estas formas de topología pueden ser vistas en *Figura 2*.

Figura 1

Arquitectura de una red Wi-Fi Mesh

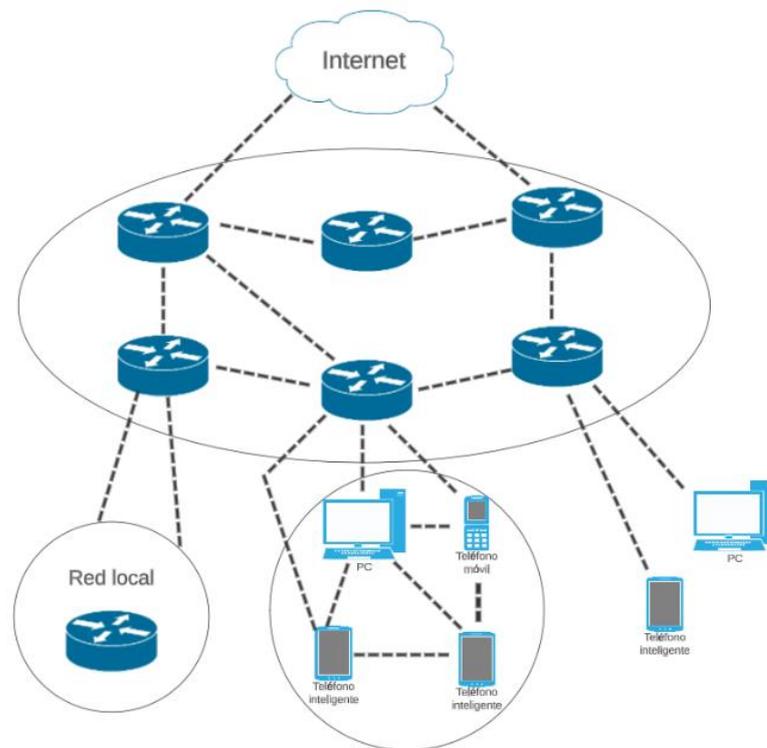
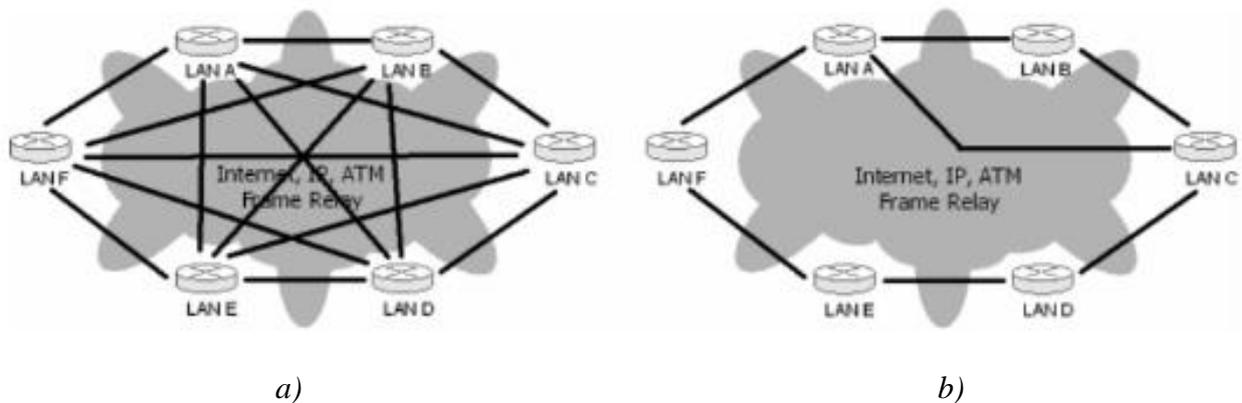


Figura 2

Topología de a) malla completa y b) parcial [15]



1.5.2 ¿Qué es el ESP32 Device Kit?

El ESP32 es un microcontrolador de bajo consumo de energía y alta capacidad de procesamiento, diseñado por la empresa Espressif Systems. Es ampliamente utilizado en proyectos de electrónica y desarrollo de sistemas embebidos debido a su versatilidad y potencia. Este microcontrolador está basado en la arquitectura de 32 bits y cuenta con una unidad central de procesamiento (CPU) de doble núcleo, lo que le permite realizar tareas de manera eficiente y multitarea.

Una de las principales características del ESP32 es su capacidad de conectividad inalámbrica. Este microcontrolador cuenta con soporte integrado para Wi-Fi, lo que le permite establecer conexiones a redes inalámbricas y comunicarse con otros dispositivos a través de Internet. Además, tiene soporte para Bluetooth, lo que lo hace compatible con dispositivos de este tipo y permite la comunicación inalámbrica de corto alcance.

El ESP32 posee una amplia variedad de periféricos y pines de entrada/salida (E/S), convirtiéndolo en una herramienta muy flexible para el desarrollo de proyectos electrónicos. Entre sus periféricos se incluyen interfaces UART, SPI, I2C, ADC (convertidor analógico-digital), DAC (convertidor digital-analógico) y GPIO (entrada/salida de propósito general), que permiten la conexión y control de diversos dispositivos y sensores externos.

Este microcontrolador es ampliamente utilizado en diferentes aplicaciones, como el Internet de las cosas (IoT), sistemas de automatización del hogar, monitoreo y control de sensores, desarrollo de dispositivos móviles, sistemas de seguridad, entre otros. Gracias a su potencia de procesamiento, conectividad inalámbrica y capacidades de E/S, el ESP32 se ha convertido en una opción popular para proyectos que requieren un control y comunicación eficientes. [16]. La información sobre los periféricos puede ser observado en *Figura 3*. Algunos datos adicionales se encuentran en *Tabla 1*.

Se debe tener en cuenta que existen varios modelos de ESP32, las cuales dependen completamente de las necesidades del usuario, existiendo variaciones en sus versiones de sus respectivos modelos.

Figura 3

Pines del ESP32 [17]

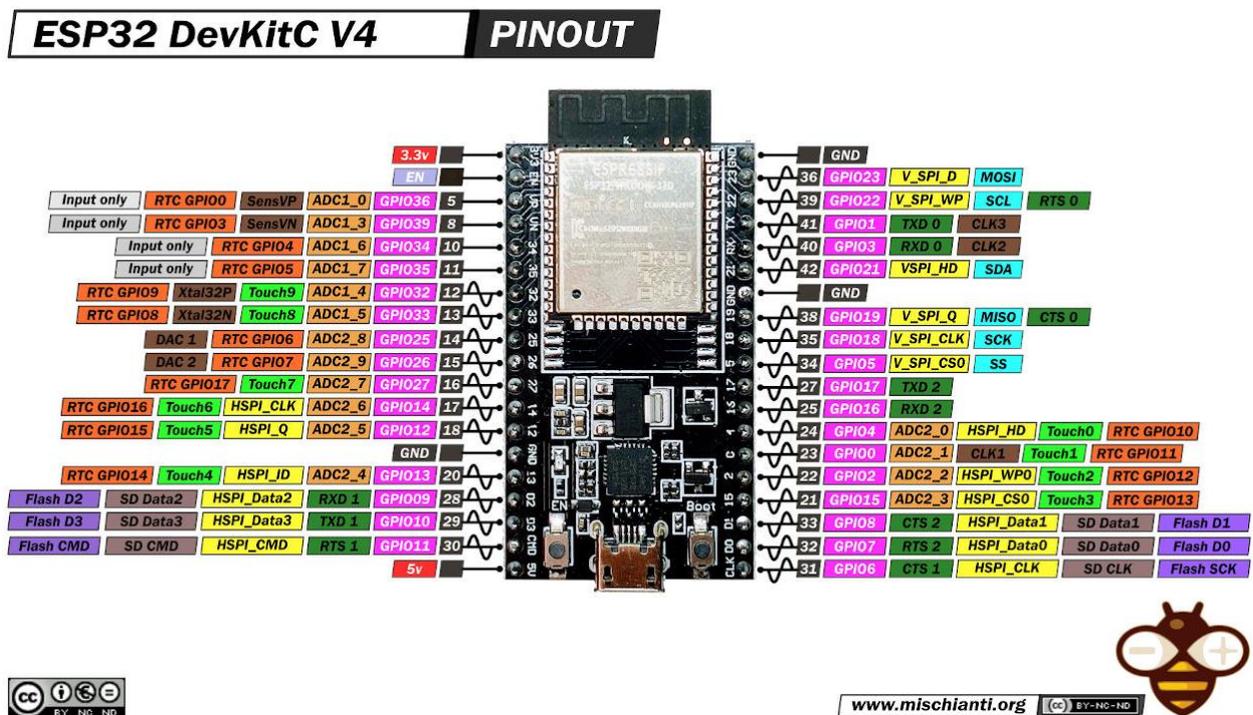


Tabla 1

Características principales del módulo ESP32 [18].

Módulo ESP32	Características
Diseño Robusto	Temperatura de funcionamiento que oscila entre $-40\text{ }^{\circ}\text{C}$ y $+125\text{ }^{\circ}\text{C}$.
Consumo ultra bajo de energía	Vin: 3.3V – 5.5V Corriente: $10\mu\text{A}$ Tecnología de escalado dinámico de potencia
Alto nivel de integración	Altamente integrado con interruptores de antena incorporados, balun de RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros y módulos de administración de energía.
Chip híbrido Wi-Fi y Bluetooth	Puede interactuar con otros sistemas para proporcionar funcionalidad Wi-Fi y Bluetooth a través de sus interfaces SPI/SDIO o I2C/UART.

1.5.3 *painlessMesh*

PainlessMesh es una librería de código abierto creada para facilitar la implementación de redes Mesh por medio de ESP32 y ESP8266, su primera versión fue lanzada en 2018 en GitHub con aporte de varias personas [19]. Permite al usuario establecer una red de este tipo sin la preocupación de su estructura o de la administración de la red, ya que realiza el rápido acoplamiento de un nuevo nodo en la red, buscando la optimización de la red. La cantidad máxima de dispositivos en la malla dependerá de la memoria de cada nodo, debido a la captación de información. *PainlessMesh* no forma una red TCP/IP, crea una red local donde los nodos poseen en su chip de 32 bits un kit de desarrollo de software (SDK) con su identificador [20]. La función *mesh.update* mantiene la sincronización de la red Mesh, ya se intercambian datos y estados de los diversos nodos enganchados, además de la sincronización de los nodos por medio de un reloj general generado por el nodo de cabecera. Esta librería permite la formación de una red por medio de diversos tipos de topología, siendo la más usual la topología tipo malla, pero dependiendo de las necesidades del usuario puede adoptar topologías mixtas. Al momento de implementarse, no permite que el usuario pueda modificar el identificador (ID) de cada nodo y su salida es directa de

la antena PCB implementada en el microcontrolador, por lo que la potencia de transmisión y la ganancia de recepción no pueden ser alteradas [21].

1.5.4 nRF24L01+PA+LNA

El nRF24L01+ es un módulo transceiver (transmisor y receptor) que trabaja en la frecuencia de 2.4 GHz fabricado por Nordic Semiconductor. Es ideal para la comunicación de proyectos de manera inalámbrica y posee un consumo ultra bajo. El módulo integra un transceiver completo de 2.4 GHz, sintetizador RF y lógica con el mejorado ShockBurst (acelerador de protocolo por software para la comunicación SPI con el microcontrolador). Esta versión del módulo posee un amplificador de potencia (PA), un circuito de amplificación de bajo ruido (LNA) y una antena SMA omnidireccional que le permite lograr un rango de cobertura de hasta 1 km con línea de vista. Posee canales de 125 canales de transmisión con una banda de 1MHz, lo que permite su trabajo desde los 2400 MHz hasta los 2525 MHz [22].

El módulo posee 8 pines a los que se le brinda una alimentación de 3.3 voltios (siendo tolerante a 5 voltios) y se comunica por medio de SPI. Este módulo puede transmitir a 3 velocidades de datos diferentes: 250 kbps, 1Mbps y 2Mbps; así como a diferentes niveles de potencia: -18 dBm, -12 dBm, -6 dBm y 0dBm. También cuenta con la opción de poder activar el LNA y darle al sistema una ganancia de recepción hasta 20 dBm [23]. La información sobre sus periféricos y observación del modelo, se lo puede observar en por medio de *Figura 4* y *Figura 5*.

Figura 4

Pines de salida del módulo nRF24L01+PA+LNA [24]

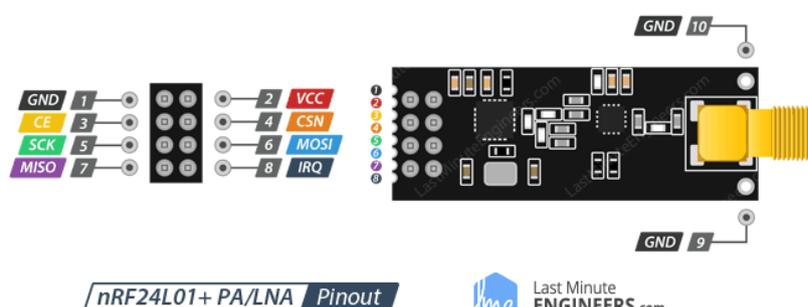
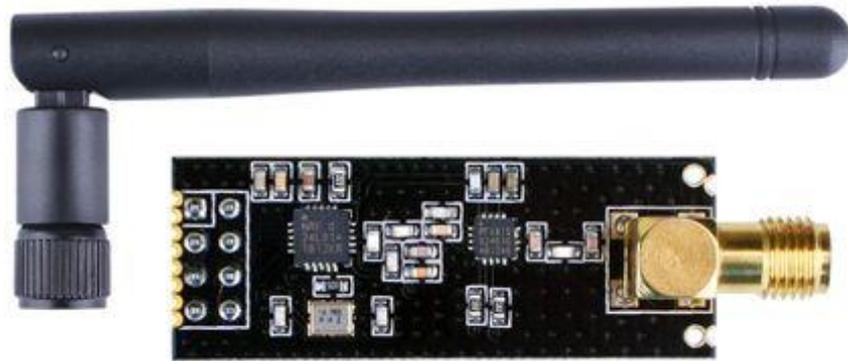


Figura 5

Pines de salida del módulo nRF24L01+PA+LNA [25]



1.5.5 RF24

RF24 es una librería que permite la comunicación por medio de radiofrecuencias, en específico en la banda de 2.4 GHz a través de los módulos nRF24L01+. Representa la capa 2 del modelo OSI para el enlace de los datos, es decir, permite el direccionamiento físico de la información que desea ser enviada o recibida. Es de fácil implementación y ofrece opciones de configuraciones avanzadas para el usuario. [26]

1.5.6 RF24 Network

RF24 Network es una librería que pertenece a la familia de RF24, el cual proporciona una capa de red simple y transparente para redes de sensores IoT. Representa la capa 3 (capa de red) del modelo OSI, es decir, para determinar la ruta y direccionamiento lógico. Incluye el enrutamiento, direccionamiento, reensamblaje y fragmentación de la red. Es de fácil implementación y tiene como requisito la adquisición de la librería RF24 [27].

Una de las grandes diferencias en comparación a painlessMesh es la topología que se puede implementar. RF24 Network solo ofrece una red con una topología tipo árbol llegando a tener limitantes dentro de la red, ya que solo permite la implementación de hasta 781 nodos en su totalidad y donde cada nodo podrá tener hasta un máximo de 5 nodos hijos. Los identificadores de

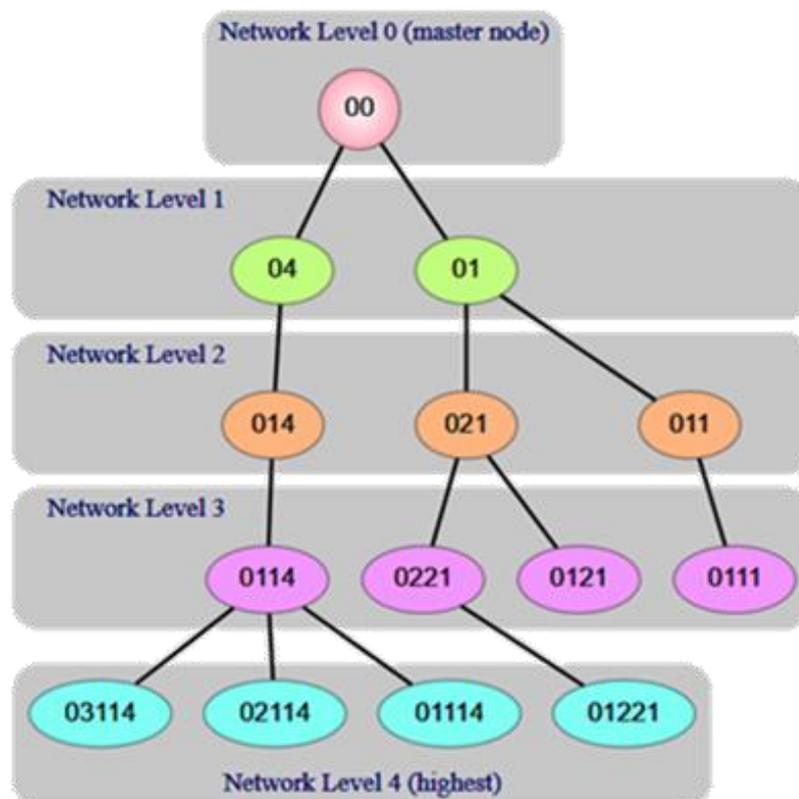
red no son valores aleatorios, estos son definidos por parte del nodo de cabecera y se encuentran clasificados dentro de los 4 niveles de red establecidos. Esto se describe de la siguiente manera:

- El nodo con identificador 00 está reservado exclusivamente para el nodo base o de cabecera.
- Los nodos 01 hasta el 05 son los nodos hijos del nodo base.
- El nodo 021 es el segundo nodo hijo del nodo 01.
- El nodo 0321 sería el tercer hijo del nodo 021, y así sucesivamente.
- La dirección de nodo más grande es 05555 (correspondiente al quinto hijo del nodo 0555) [28].

Este ejemplo puede ser visualizado con la *Figura 6*.

Figura 6

Estructura de nodos en RF24 Network [28]



1.5.7 RF24 Mesh

RF24 Mesh es una librería perteneciente a la familia de RF24 que permite la estructuración de una red tipo Mesh por medio de módulos nRF24L01+, representa la capa 7 del modelo OSI para el servicio de red a aplicaciones. Permite la implementación de redes de malla automatizadas e inalámbricas para una red de sensores o dispositivos IoT [29]. Automatiza el direccionamiento, descubrimiento y operación de la red comunicada por RF24. En esta red autónoma solo se puede implementar hasta un total de 256 nodos. En este tipo de red se mantiene reservado su ID con valor 0 al nodo de cabecera o base. Este ID no se encuentra relacionado con el identificador de RF24 Network y deben ser establecidos previo a la activación del microcontrolador. A lo largo de esta red, los nodos se pueden moverse físicamente y podrán cambiar su identificador de red, reconfigurándose a conveniencia para mantenerse dentro de la red Mesh, siempre y cuando se respeten las limitantes de RF24 Network. El nodo de cabecera tiene como responsabilidad realizar el seguimiento de los IDs de los nodos enlazados a la red y sus direcciones asignadas en RF24 Network. Tiene como requisito la instalación previa de las librerías RF24 y RF24 Network [30].

1.6 Antecedentes

En *Wireless Mesh Networks in IoT networks* (2017) se realizó un análisis de las ventajas de adquirir una red Wi-Fi Mesh, la cual permite una disminución de consumo en la implementación y soporte de las redes usando dispositivos más pequeños que cumplan casi la misma función, con el fin de mantener el control e interacción entre estos. Los sensores pueden ubicarse en lugares de difícil acceso o de conectividad, limitando el alcance que tiene el enrutador central de la red debido a la posición en la que debe estar colocado para obtener una mayor cobertura, esta limitante impide alcanzar a dichos sensores. Debido a esta situación, una solución viable es la implementación de redes de malla inalámbrica (WMN), siendo conformada por un conjunto de nodos organizados en una topología de malla en lugar de una topología de estrella usada en la mayoría de las redes. De aquí se establece una estructura de red troncal usando la

WMN, que permite la conectividad de los clientes hacia los nodos. En este documento se ejemplificaba la red usando enrutadores para la formación de la troncal por medio de enlaces de autoconfiguración, permitiendo la conectividad a Internet. Esta red es de mayor simpleza, muy usada y fácil de integrar con dispositivos existentes, donde solo los enrutadores deben ser instalados en la red de malla. [31]

Espressif System, fabricante del ESP32, difundió la primera versión pública de la librería para implementar redes MESH usando los módulos ESP32. Actualmente, la librería se encuentra en la versión 3.3.5. ESP-MESH permite la implementación de un protocolo de red y el enlace de numerosos dispositivos repartidos en un área física grande (tanto en zonas interiores como exteriores) interconectadas en una sola red inalámbrica de área local (WLAN). En caso de que un nodo se retire de la red, el protocolo permite la autoreparación y reorganización de esta, manteniéndose de manera autónoma [32].

Durante una conferencia, Urvi Bhagat, Nikhil Gujar y Sunil Patel (2018) mencionaron la implementación de una red Wi-Fi Mesh, en la que cada uno de sus 10 nodos finales (luminarias conectadas a un ESP32) esté conectada a una nube, para que, por medio de un servidor MQTT y un control máster, se pueda mantener el control de las iluminarias de un campus universitario (la distancia promedio entre luminarias es de 30 metros aproximadamente), logrando la mejora de la eficiencia del sistema eléctrico en un 30% debido a la automatización implementada con sensores enlazadas al ESP32 correspondiente a cada nodo, esta implementación fue desarrollada para un entorno outdoor. [33]

En el artículo [34], Allan Guevara y Héctor Morán (2022) publicaron la implementación de una red IoT Mesh ad-hoc para interiores usando 5 microcontroladores: 2 ESP32, 2 Lopy4 y un Sonoff; haciendo uso de un Raspberry Pi como servidor. Entre los resultados de la investigación, se logró obtener un análisis de latencia con respecto a 3 escenarios (línea de vista, sin línea de vista y transmisión de larga distancia) con 3 diferentes tiempos de muestreo. El análisis indicó que el

peor tiempo de muestreo es el de 1 segundo debido a la enorme pérdida de paquetes, se obtiene en promedio un 21% de paquetes recibidos de manera exitosa; el escenario con un tiempo de muestreo de 3 minutos presentaba una equivalencia entre los 3 escenarios, donde se lograba un mínimo y máximo de 45 y 51% respectivamente de paquetes recibidos; y finalmente, cuando el tiempo de muestreo es de 1 minuto se obtiene en promedio entre un 89% y 93% de datos recibidos de manera exitosa. Este estudio fue desarrollado en un entorno indoor.

Capítulo 2

2.1 Estructura de investigación

Para el desarrollo de este proyecto se tuvo como fundamento el uso del microcontrolador ESP32 en conjunto con la implementación de las redes Mesh. Estas redes fueron formadas por 6 dispositivos ESP32, la plataforma ThingSpeak, una laptop, 1 capacitor de $10\mu\text{F}$ -50V y 5 módulos nRF24L01+PA+LNA con sus respectivas antenas; siendo puesta a prueba en tres escenarios: alrededor de una zona deportiva (*Figura 17*), en un parque (*Figura 20*) y una zona acuática (*Figura 23*), donde los nodos serían puestos alrededor de estas zonas; la distancia mínima dispuesta a cubrir es de aproximadamente de 786.13 m correspondientes al perímetro de una piscina camaronera presentada en *Figura 7*, tomando en cuenta que la red debe contar con resiliencia, escalabilidad y disponibilidad.

Figura 7

Fotografía satelital de la piscina camaronera



Los datos analizados fueron en base, de manera inicial a las distancias máximas de conectividad entre los dispositivos de la red, por medio del envío de mensajes indicando el

identificador del nodo o por el envío de los datos por parte del emisor hasta su máxima distancia para proceder a enlazar otro dispositivo.

Una vez obtenida la distancia máxima de cobertura de los dispositivos, se procede a agregar poco a poco el resto de los microcontroladores para realizar el contraste de la calidad de la red contra la cantidad de dispositivos conectados. Finalmente se analizaron las interferencias presentes en el entorno¹, con objetos u otros dispositivos que trabajen en la frecuencia de 2.4GHz en los escenarios propuestos de este proyecto para obtener los datos de latencia y realizar gráficos para su posterior análisis en base al rendimiento de la red.

Para la obtención de los datos de latencia, se debe emplear el método ping, el cual consiste en el envío de un mensaje desde el nodo que recibe los datos hacia el que los envía, este último lo reenvía y se obtiene el tiempo de ida-vuelta de dicho mensaje. El mensaje que se enviará solo estará conformado por un byte (8 bits), lo cual es representado por un carácter.

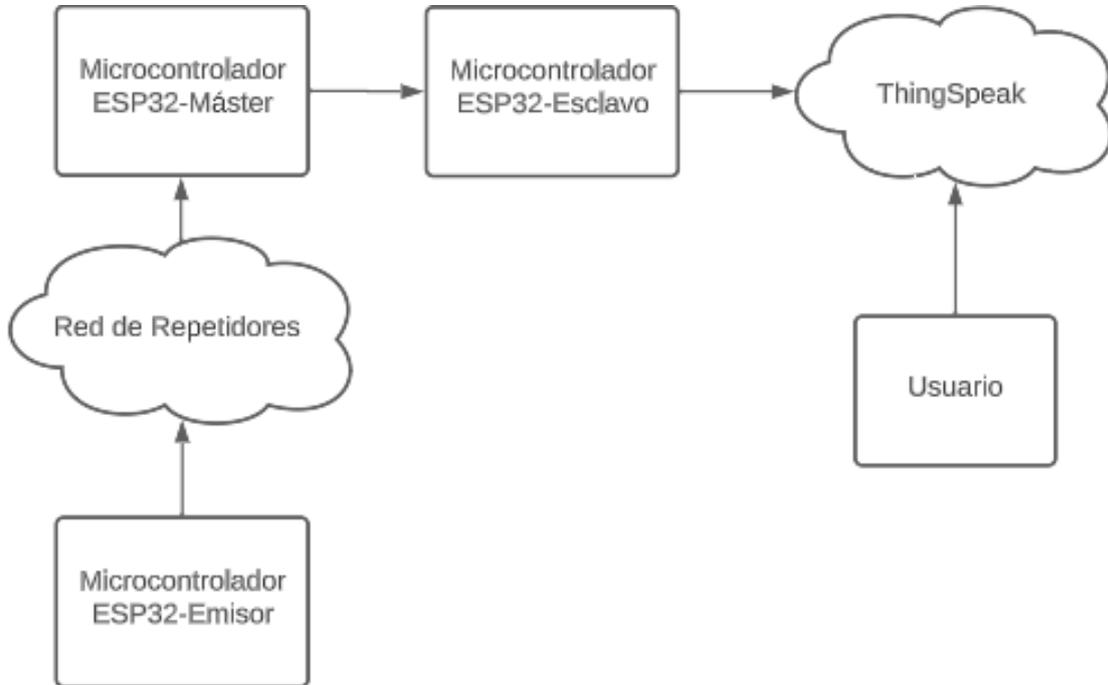
Se debe considerar que, al trabajar en las distancias máximas, esto implica que los dispositivos están trabajando con el nivel mínimo requerido de sensibilidad en su receptor para poder obtener los datos que están siendo enviados.

Por lo tanto, se planteó un diagrama de bloques que represente la transición de datos y estructura de la red implementada, que consta de: un nodo emisor, encargado del envío de datos hacia la red; nodos intermedios, elementos que ayuden en la transmisión de datos hacia el nodo de cabecera; nodo de cabecera/maestro, que permita la recepción de datos por parte del emisor y realice pruebas de latencia de la red, al obtener estos datos, los envíe hacia un nodo esclavo; finalmente, el nodo esclavo está encargado del envío de datos hacia la plataforma ThingSpeak, enlazándose por medio de Internet. Todo esto puede ser observado por medio de *Figura 8*.

¹ En su mayoría este tipo de interferencias se darán por agentes no controlados del ambiente como árboles, usuarios dentro de las zonas de pruebas, equipos que operen en la frecuencia de 2.4 GHz, edificaciones

Figura 8

Diagrama de bloques de la red



2.1.1 Librería *painlessMesh*

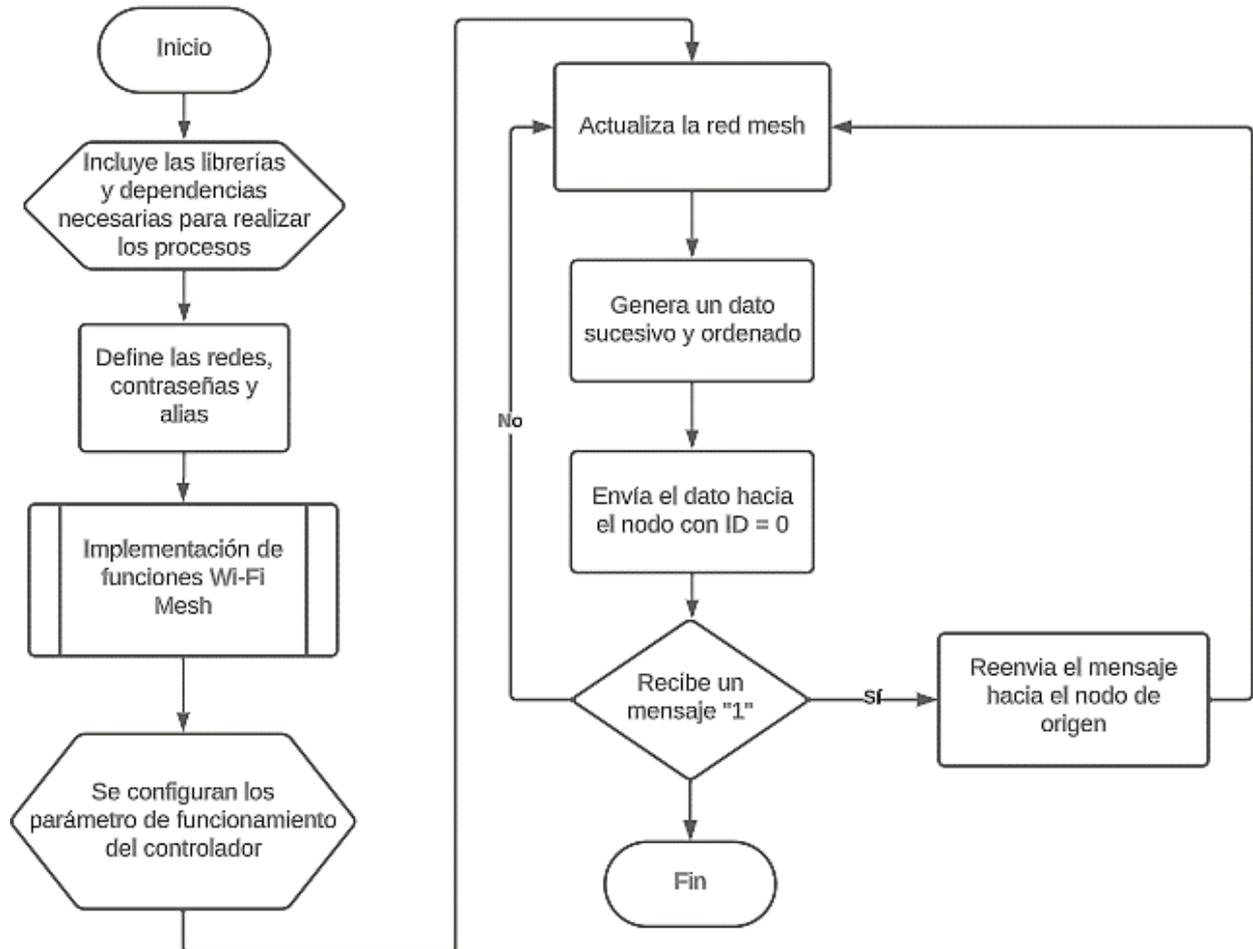
En esta sección se detallarán las funciones que tendrán cada uno de los nodos usando la librería *painlessMesh* y la presentación de diagramas de bloques de estos.

Nodo emisor. Genera un dato entero de manera sucesiva y ordenada (toma valores desde 0 y va en aumento en uno cada 5 segundos). Este valor de dato entero tuvo que ser procesado para poder transformarlo en una cadena de caracteres² y poder ser enviado a través de la red. El nodo se mantiene enlazado a la red Wi-Fi Mesh, obteniendo actualizaciones por parte de los otros nodos sobre la red, sincronizándose con la red y realiza el respectivo envío de sus datos a lo largo de la misma red. Si recibe un mensaje con el carácter “1”, lo reenvía hacia el nodo de origen. Su diagrama de bloques está presentado en *Figura 9*.

² Se debe aclarar que *painlessMesh* solo permite el envío de cadenas caracteres y no valores enteros, flotantes o de otro tipo. Por lo que todos los mensajes que se envíen se encontrarán limitados a este tipo de dato.

Figura 9

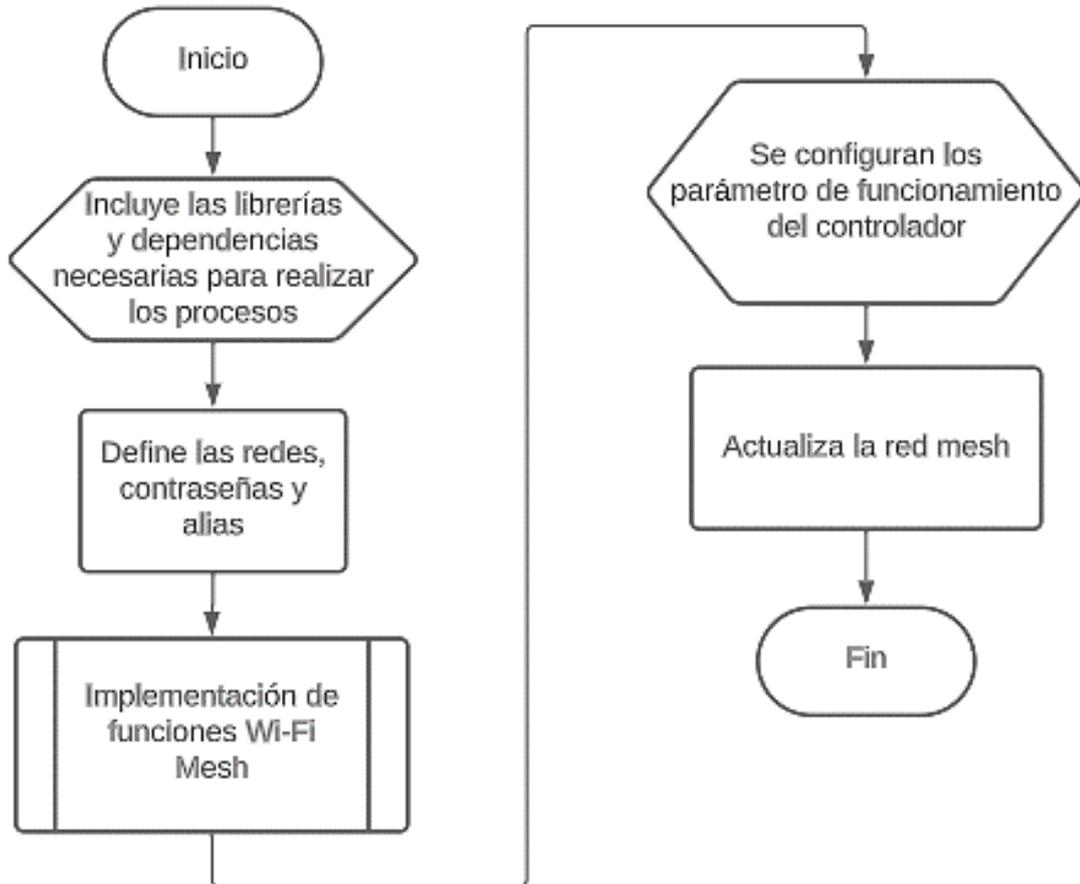
Diagrama de bloques del nodo emisor



Nodo repetidor. Este nodo recopila los datos enviados por parte de los nodos emisores a los que se encontró enlazados. En caso de que estos mensajes fueran hacia este nodo, este los procesaría; caso contrario, ayudaría en el envío del dato transmitido hacia el nodo que le corresponda. Actualiza constantemente el estado de la red y de sus nodos, manteniendo sincronización de esta. En caso de existir algún cambio, debe notificarlo hacia los demás dispositivos para mantener actualizada la lista de nodos activos con sus respectivos IDs. Su diagrama de bloques se muestra en *Figura 10*.

Figura 10

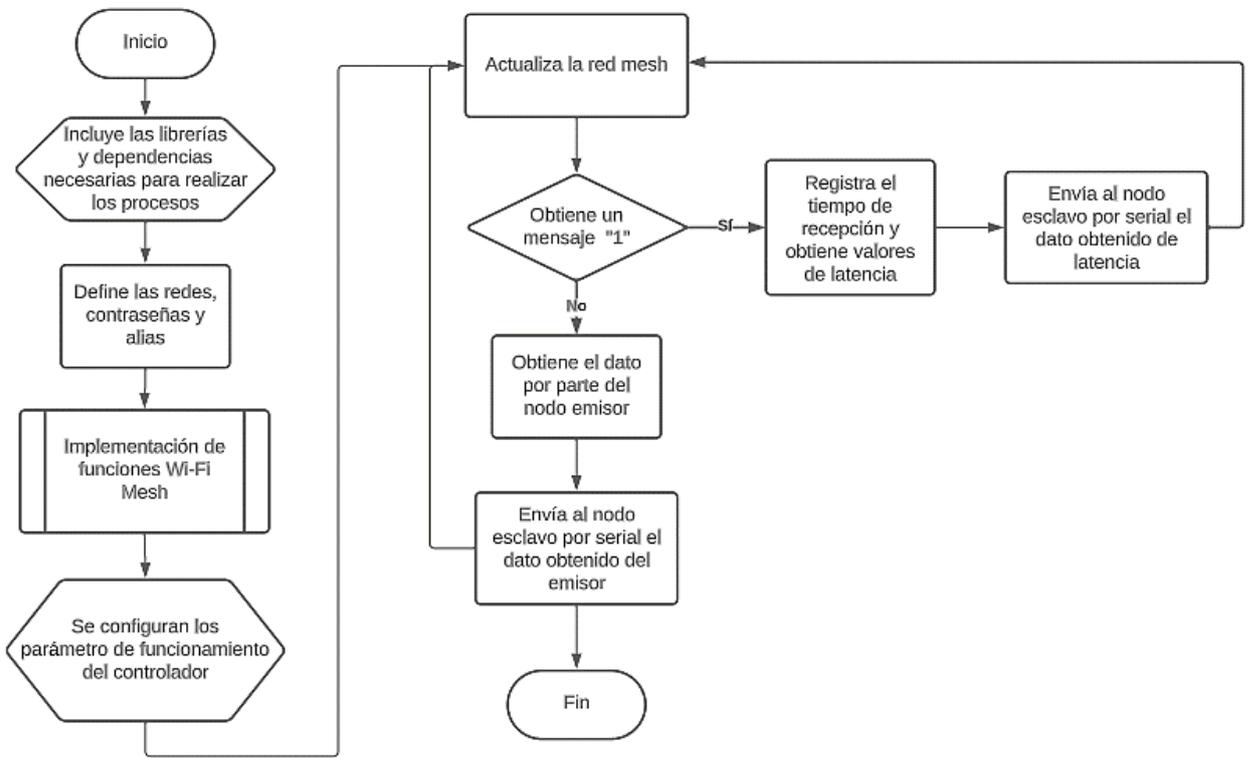
Diagrama de bloques del nodo repetidor



Nodo de cabecera/maestro. El nodo de cabecera/maestro obtiene todos los datos reorientados por medio de los nodos repetidores. Cada vez que recibe un mensaje con datos del emisor, envía un mensaje de un solo byte, representado por parte de un carácter “1” para poder registrar el tiempo de envío en referencia a este nodo, se envía de regreso el mismo carácter y registra su tiempo de recibo. Estos tiempos son procesados para obtener la latencia en milisegundos. Tras obtener los datos por parte del emisor y el registro de la latencia obtenida, envía esta información hacia el nodo esclavo por medio de la comunicación serial. El diagrama de flujo de este nodo puede ser apreciado en *Figura 11*.

Figura 11

Diagrama de bloques del nodo cabecera/maestro

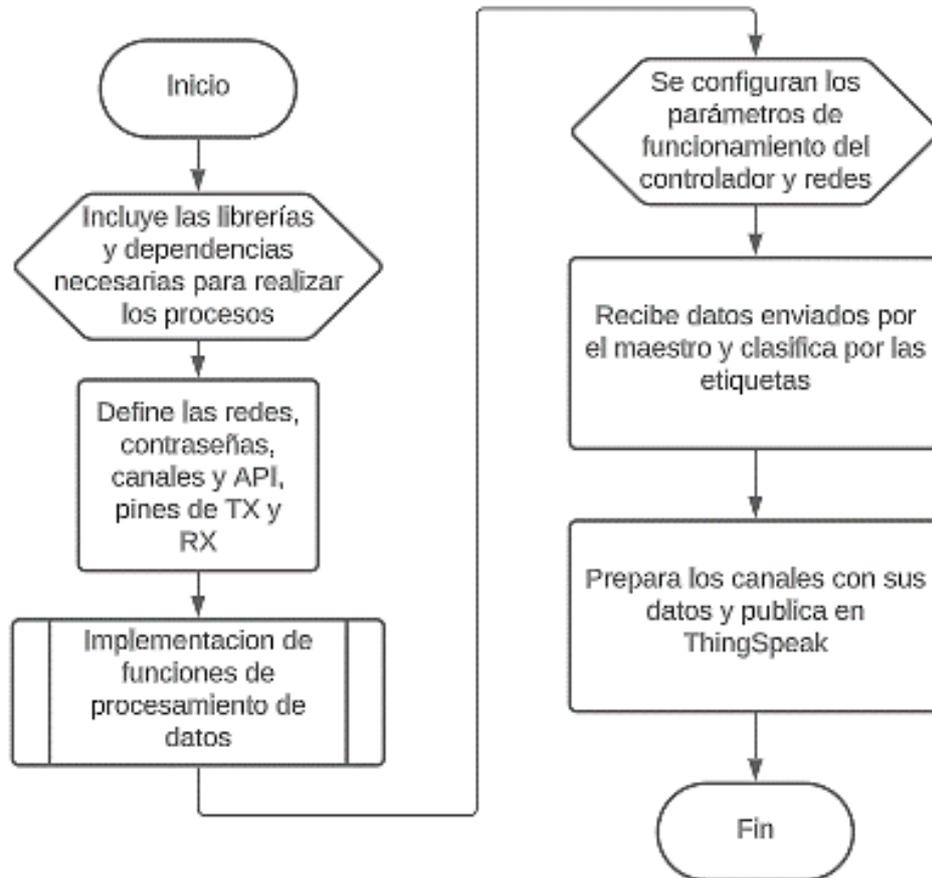


Nodo esclavo. Recibe todos los datos procesados del nodo maestro por parte de la comunicación serial en un serial con diferente tasa de datos a la de presentación de datos por pantalla. Se enlaza a una red Wi-Fi emitida por una laptop³, procesa los datos recibidos por parte del nodo maestro en base a las etiquetas de escritura de los datos. Las etiquetas se deban por medio de la recepción de la cadena de caracteres, empezando con “T” o “L”. Prepara los campos de escritura de ThingSpeak y envió los a ThingSpeak en su canal respectivo. El diagrama de flujo del nodo se visualiza en *Figura 12*.

³ Se escogió una red Wi-Fi emitida por un Hot-Spot de una laptop debido a la conveniencia de movilidad para la realización de las pruebas de este proyecto, el nodo puede ser configurado para conectarse directamente a un enrutador o dejarlo directamente por algún otro dispositivo móvil. La frecuencia de trabajo establecida para esta red Wi-Fi fue de 2.4 GHz

Figura 12

Diagrama de bloques del nodo esclavo

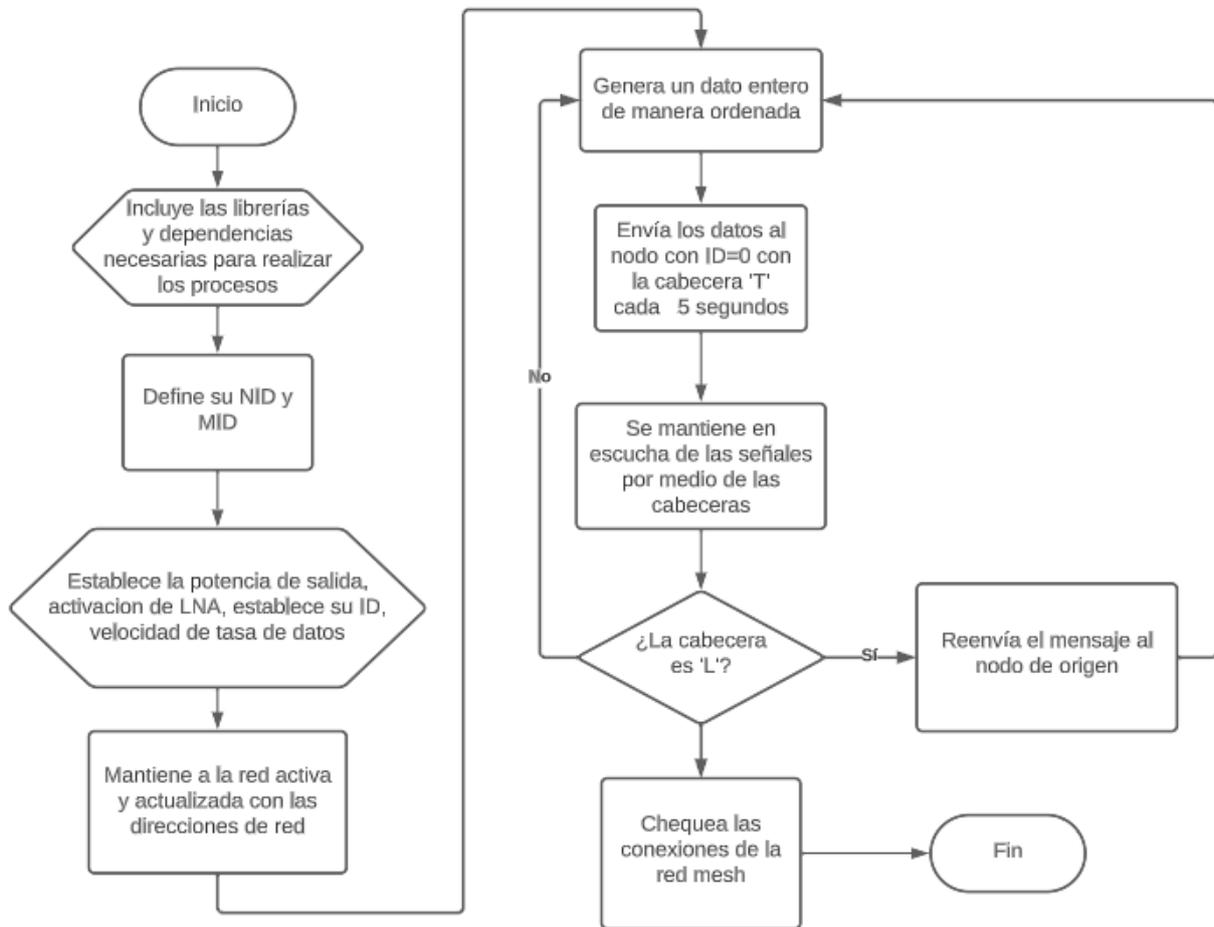


2.1.2 Librería RF24 Mesh

Nodo emisor. Genera datos sucesivos y ordenados, y los envía por la red con una etiqueta ‘T’ cada 5 segundos para una correcta sincronización con la plataforma ThingSpeak. Verifica la llegada de mensajes con etiqueta ‘L’ para realizar el reenvío del mensaje “1” hacia el nodo de cabecera cada que este mensaje sea recibido. Mantiene la red Mesh actualizada y en caso de que el nodo falle, se reconfigura para adaptarse de nuevo a la red, volviéndose a conectar hacia la red, ya sea hacia el mismo nodo al que pertenecía previamente o enlazándose a uno diferente. Esto puede ser apreciado en *Figura 13*.

Figura 13

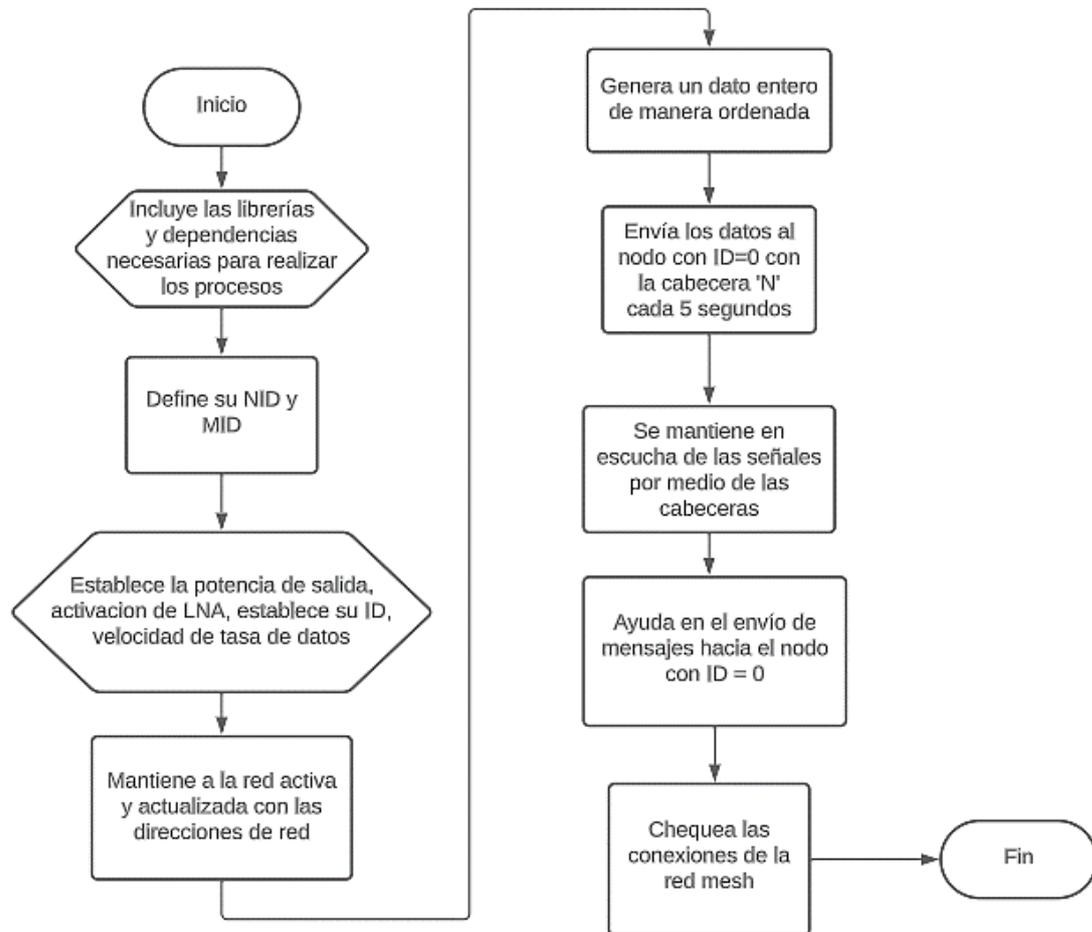
Diagrama de bloques del nodo emisor



Nodo repetidor. Por medio de *Figura 14* se puede observar el diagrama de flujo del nodo. Este nodo recopila los datos de los nodos emisores a los que se encontró enlazados, recibe los mensajes por parte del nodo emisor, ayudando en el envío de la información. Actualiza constantemente el estado de la red y de sus nodos. En caso de existir algún cambio, debe notificarlo en la red. Envía mensajes con valores ordenados y consecutivos con la etiqueta 'N' hacia el nodo de cabecera cada 5 segundos para poder verificar la distancia máxima entre los nodos, lo cual permite la observación del nodo a través de la red, ya que, en caso de no recibir estos datos, se buscaría la reconexión a la red.

Figura 14

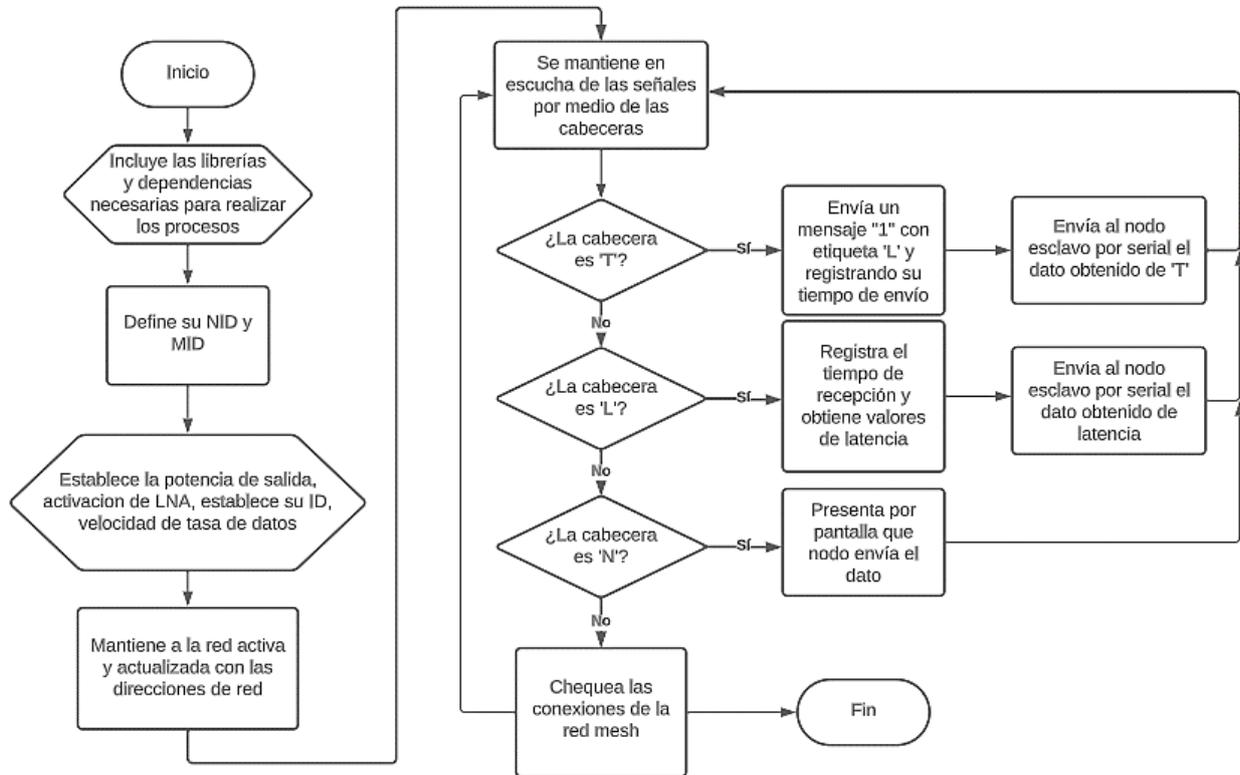
Diagrama de bloques del nodo repetidor



Nodo de cabecera/maestro. Encargado de la recepción de los datos reorientados por parte de los nodos repetidores. Es el coordinador de toda la red tanto a nivel Mesh como a nivel de red, logrando alterar los valores de las NID en caso de ser necesario. Cada que recibe un mensaje con una etiqueta tipo “T”, envía un mensaje de 1 byte representado por el carácter “1”, para realizar el proceso de ping. Tras procesar los datos obtenidos por medio del ping a través de la etiqueta ‘L’ y la recepción de los datos por parte del emisor, reenvía estos datos por medio de transmisión serial hacia el nodo esclavo. Recibe datos con la etiqueta tipo ‘N’ para la presentación por pantalla del nodo que envía el mensaje y su respectivo dato, con el fin de obtener las distancias máximas entre los nodos. Esto se puede visualizar por medio de *Figura 15*

Figura 15

Diagrama de bloques del nodo cabecera/maestro



Nodo esclavo. Recibe todos los datos por parte del nodo maestro a través de la comunicación Serial y lo clasifica por las etiquetas definidas. Se enlaza a una red Wi-Fi y envía los datos a ThingSpeak. Su comportamiento es similar al descrito en *Figura 12*.

2.2 Descripción de escenarios

En esta sección se presentarán los escenarios donde se realizarán las pruebas, y de ser necesario, detallar el caso de estudio en dicho escenario

2.2.1 Escenario 1: Canchas de FIEC - ESPOL

El terreno que fue escogido para obtener datos de las redes fue el espacio de canchas de fútbol de FIEC, ubicado en la Facultad de Electricidad y Computación (FIEC) en el campus Gustavo Galindo, en la ciudad de Guayaquil, Ecuador. Este terreno tiene un área aproximada de $15245.20 m^2$ (1.52 hectáreas), el cual puede ser observado en *Figura 16*. La ubicación posee las

siguientes coordenadas: 2° 8' 33.009"S 79° 58' 3.794" W (-2.142592, -79.967427). La zona presenta completa planicie en su terreno, facilitando la línea de vista. Esto se puede observar por medio de *Figura 17* y *Figura 18*.

Figura 16

Medidas del complejo deportivo de FIEC

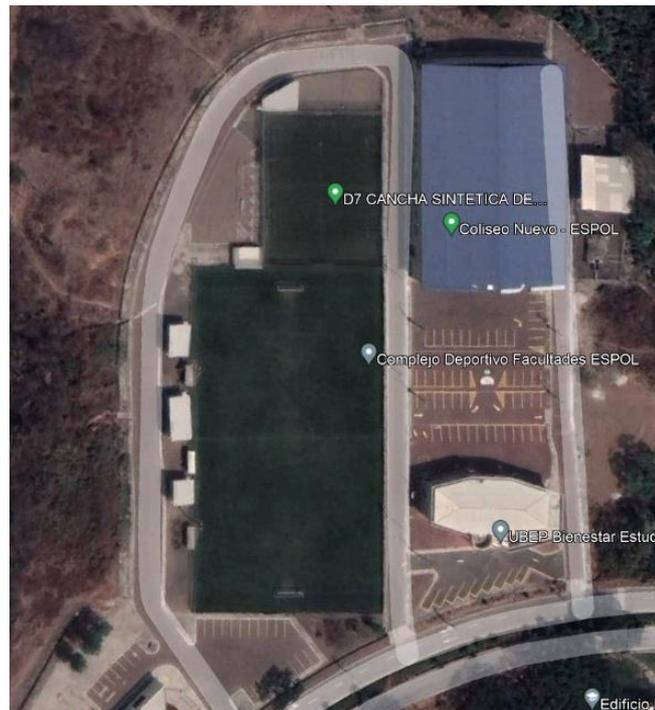


Figura 17

Fotografía diagonal hacia el complejo deportivo



Figura 18. Fotografía lateral del complejo deportivo

Fotografía lateral del complejo deportivo



2.2.2 Escenario 2: Parque Samanes

El terreno que fue escogido para obtener datos de las redes fue el parque Samanes, ubicado en Samanes 6, parroquia Tarqui, ciudad de Guayaquil, Ecuador. Este terreno tiene un área aproximada de 817673.30 m^2 (81.77 hectáreas), el cual puede ser observado en *Figura 19*. La ubicación posee las siguientes coordenadas: $2^\circ 6' 23.993''\text{S } 79^\circ 53' 56.43'' \text{ W}$ (-2.103964, -79.899222). Este fue escogido debido a la facilidad de su terreno, ya que es completamente plano y cuenta con extensos caminos en los que se puede conseguir línea de vista; sin embargo, la gran presencia de personas, dispositivos que trabajan en la frecuencia de 2.4 GHz, autos que se encuentran en el recinto, pueden presentar posibles interferencias en las mediciones. La zona posee un entorno híbrido, donde en algunas secciones presentarán regiones libres de obstáculos y en otras se presentarán con presencia de árboles y edificaciones. Esto puede ser observado por medio de *Figura 20* y *Figura 21*.

Figura 19

Fotografía satelital del parque Samanes



Figura 20

Fotografía 1 de zona prueba en Parque Samanes



Figura 21

Fotografía 2 de zona de prueba



2.2.3 Escenario 3: Lago de ESPOL

El terreno que fue escogido para el desarrollo de este escenario fue el lago de ESPOL, ubicado en la Facultad de Arte, Diseño y Comunicación Social (FADCOM) en el campus Gustavo Galindo, en la ciudad de Guayaquil, Ecuador. Este cuerpo marino tiene un área aproximada de 44.000 m^2 (4.4 hectáreas), apreciable por medio de *Figura 22*. La ubicación posee las siguientes coordenadas: $2^{\circ}8'42.454''\text{S } 79^{\circ}57'45.212''\text{ W}$ (-2.145126, -79.962559). Dentro de esta localidad se debe considerar la gran presencia de varias especies de árboles, así como la presencia del cuerpo marino, lo cual puede causar un decaimiento en la calidad de la red, viéndose reflejado en la cobertura de los nodos y los valores de latencia, como se puede observar en *Figura 23*. Dentro de este escenario se presentan dos casos de estudio para ambas topologías. Para *painlessMesh*, su caso 1 corresponde al análisis de la red buscando completa linealidad en el perímetro, mientras que el caso 2 hace estudio del rendimiento de la red en una de las curvas correspondientes al lago. En *RF24 Mesh*, el estudio del caso 1 se centrará en el análisis de la red realizando una transmisión

sobre el agua y buscando la mayor linealidad en el perímetro del cuerpo marítimo; finalmente, el caso 2 de RF24 Mesh corresponderá a un análisis en el que únicamente se transmiten datos sobre el agua.

Figura 22

Fotografía satelital del lago ESPOL

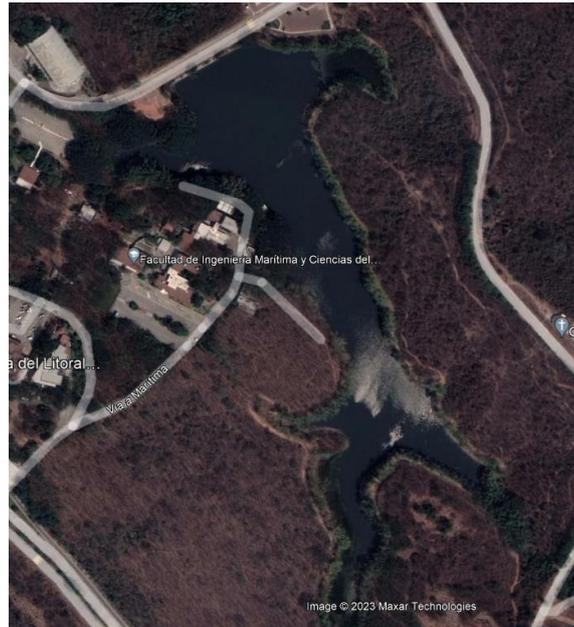
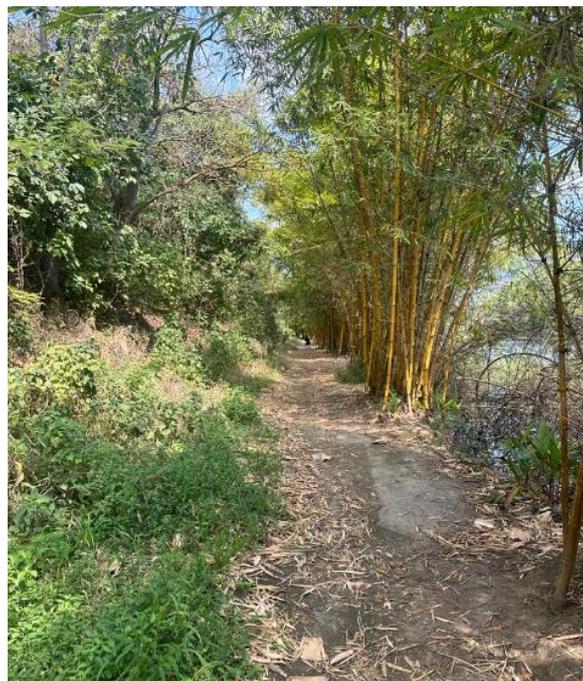


Figura 23

Fotografía del perímetro del lago ESPOL



2.3 Diseño de la red Mesh

En esta sección se darán detalles sobre la topología a analizar, los elementos usados para la realización de las pruebas e implementación de las redes Mesh tanto a nivel de hardware como de software

2.3.1 Topología

A continuación, se presentarán 3 topologías: la de posible implementación usando `painlessMesh` y RF24 Mesh en un escenario normal y la topología en la que se basará el estudio

Librería `painlessMesh`. La topología propuesta fue una red parcialmente Mesh, lo cual permite una mejor manipulación de la red para poder ser adaptada a otros tipos de topologías, brindando escalabilidad y una mayor confiabilidad en la red. La topología tipo anillo, la cual es una red parcialmente Mesh, fue descrita en *Wireless Mesh Networks in IoT networks* [30] siendo propuesta como una red híbrida para ser implementada con enrutadores. El propósito de dicha topología es que, en caso de que un nodo se encuentre caído, se puede hacer uso de otra ruta para enviar el paquete de datos de la medición obtenida hacia el nodo de cabecera, permitiendo la adaptación de otros nodos dentro del anillo y logrando que cada nodo del anillo se convierta en el nexo entre otros dispositivos que no puedan enlazarse a esta estructura, manejando una topología propia (ya sea una topología de árbol, anillo, malla, entre otros) en esta subred. En *Figura 24* se puede observar la topología anillo y su posible implementación en *Figura 25*.

La topología busca cubrir la mayor cantidad de distancia posible de un perímetro que logra ocupar un área aproximada de 3 hectáreas, sin el uso de algún módulo o modificación al ESP32. Sin embargo, para el análisis se consideró un caso crítico, donde el nodo final no se enlaza directamente con el nodo maestro, tomando similitud con una topología tipo árbol, donde cada nodo será el hijo de otro, tal como se presenta en *Figura 28* y *Figura 29* como su posible posicionamiento, esta topología será la implementada para el Capítulo 3.

La calidad de la red se basará en la medición de la latencia de la red Mesh, la cual será medida por medio del método ping-pong. Este método consiste en que el nodo de cabecera envía un mensaje de un solo carácter (“1”) hacia el nodo emisor, registrando el tiempo de envío de datos. El nodo emisor recibe el dato y lo reenvía hacia el nodo de cabecera. Al momento de recibir el mensaje, se registra el tiempo de recepción.

Figura 24

Topología de condición ideal usando painlessMesh

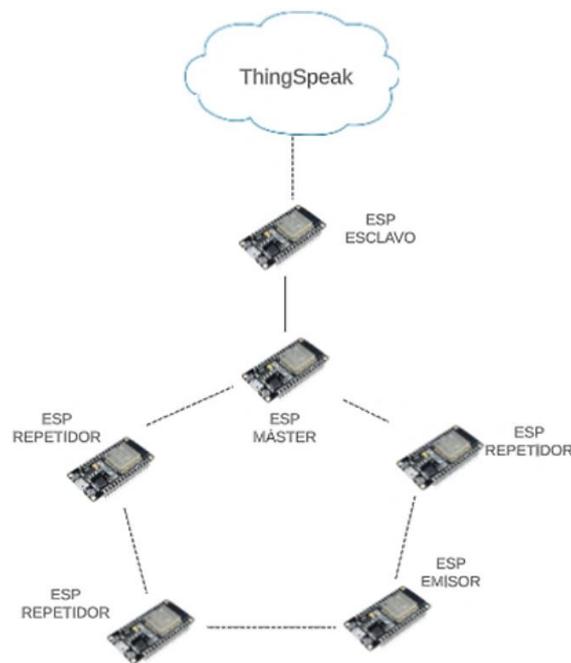


Figura 25

Posible posicionamiento de nodos en condiciones ideales con painlessMesh



Librería RF24 Mesh. La topología propuesta fue basada en los lineamientos de la librería presentada en su documentación [28], donde solamente se pudo aplicar una topología tipo árbol, presentado en *Figura 26* y *Figura 27* como las posiciones dentro de la piscina para su implementación. De igual manera, se escogió el escenario donde se encontrará un nodo en cada nivel de red proporcionado por RF24 Mesh, el cual toma similitud con una topología lineal y está representada en *Figura 28* y *Figura 29* para su posterior implementación y análisis en el Capítulo 3. La calidad de la red se basará en la medición de la latencia de la red Mesh, la cual será medida por medio del método ping-pong. En esta red se enviarán mensajes tipo ping hacia el nodo emisor, registrando los tiempos de envío y de recepción del mensaje “1”. Para el posterior análisis se consideró el caso donde una ramificación completa de la red, donde el nodo final se encuentra en el último nivel posible de la red.

Figura 26

Topología de condición ideal usando RF24 Mesh

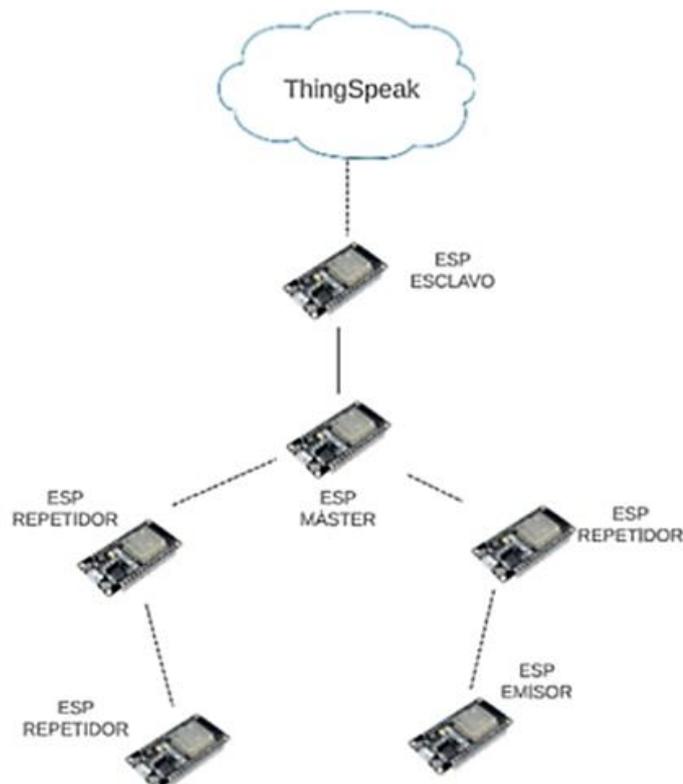


Figura 27

Posible posicionamiento de nodos en condiciones ideales con RF24 Mesh



Figura 28

Topología propuesta para el análisis de las redes

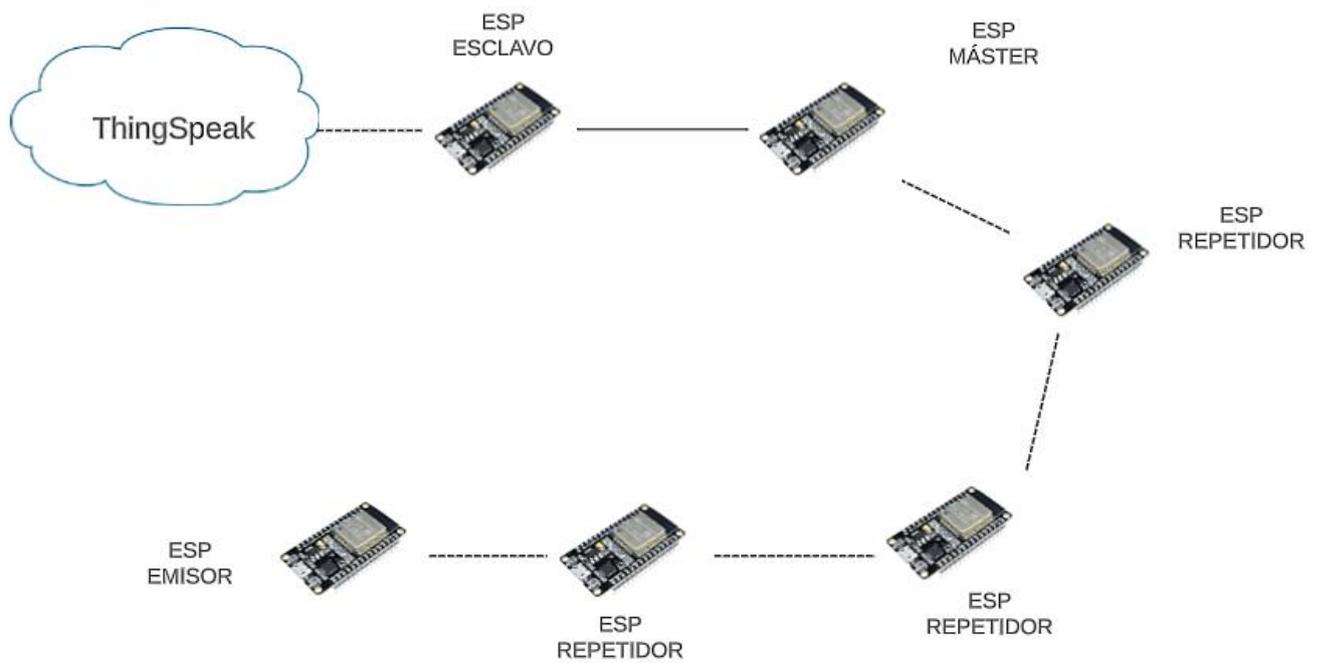


Figura 29

Posible posicionamiento de nodos en los casos críticos



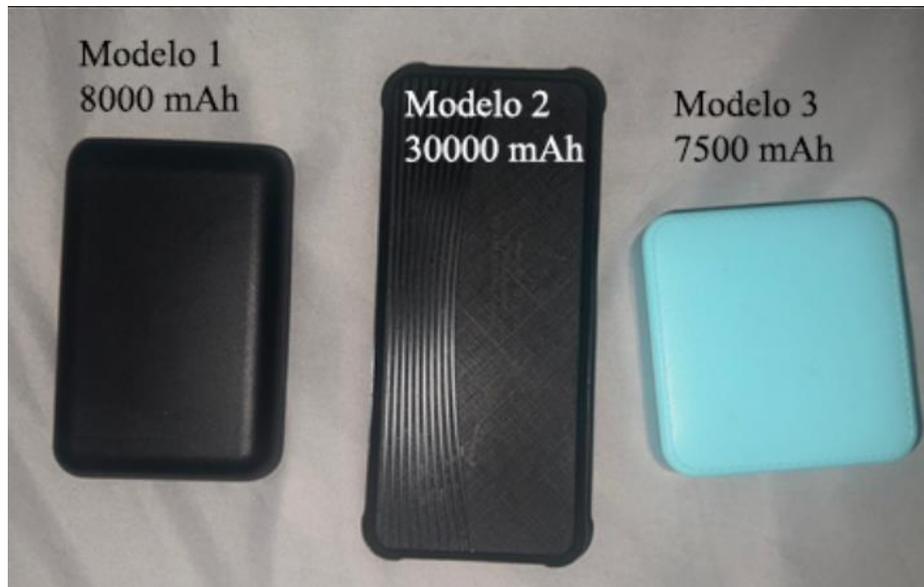
2.3.2 Equipos de red

El conjunto de equipos usados fueron 4 ESP32 WROOM 32D y 2 ESP32 WROOM 32, ambos grupos siendo parte de la familia de los ESP32 Device Kit versión 4 (ESP32_Devkitc_V4). Los ESP32 WROOM 32 fueron usados para la implementación de los nodos emisores y esclavo, mientras que en los ESP32 WROOM 32D se activaron los nodos de cabecera y repetidores.

Para la alimentación de todos los nodos de la red se hizo uso de 3 distintos bancos de energía: 1 banco de 8000mAh con salida de 5V a un máximo de 2.4 A, 1 banco de 30000 mAh con una salida máxima de 5V a 2 A y 2 bancos de 7500 mAh con una salida máxima de 5V a 3 A; estos modelos pueden ser revisados por medio de *Figura 30*. Estos bancos dieron alimentación a 4 de los 6 nodos implementados por medio de cables USB-micro USB tipo B. Los nodos de cabecera-maestro y esclavo fueron alimentados por la conexión USB de la laptop usada para la verificación de los datos y la emisión de la señal Wi-Fi.

Figura 30

Modelos de baterías usadas



2.3.3 Software

ThingSpeak. ThingSpeak es una plataforma de análisis de IoT que permite agregar, visualizar y analizar diversos flujos de datos en vivo en la nube. Se puede enviar datos a ThingSpeak desde diversos dispositivos, creando alertas y visualización instantánea de datos en tiempo real por medio de la plataforma o de su aplicación móvil. Se puede programar la plataforma para poder tener reacciones dependiendo de los datos obtenidos, así como de su análisis y visualización de los datos en Matlab. [35]

Arduino IDE. Arduino IDE es un programa que permite el desarrollo, depuración y escritura de un programa, llamado sketch dentro del programa. Su programación se basa en el lenguaje C++. A su vez, permite la descarga e implementación de diversas librerías para la el uso de diversas funciones.

2.3.4 Librerías

Las librerías a usadas en este proyecto para la implementación de las redes Mesh fueron painlessMesh y RF24 Mesh. En el caso de RF24 Mesh, se le tuvo que agregar las librerías

requeridas como RF24 y RF24 Network. Para `painlessMesh` se tuvo que descargar las dependencias `ArduinoJson`, `TaskScheduler` y `AsyncTCP`, las cuales son requisitos para el correcto funcionamiento de este; sin embargo, `TaskScheduler` tiene una orientación hacia microcontroladores Arduino y no ESP32, por lo que se realizó un contador para poder realizar las tareas cada cierto periodo de tiempo establecido.

Para la comunicación entre los ESP32 maestro y esclavo se hizo uso de la librería `HardwareSerial`, para la transferencia de datos por los pines de estos dispositivos. Adicional, el ESP32-esclavo contó con las librerías `WiFi` y `ThingSpeak` para el enlace de datos en la plataforma para el posterior análisis de datos.

2.4 Implementación

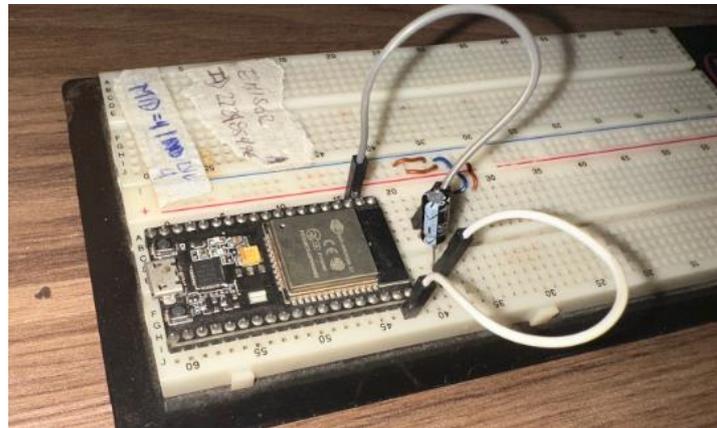
En esta sección se detallan todas las conexiones de los microcontroladores con sus módulos y otros elementos que son necesarios para un correcto funcionamiento de la red. Se debe aclarar que todos estos nodos se encontrarán conectados a alguno de los modelos de baterías descritos en la sección 2.3.2 de este documento y representados por medio de *Figura 24*.

2.4.1 Librería `painlessMesh`

Para el nodo emisor se realizó su conexión de un capacitor de 10 μ F a 50V entre los pines de EN (Enable) y GND (Ground) con dos propósitos: estabilizar la conexión a una red Wi-Fi a largo plazo y evitar mantener aplastada la botonera de Boot al momento de subir el código a este microcontrolador. Este ajuste debe solo realizarse a las versiones V1.1 de los ESP32, que, en este caso, debe ser aplicado a dos nodos presentes en la red. Estas conexiones pueden ser observadas en *Figura 31*.

Figura 31

Implementación de nodos emisores



Los nodos intermedios no necesitan tener una configuración especial o adicional, ya que su programación está orientada hacia la retransmisión de datos y no se encuentran sujetos a la interacción con sensores o módulos para su correcto funcionamiento. Su implementación puede ser visualizada por medio de *Figura 32*.

Figura 32

Implementación de nodos repetidores

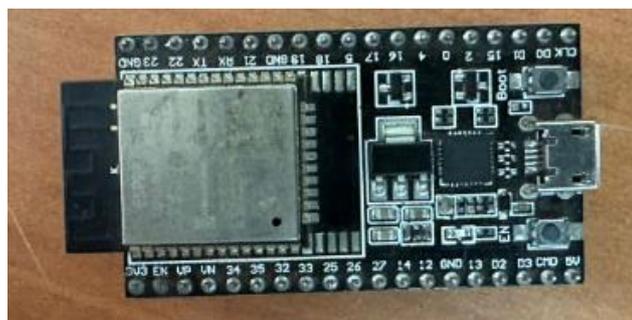


Figura 33 nos muestra que el nodo de cabecera/maestro y esclavo únicamente deben estar conectados en sus pines 16 y 17, ya que se encuentran sus transmisores y receptores para realizar la transmisión de modo maestro-esclavo a través de la comunicación serial, sin requerir conexiones adicionales o configuraciones especiales, exceptuando lo especificado en su programación, el cual solo consiste en realizar la conexión a Internet para poder subir los datos a la plataforma ThingSpeak..

Figura 33

Implementación de nodos cabecera/maestro y esclavo

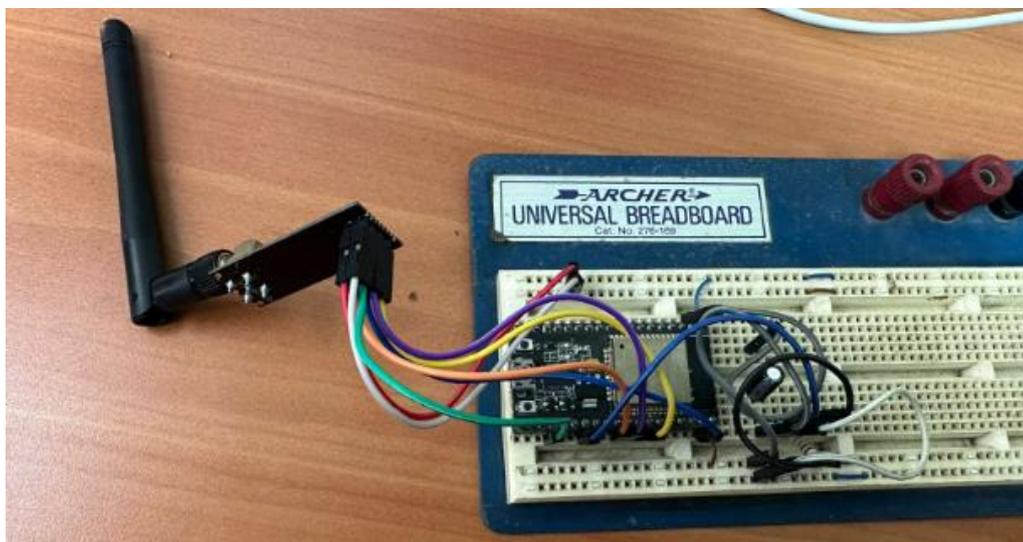


2.4.2 Librería RF24 Mesh

Para el nodo emisor se realizó su respectiva implementación en un protoboard. Para la conexión con el módulo nRF24L01+PA+SMA se hicieron uso de los pines 15 y 5 para los puertos CE y CNS respectivamente, GND conectado al pin del ESP32 con el mismo nombre, VCC al terminal de 3.3V, el pin 19 para el pin MISO, pin 23 para el pin MOSI y el pin 18 para el terminal SCK. De manera similar, se encuentra un capacitor de $10\ \mu\text{F}$ a 50 V para la carga rápida del código sin necesidad de mantener presionado el pulsador de Boot. Su representación se encuentra en *Figura 34*.

Figura 34

Implementación de nodos emisores



Los nodos intermedios solo necesitan la conexión del módulo nRF24L01+PA+LNA, por lo que no necesitan alguna otra configuración adicional, y ya que su programación está orientada hacia la retransmisión de datos, no se necesita conectarla con algún sensor. Sus conexiones se pueden visualizar en *Figura 35*.

Figura 35

Esquemático del nodo intermedio usando RF24 Mesh

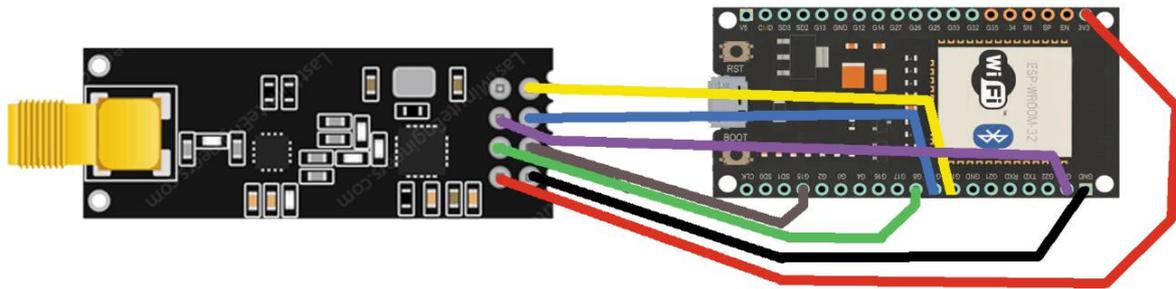
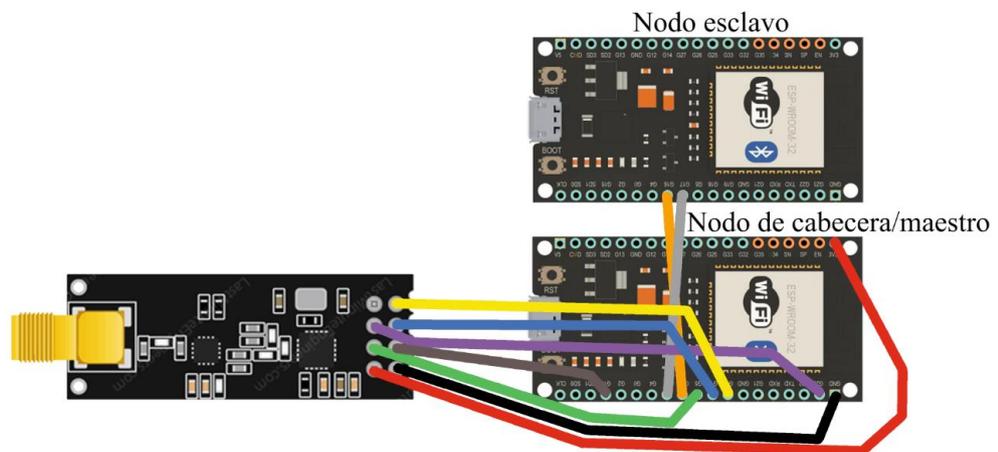


Figura 36 nos muestra que el nodo de cabecera y esclavo únicamente deben estar conectados en sus pines 16 y 17, ya que se encuentran sus transmisores y receptores para realizar la transmisión de modo maestro-esclavo por la comunicación serial. El nodo de cabecera/cabecera es el único de los dos nodos que se encontrará conectado con el módulo nRF24L01+PA+LNA.

Figura 36

Esquemático del nodo cabecera/maestro y esclavo usando RF24 Mesh



Capítulo 3

3.1 Resultados y análisis

En esta sección se realizará el análisis de los resultados obtenidos por medio de la plataforma ThingSpeak y el posicionamiento en base a coordenadas almacenadas en Google Maps respecto a la ubicación de los nodos, y determinar de manera más precisa la distancia estimada entre nodos y la posible área de cobertura.

3.2.1 Pruebas de enlace

Dentro de esta sección se realizarán las pruebas de conexión e intercambio de datos entre los distintos nodos y la escritura de los datos en la plataforma ThingSpeak para un posterior análisis.

painlessMesh. Para la prueba de enlace con painlessMesh se hizo uso de dos nodos para establecer una distancia máxima entre ambos mostrada en *Figura 37*, así como la correcta escritura de datos obtenidos en el nodo maestro y enviado hacia el esclavo visible en *Figura 38*, con su publicación en la plataforma ThingSpeak, presentada en *Figura 39*.

Figura 37

Ubicación de los nodos para la prueba de enlace

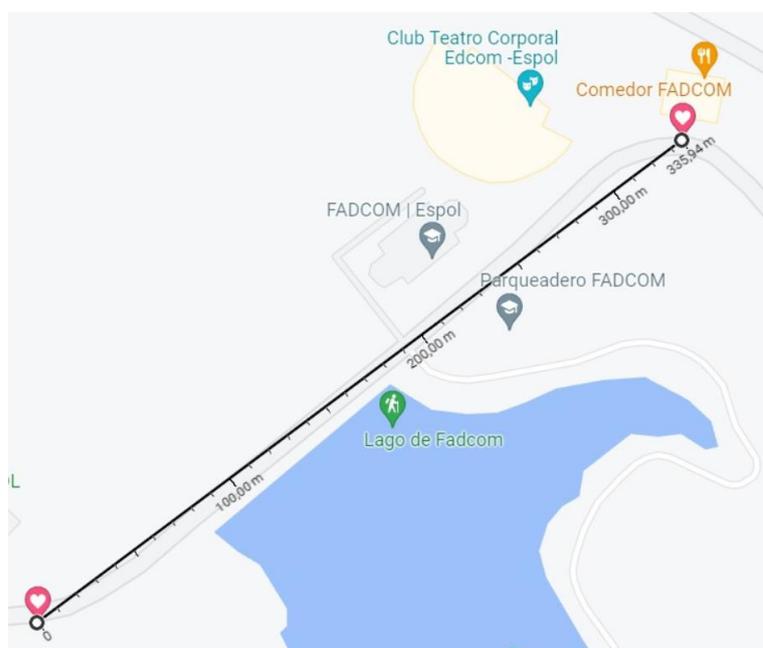


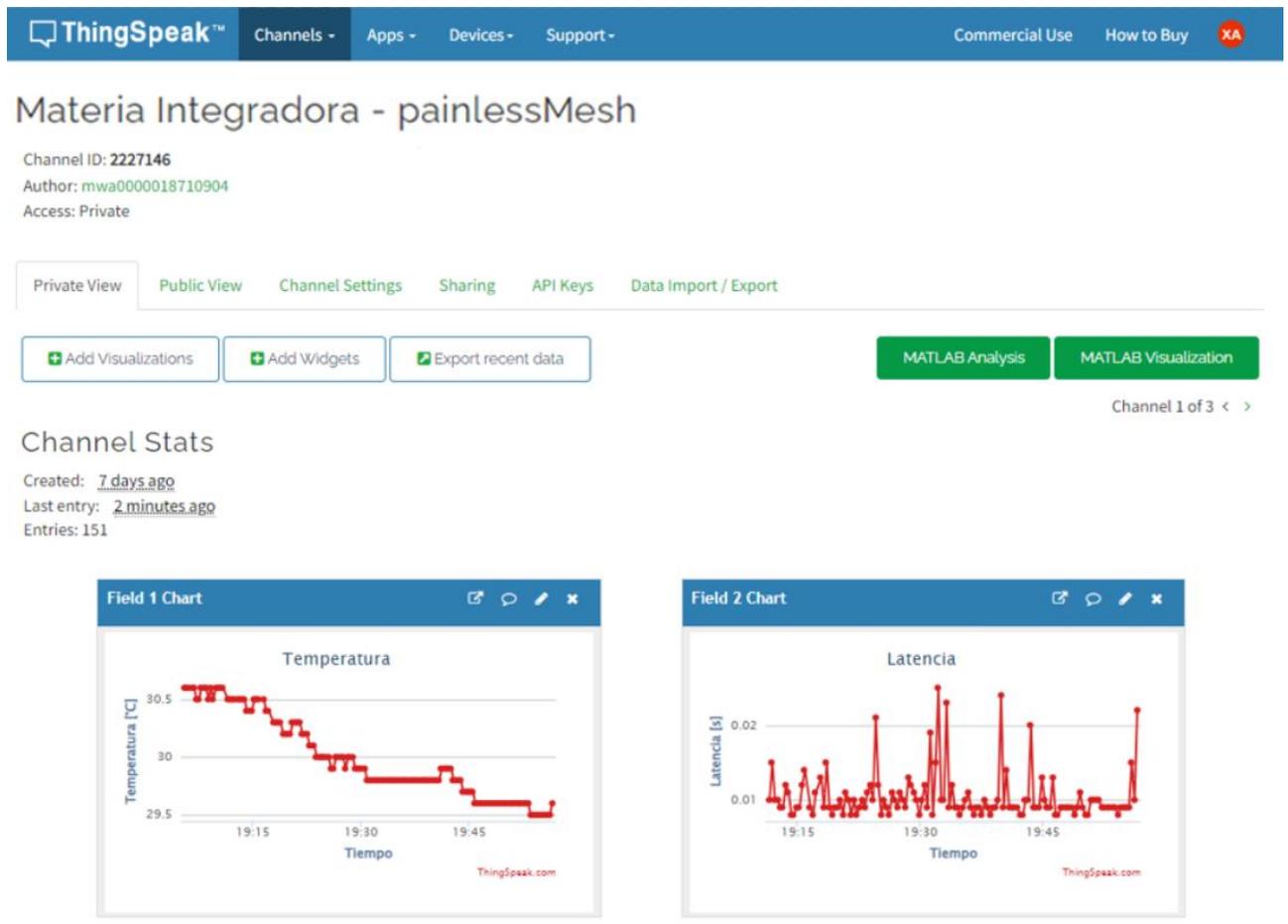
Figura 38

Recepción y publicación de datos en el nodo esclavo

```
Conectando a la red Wi-Fi...
Conectando a la red Wi-Fi...
Conexión Wi-Fi establecida.
Received temperature: 30.30
Received latency: 0.013 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.30
Received latency: 0.009 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.30
Received latency: 0.015 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.30
Received latency: 0.009 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.20
Received latency: 0.008 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.20
Received latency: 0.009 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.20
Received latency: 0.009 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.20
Received latency: 0.010 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.30
Received latency: 0.008 segundos
Dato publicado en ThingSpeak.
Received temperature: 30.30
Received latency: 0.011 segundos
```

Figura 39

Publicación de datos en ThingSpeak



RF24 Mesh. De manera similar a *painlessMesh*, se instalaron las dependencias necesarias para poder establecer conexión usando RF24 Mesh y 4 nodos para la verificación de los multiniveles presentes en la red con su recepción de datos, como se puede observar en *Figura 40*, donde de manera visible se puede observar el cambio que existe entre el NID de los diversos nodos sin que su MID sea alterado; esto con el propósito de poder optimizar la red en base a niveles de potencia que se reciben en referencia al nodo de cabecera de la red Mesh. En *Figura 41* se puede evidenciar la correcta publicación de los datos obtenidos dentro de ThingSpeak.

Figura 40

Pruebas de enlace – Recepción y publicación de datos en el nodo esclavo

```

*****Assigned Addresses*****
NodeID: 1 RF24Network Address: 05
NodeID: 2 RF24Network Address: 045
NodeID: 5 RF24Network Address: 0445
*****
*****Assigned Addresses*****
NodeID: 1 RF24Network Address: 05
NodeID: 2 RF24Network Address: 04
NodeID: 5 RF24Network Address: 0445
*****
*****Assigned Addresses*****
NodeID: 1 RF24Network Address: 05
NodeID: 2 RF24Network Address: 045
NodeID: 5 RF24Network Address: 0445
*****
*****Assigned Addresses*****
NodeID: 1 RF24Network Address: 05
NodeID: 2 RF24Network Address: 045
NodeID: 5 RF24Network Address: 0445
*****
Temperatura:28.40
Latencia: 19

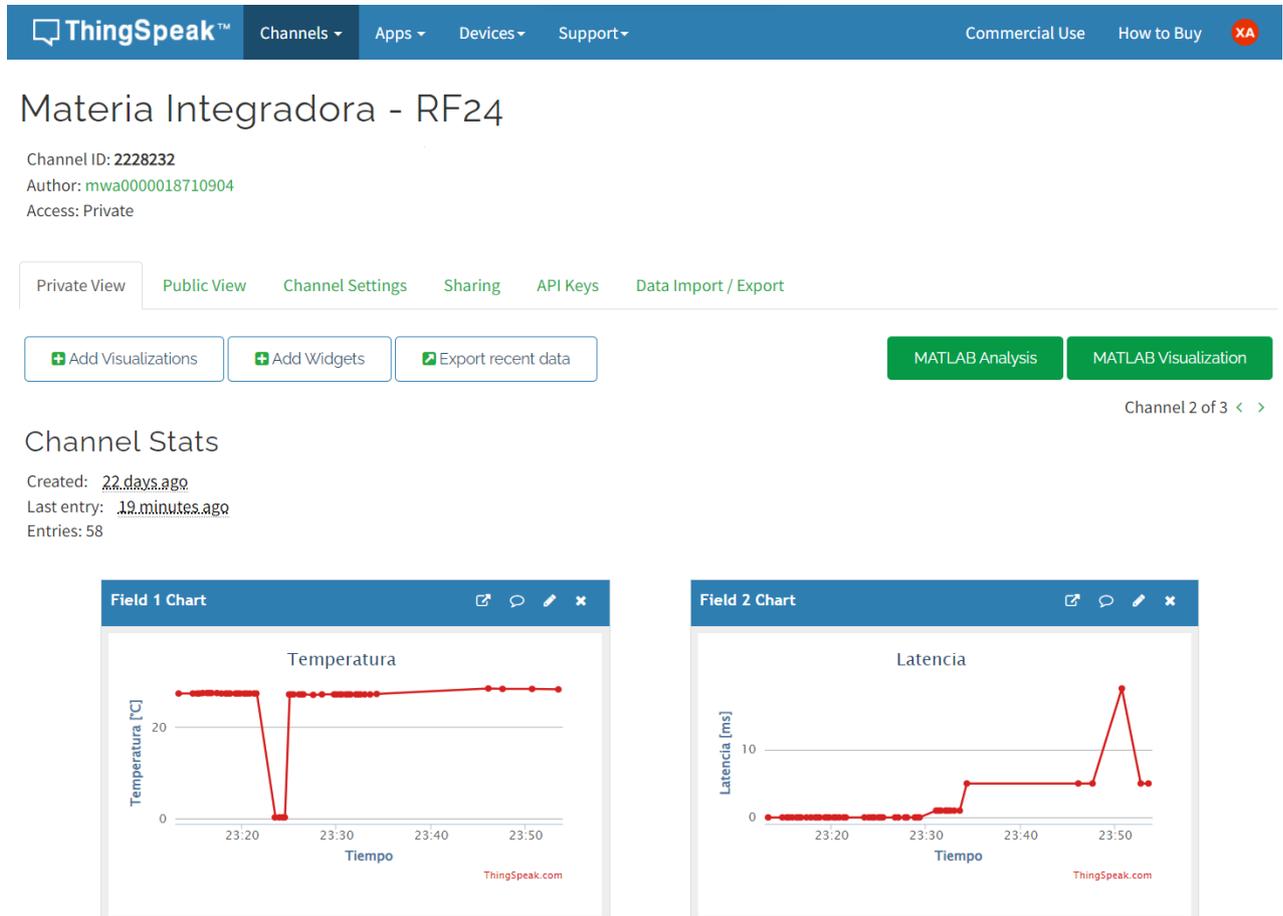
*****Assigned Addresses*****
NodeID: 2 RF24Network Address: 04
NodeID: 1 RF24Network Address: 05
NodeID: 5 RF24Network Address: 044
*****
Temperatura:28.30
Latencia: 5

*****Assigned Addresses*****
NodeID: 2 RF24Network Address: 05
NodeID: 1 RF24Network Address: 04
NodeID: 5 RF24Network Address: 044
*****
*****Assigned Addresses*****
NodeID: 2 RF24Network Address: 04
NodeID: 1 RF24Network Address: 05
NodeID: 5 RF24Network Address: 044
*****
*****Assigned Addresses*****
NodeID: 2 RF24Network Address: 05
NodeID: 1 RF24Network Address: 04
NodeID: 5 RF24Network Address: 044
*****

```

Figura 41

Pruebas de enlace – Publicación de datos en ThingSpeak



3.2.2 Pruebas en Complejo deportivo de ESPOL

painlessMesh. En este escenario, la red presenta datos de latencia de 2 ms y 10.445 s como valores mínimos y máximos de latencia, con un promedio de 516.69 ms. Los valores de latencia se encuentran en su mayoría entre 8 ms y 24 ms. Los picos que se pueden observar en *Figura 43*, están relacionados con la implementación de los nuevos nodos repetidores que se iban agregando poco a poco en la red. Al momento que estos nodos se incorporan a la red, los nodos se reorganizan para optimizar la red y buscar asegurar la retransmisión del dato originado por el nodo transmisor.

La red tuvo una distancia total de alrededor de 145.52 m a lo largo del perímetro del complejo deportivo, tomando como máximos y mínimos de distancia los valores de 43.02 m y 30.71 m respectivamente, esto puede ser visualizado en *Figura 42* por medio de la toma de las distancias por medio de Google Earth.

La poca cobertura se debe a la presencia de 2 señales Wi-Fi (ESPOL y eduroam) dentro del complejo, así como cables de alta tensión cerca de los alrededores del complejo deportivo, lo cual produce interferencias con la red Wi-Fi Mesh. Por lo que esta red se encuentra a un 18.51% de poder llegar a la distancia planteada de 786.13 m aproximadamente

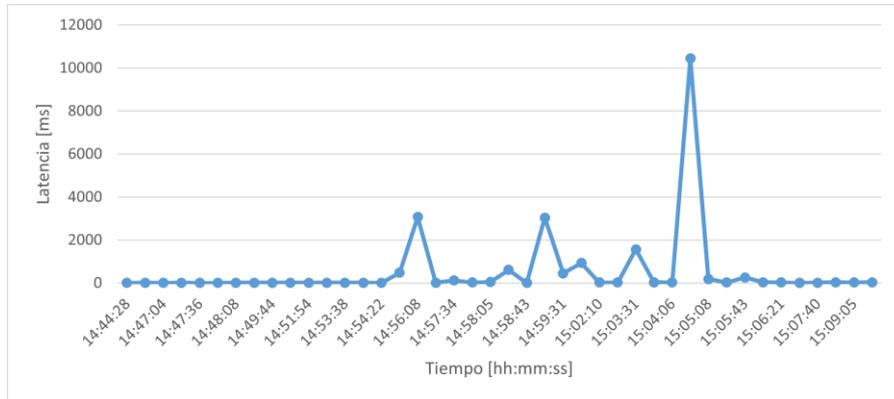
Figura 42

Ubicación de los nodos painlessMesh en el complejo deportivo de ESPOL



Figura 43

Datos de latencia obtenidos en escenario 1 – painlessMesh



RF24 Mesh. Los valores obtenidos en el complejo deportivo arrojaron resultados con valores de latencia de 72 ms y 131 ms, como valores mínimos y máximos respectivamente, con un promedio de 102.89 ms, su gráfica se puede ver en *Figura 44*. Esta red logró obtener una comunicación lineal de 441.86 m, teniendo una distancia máxima y mínima de 203.97 y 52.24 m de distancia entre nodos; por lo que esta red, cumple con el 56.21% de la meta propuesta. La ubicación de los nodos se puede visualizar en *Figura 45*. Se debe tener en consideración, así como en el uso de painlessMesh, la presencia de una torre de alta tensión que produjo interferencia hacia la red y la presencia de las redes Wi-Fi (ESPOL y eduroam)

Figura 44

Datos de latencia obtenidos de Escenario 1 – RF24

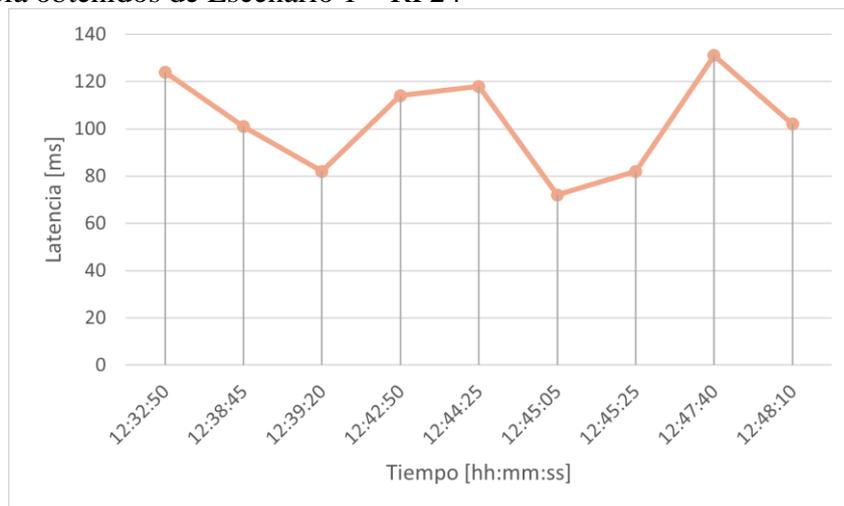
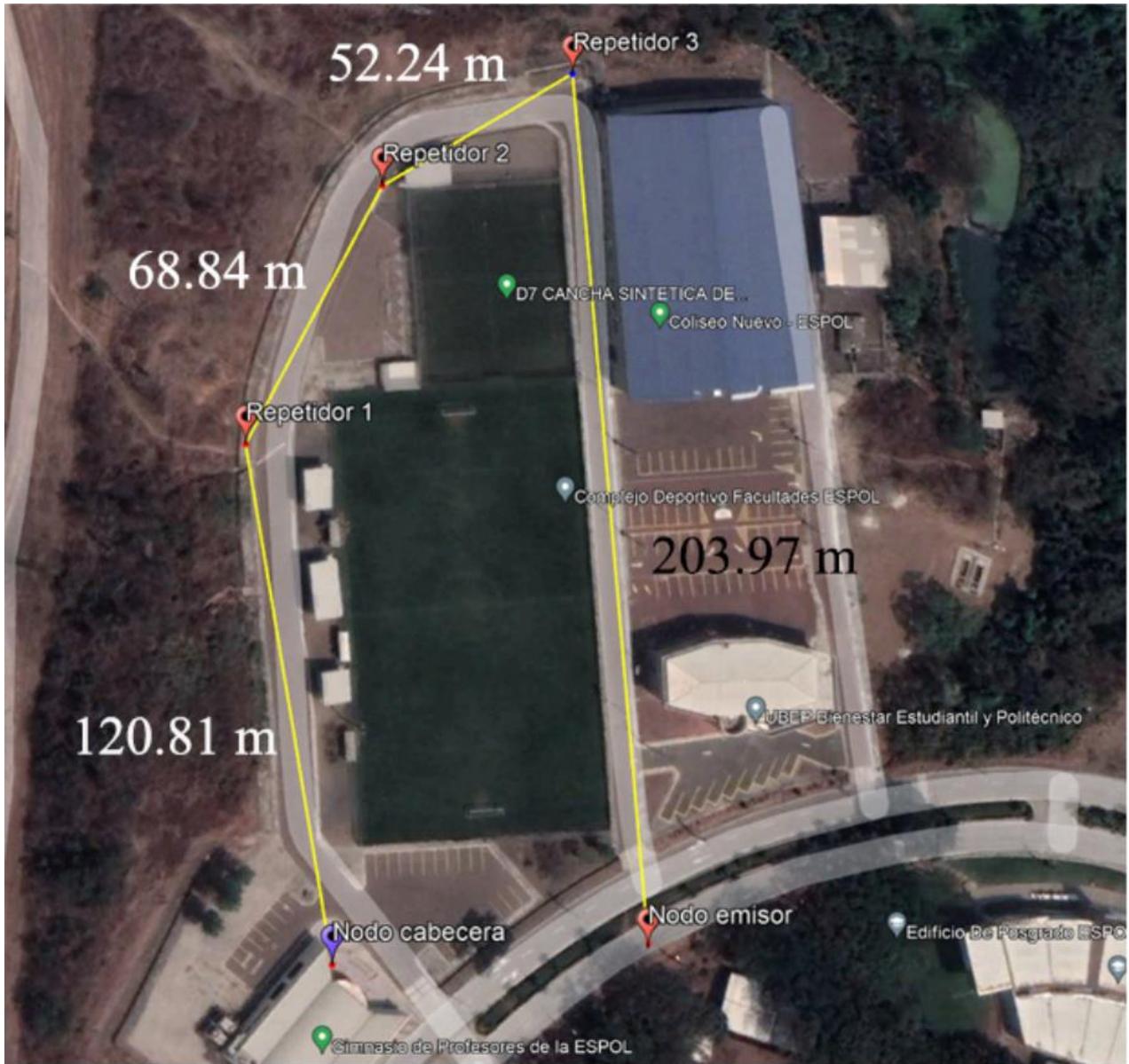


Figura 45

Ubicación de los nodos RF24 en el complejo deportivo



3.2.3 Pruebas en Parque Samanes

painlessMesh. Los valores considerados dentro de la prueba en el parque Samanes son aquellos que se encuentran desde las 15:20 horas debido a pruebas previas de conectividad de los nodos. Estos datos arrojaron un valor de latencia promedio de 906.51 ms, en conjunto con valores mínimos y máximos de 294 ms y 10.54 segundos, respectivamente. A medida que la cantidad de nodos aumentaba, la latencia promedio iba en aumento, esto puede ser observado en *Figura 47*.

Situaciones especiales se llegaron a dar al momento de alcanzar las distancias máximas de transmisión del nodo emisor, ya que en esos instantes se presentaron los picos de latencia en la plataforma. Esta red llegó a tener una distancia de 237.85m con una distancia máxima de 110.07 m en uno de los nodos y una mínima de 19.52 m. Esta red cumple únicamente con el 30.96% de éxito respecto a la meta. La ubicación de los nodos puede ser evidenciado en el *Figura 46*.

Figura 46

Ubicación de los nodos painlessMesh en el parque Samanes

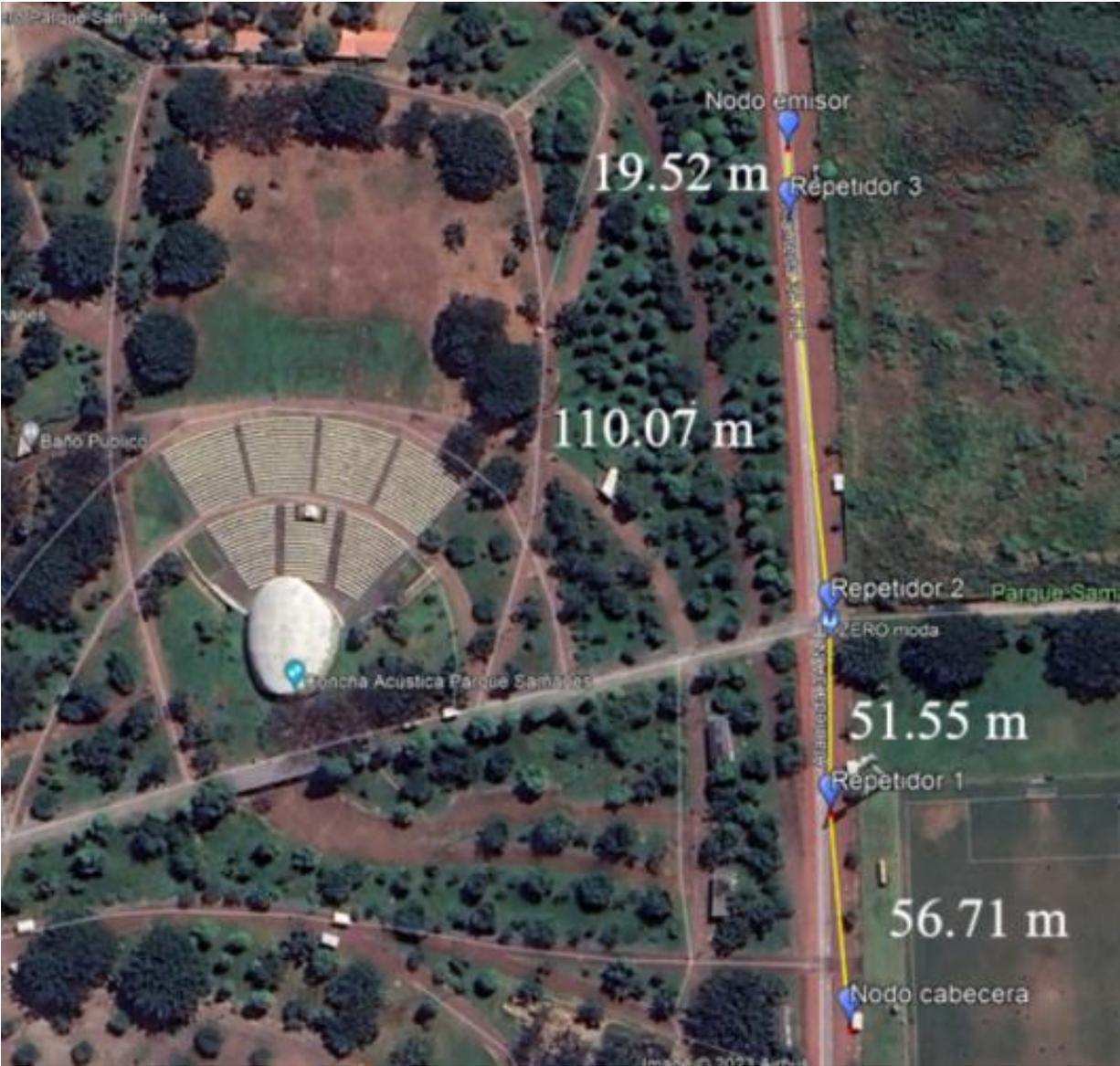
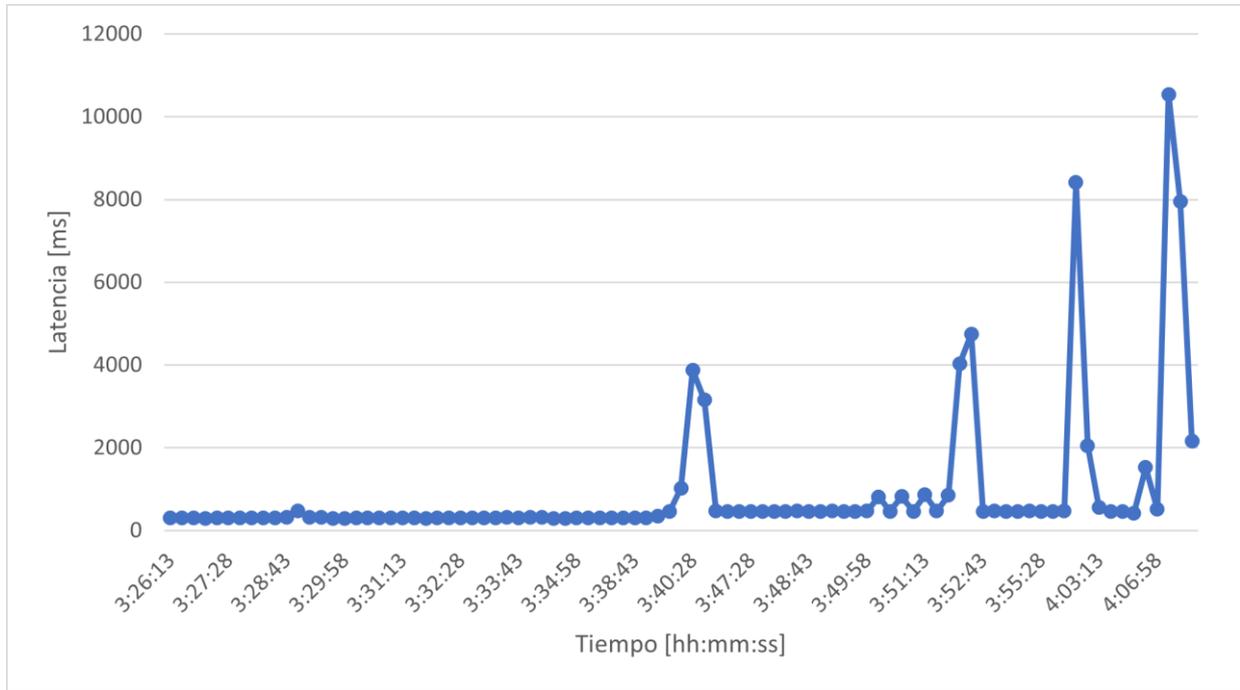


Figura 47

Datos de latencia obtenidos en escenario 2 – painlessMesh



RF24 Mesh. Los valores obtenidos en Parque Samanes por medio de la plataforma ThingSpeak, como se observar en *Figura 48*, arrojan resultados con valores de latencia de 210 ms y 310 ms como valores mínimos y máximos respectivamente, con un promedio de 227.85 ms. En su mayoría, la latencia se mantiene alrededor de valores entre 220 y 250 ms de latencia en la red. Esta red logró obtener una cobertura de 843.19 m, con una distancia máxima y mínima de 261.74 y 110.26 m respectivamente, obteniendo un 107.26% con respecto a la distancia de 786.13 m, logrando cumplir con el objetivo. La ubicación de los nodos se puede observar en *Figura 49*. Se debe tener en cuenta que varias interferencias fueron producidas por agentes externos tales como el paso de vehículos pesados, la presencia de interferencia por las redes Wi-Fi propias del parque Samanes y de los usuarios de esta área pública con el uso de redes Bluetooth y la presencia de árboles.

Figura 48

Datos de latencia obtenidos en escenario 2 – RF24

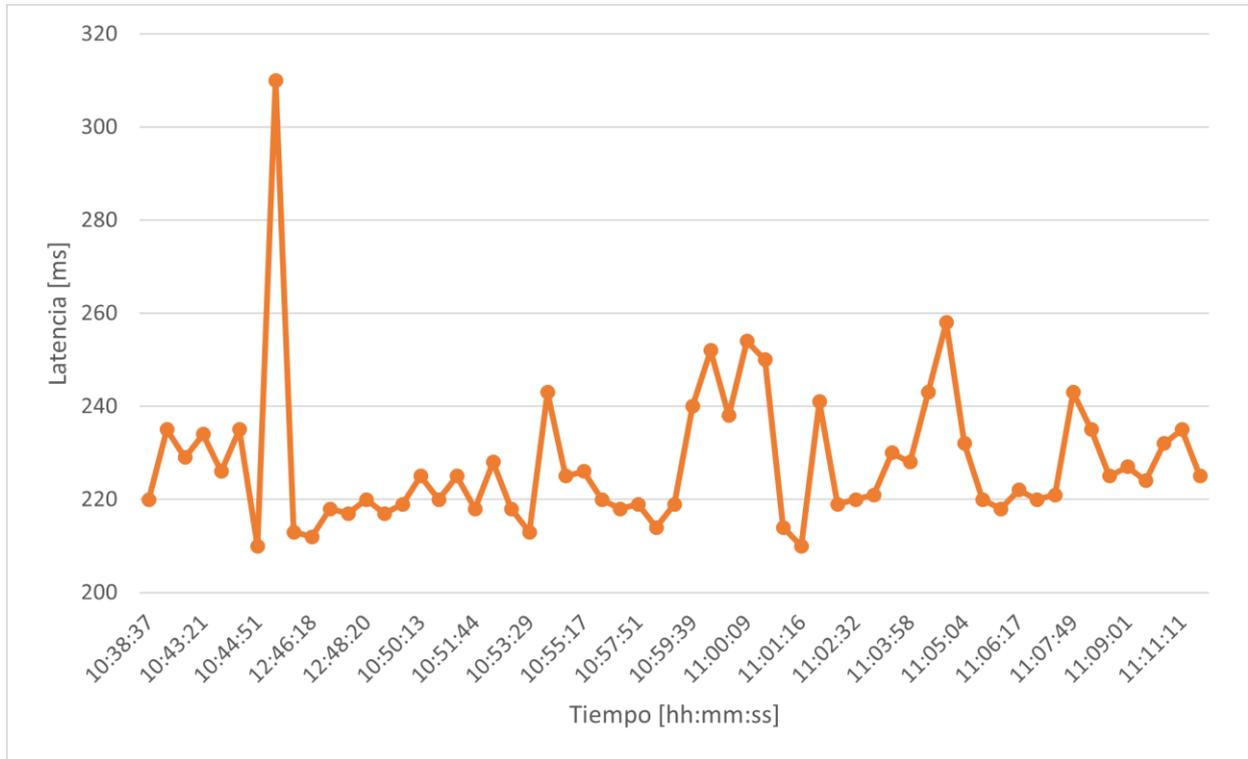


Figura 49

Ubicación de los nodos RF24 en Parque Samanes



3.2.4 Pruebas en Lago ESPOL

painlessMesh.

Caso 1. Los valores considerados dentro de la prueba alrededor del lago de ESPOL son aquellos que se encuentran desde las 14:26 horas debido a fallos de conexión, ya que se buscaba la transmisión de al menos uno de los nodos sobre el agua, pero esto no fue posible debido al poco alcance presentado, lo cual obligó a el reposicionamiento de los nodos dentro del área. Estos datos arrojaron un valor de latencia promedio de 461.52 ms, en conjunto con valores mínimos y máximos de 101 ms y 5.5 segundos respectivamente.

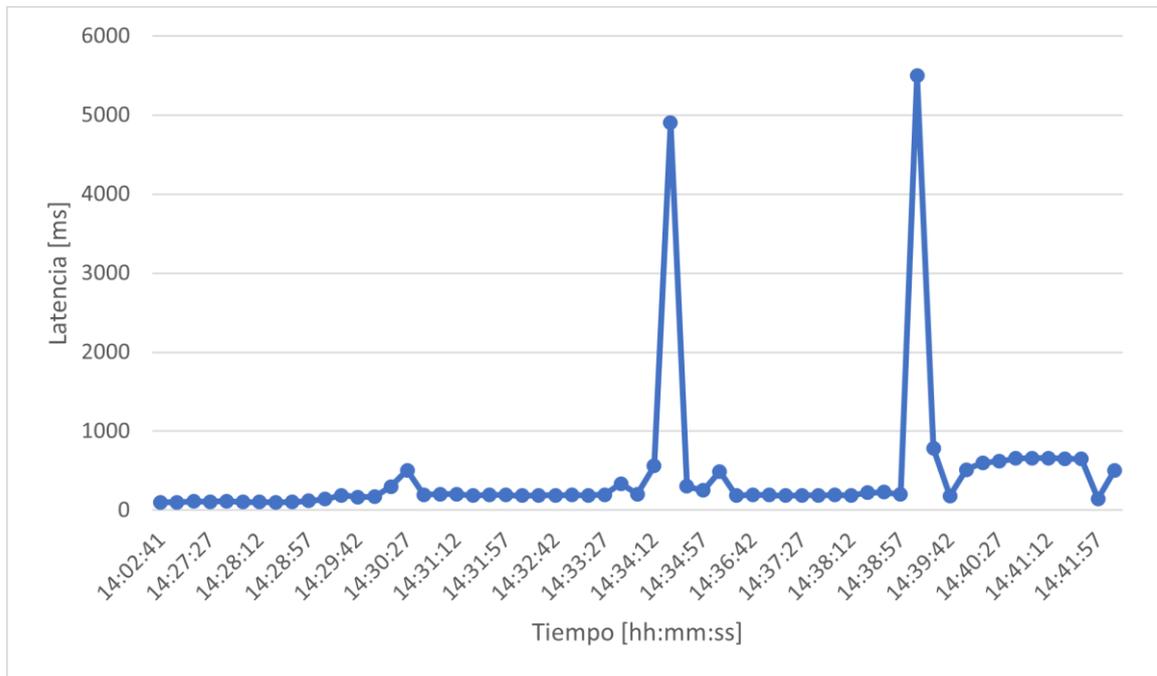
A medida que la cantidad de nodos aumentaba, la latencia promedio iba en aumento, esto puede ser observado en *Figura 50*. Situaciones especiales se llegaron a dar al momento de alcanzar las distancias máximas de transmisión del nodo emisor, en la locación donde se mantiene el nodo repetidor para poder activarlo y dar continuidad en la red, en esos instantes es donde se presentaron los picos de latencia en la plataforma.

La recepción de la información tampoco se la consideró óptima debido al tipo de antena que se usaba para esta librería, la cual es la antena PCB. Este tipo de antena carece de buena recepción por la parte posterior y también por las consideraciones relacionadas al ambiente, teniendo en cuenta que en el perímetro del lago de ESPOL existe la presencia de varios árboles, lo cual complica la correcta recepción de datos. La red llegó a cubrir una distancia de 129.95 m, con una distancia máxima entre nodos de 53.06 m y una mínima de 9.81 m. Estos datos solo llegan a cubrir alrededor de un 16.92% de la distancia propuesta. La ubicación de los nodos puede ser evidenciado en el *Figura 51*.

En este caso, se buscó la mayor cantidad de linealidad a lo largo del perímetro del lago ESPOL, pero debido a la gran presencia de árboles la relación entre distancias de los nodos se fue deteriorando de manera abrupta, dando poco alcance a pesar de la linealidad de la red.

Figura 50

Datos de latencia obtenidos en escenario 3: caso 1 – painlessMesh



Caso 2. Los obtenidos presentan un valor de latencia promedio de 475.86 ms, en conjunto con valores mínimos y máximos de 8 ms y 2.256 segundos respectivamente. A medida que la cantidad de nodos aumentaba, la latencia promedio iba en aumento, esto puede ser observado en *Figura 52*. Situaciones similares al caso 1 se lograron observar al momento de alcanzar las distancias máximas, se presentaron los picos de latencia en la plataforma. Esta red logró una distancia de 181.42 m con una distancia máxima de 61.28 m y una mínima de 28.62 m. Dada la distancia obtenida por parte de la red, esta simplemente representa un 23.07% de la distancia meta. La ubicación de los nodos puede ser evidenciado en el *Figura 53*. En este caso se decidió poner a prueba la red bajo condiciones no completamente lineales dentro del perímetro del lago ESPOL, realizando pruebas de conexión en una curvatura del perímetro para su análisis.

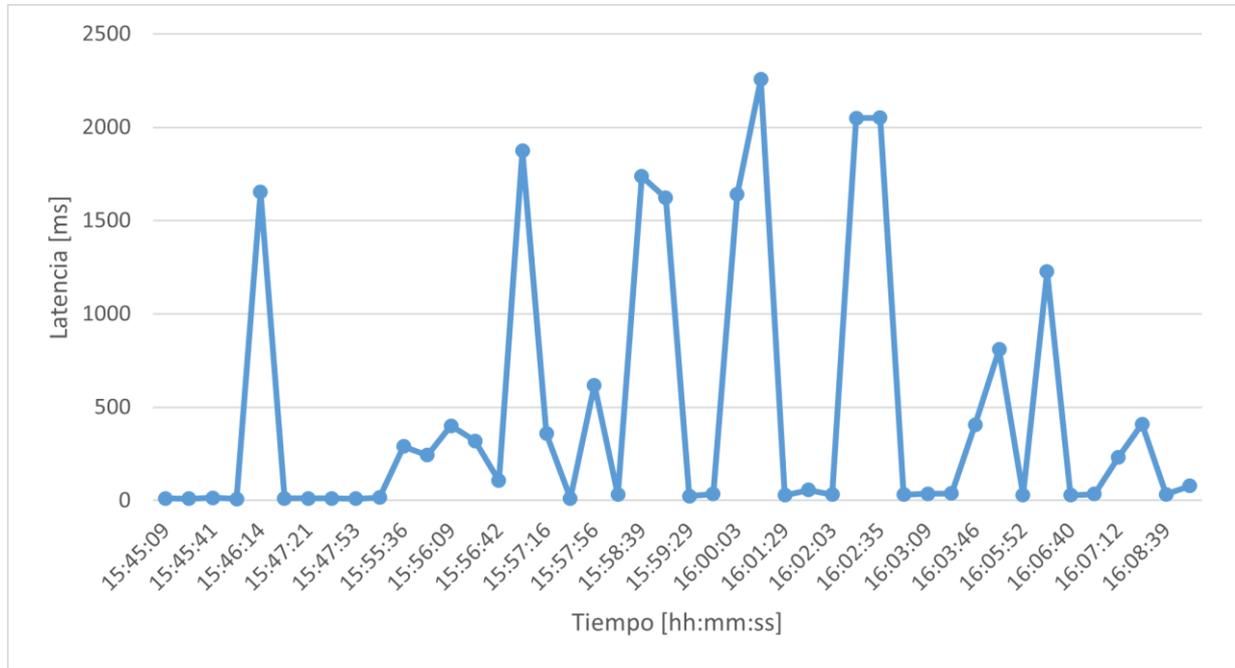
Figura 52

Datos obtenidos en ThingSpeak - painlessMesh



Figura 53

Datos de latencia obtenidos en escenario 3: caso 2 – painlessMesh



RF24 Mesh

Caso 1. Los valores obtenidos en el lago ESPOL arrojan resultados con valores de latencia de 210 ms y 492 ms como valores mínimos y máximos respectivamente, y con una latencia promedio de 225.82 ms. En su mayoría, la latencia se mantiene alrededor de valores entre 210 y 250 ms en la red como se puede observar en *Figura 54*. Esta red logró obtener una distancia de 293.60 m, con distancias máximas y mínimas entre nodos de 161.13 y 8.76 m respectivamente, lo cual representa un 37.37% de éxito respecto a la distancia meta. La ubicación de los nodos se visualiza en *Figura 55*.

Se debe considerar la múltiple presencia de árboles, así como la no tan estricta linealidad presente en varios tramos a lo largo del perímetro del lago. Para este caso, se buscó la comunicación estrictamente en el perímetro, lo cual, de manera similar al caso 1 haciendo uso de painlessMesh no se dio de la mejor manera debido a la presencia de varios árboles.

Figura 54

Datos de latencia obtenidos en escenario 3: caso 1 – RF24

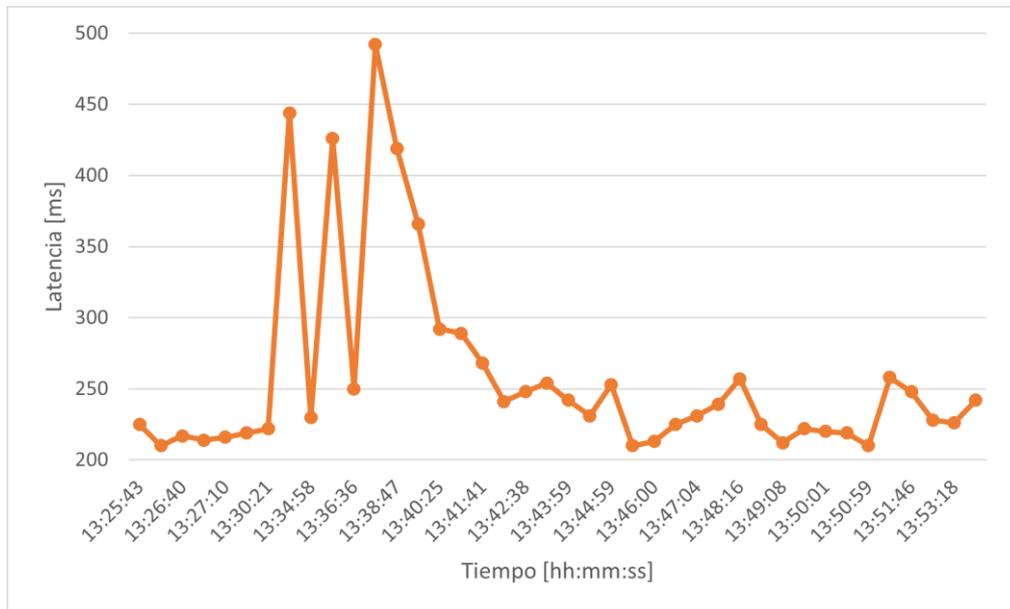


Figura 55

Ubicación de los nodos RF24 en el lago ESPOL



Caso 2. Los obtenidos presentan un valor de latencia promedio de 253.219 ms, en conjunto con valores mínimos y máximos de 91 ms y 499 ms respectivamente. En este caso, no se pudo obtener un patrón de latencia definido dentro del tiempo de prueba, esto puede ser observado en *Figura 57*. Esta falta de definición en el patrón se puede deber a dos situaciones presentes dentro del ambiente y situación de estudio. A diferencia del caso 1, se decidió realizar la transmisión de datos sobre el cuerpo marítimo, lográndose de manera exitosa, pero existía una abundante presencia de árboles, lo cual dificulta en mayor medida la transmisión de datos a través de los nodos.

Se logró una distancia máxima de 165.34 m y una mínima de 69.89 m. Se obtuvo una comunicación lineal de 408.37 m, lo cual representa únicamente un 51.95% de la distancia. La ubicación de los nodos puede ser evidenciado en el *Figura 56*.

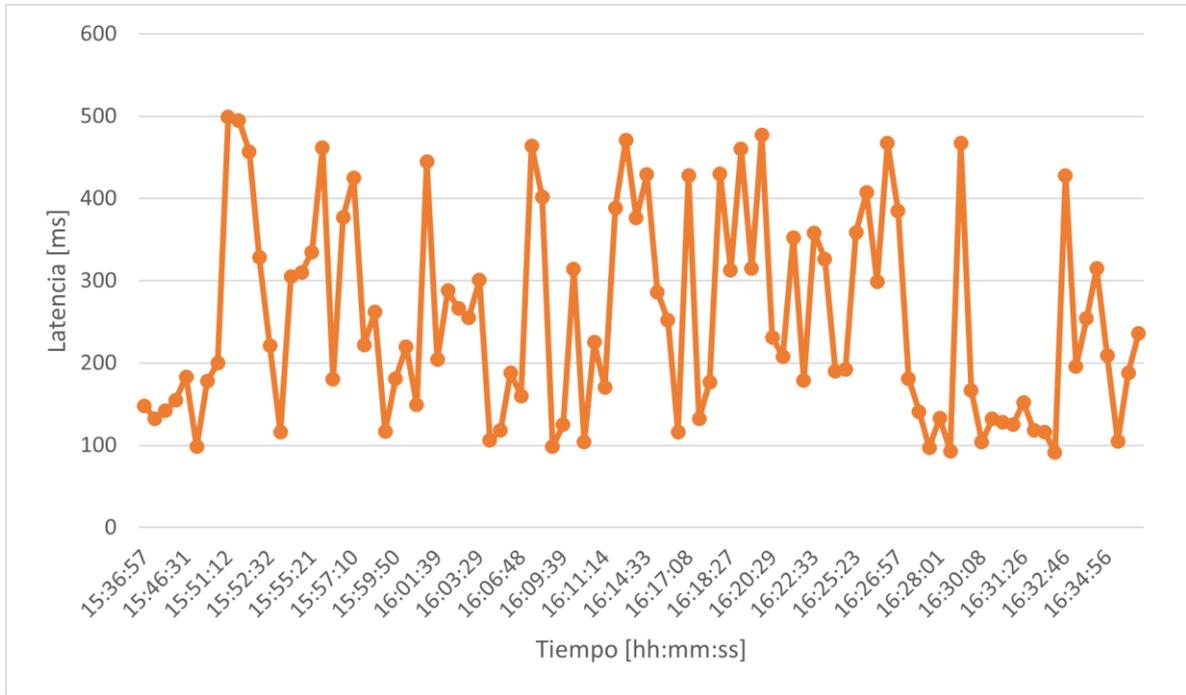
Figura 56

Ubicación de los nodos RF24 en el lago ESPOL



Figura 57

Datos de latencia obtenidos en escenario 3: caso 2 – RF24



3.2.5 Recopilación de datos

Estos datos son resumidos a continuación por medio de *Tabla 2-5*. En estas tablas se presentan los rangos de latencia obtenidos por cada nodo repetidor involucrado en la red, en conjunto con la distancia total de la red y su porcentaje de éxito respecto a la meta de 786.13 m.

Tabla 2

Resumen de datos recopilados del escenario 1

<i>Escenario 1</i>						
Rangos	Latencia sin repetidores [ms]	Latencia con Repetidor 1 [ms]	Latencia con Repetidor 2 [ms]	Latencia con Repetidor 3 [ms]	Distancia alcanzada [m]	Porcentaje respecto a la meta
Librería						
<i>painlessMesh</i>	[6,474]	[9,3067]	[22,3044]	[2,10445]	145.52	18.51%
<i>RF24 Mesh</i>	[82,124]	[114,118]	[72,82]	[102,131]	441.86	56.21%

Tabla 3*Resumen de datos recopilados del escenario 2*

<i>Escenario 2</i>						
<i>Rangos</i>	<i>Latencia sin repetidores</i>	<i>Latencia con Repetidor 1</i>	<i>Latencia con Repetidor 2</i>	<i>Latencia con Repetidor 3</i>	<i>Distancia alcanzada [m]</i>	<i>Porcentaje respecto a la meta</i>
<i>Librería</i>	<i>[ms]</i>	<i>1 [ms]</i>	<i>2 [ms]</i>	<i>[ms]</i>		
<i>painlessMesh</i>	<i>[70,300]</i>	<i>[90,3100]</i>	<i>[110,3300]</i>	<i>[200,11350]</i>	<i>237.85</i>	<i>30.26%</i>
<i>RF24 Mesh</i>	<i>[210,310]</i>	<i>[214,243]</i>	<i>[210,254]</i>	<i>[218,258]</i>	<i>843.19</i>	<i>107.26%</i>

Tabla 4*Resumen de datos recopilados del escenario 3 – caso 1*

<i>Escenario 3 – Caso 1</i>						
<i>Rangos</i>	<i>Latencia sin repetidores</i>	<i>Latencia con Repetidor 1</i>	<i>Latencia con Repetidor 2</i>	<i>Latencia con Repetidor 3</i>	<i>Distancia alcanzada [m]</i>	<i>Porcentaje respecto a la meta</i>
<i>Librería</i>	<i>[ms]</i>	<i>1 [ms]</i>	<i>2 [ms]</i>	<i>3 [ms]</i>		
<i>painlessMesh</i>	<i>[101,300]</i>	<i>[184,560]</i>	<i>[184,4900]</i>	<i>[143,5498]</i>	<i>129.95</i>	<i>16.53%</i>
<i>RF24 Mesh</i>	<i>[210,444]</i>	<i>[240,491]</i>	<i>[209,253]</i>	<i>[209,257]</i>	<i>293.60</i>	<i>37.34 %</i>

Tabla 5*Resumen de datos recopilados del escenario 3 – caso 2*

<i>Escenario 3 – Caso 2</i>						
<i>Rangos</i>	<i>Latencia sin repetidores</i>	<i>Latencia con Repetidor 1</i>	<i>Latencia con Repetidor 2</i>	<i>Latencia con Repetidor 3</i>	<i>Distancia alcanzada [m]</i>	<i>Porcentaje respecto a la meta</i>
<i>Librería</i>	<i>[ms]</i>	<i>1 [ms]</i>	<i>2 [ms]</i>	<i>3 [ms]</i>		
<i>painlessMesh</i>	<i>[10,1800]</i>	<i>[70,1930]</i>	<i>[60,2200]</i>	<i>[70,2250]</i>	<i>181.42</i>	<i>23.08%</i>
<i>RF24 Mesh</i>	<i>[98,499]</i>	<i>[98,471]</i>	<i>[116,477]</i>	<i>[91,467]</i>	<i>408.37</i>	<i>51.95%</i>

3.2 Costos de implementación

Para la implementación de este proyecto fue necesario hacer uso de los recursos detallados a continuación. Dentro de *Tabla 6* y *Tabla 7*. Todos estos valores se encuentran en referencia a la tienda “Electrónica Orellana”, ubicada en P. Icaza y Boyacá, centro de la ciudad de Guayaquil, Ecuador.

Tabla 6

Detalle de costos usando painlessMesh

Cant.	Artículo	P.Unit.	P.Total
6	Módulo Node MCU Bluetooth/WiFi ESP32 WROOM 32	\$10.00	\$60.00
1	Módulo sensor Humedad/Temperatura DHT22/AM2302 blanco	\$6.50	\$6.50
1	Protoboard (830 puntos)	\$3.20	\$3.20
6	Jumpers macho-macho	\$0.05	\$0.30
2	Jumpers hembra-hembra	\$0.05	\$0.10
1	Capacitor 10MF-50V	\$0.05	\$0.05
6	Baterías portátiles (Power Banks)	\$10.00	\$60.00
		TOTAL	\$130.15

Tabla 7

Detalle de costos usando RF24 Mesh

Cant.	Artículo	P.Unit.	P.Total
6	Módulo Node MCU Bluetooth/WiFi ESP32 WROOM 32	\$10.00	\$60.00
1	Módulo sensor Humedad/Temperatura DHT22/AM2302 blanco	\$6.50	\$6.50
1	Protoboard (830 puntos)	\$3.20	\$3.20
6	Jumpers macho-macho	\$0.05	\$0.30
37	Jumpers hembra-hembra	\$0.05	\$1.85
1	Capacitor 10MF-50V	\$0.05	\$0.05
6	Baterías portátiles (Power Banks)	\$10.00	\$60.00
5	Módulos Wireless NRF24L01+PA+LNA+ANTENA	\$3.80	\$19.00
		TOTAL	\$150.90

La diferencia en la implementación entre ambos sistemas de comunicación solo difiere por el uso de los módulos nRF24+PA+LNA con su respectiva antena y jumpers para poder conectar el módulo con el microcontrolador. Esta diferencia se ve reflejada de manera económica por \$20,00 (veinte dólares americanos). Por lo que, para la implementación de largo alcance, es recomendable la adquisición de estos módulos de radiofrecuencia

Capítulo 4

4.1 Conclusiones y recomendaciones

4.1.1 Conclusiones

- Dentro del desarrollo del proyecto se pudo determinar la distancia máxima de cada una de las redes en 3 diferentes escenarios haciendo uso exclusivo de 6 dispositivos ESP32 Device Kit. La distancia obtenida por parte de las redes painlessMesh se encuentran entre los 129.95 m y los 237.85 m con latencias entre 461.52 ms y 906.51 ms promedio. Se puede determinar que este tipo de redes puede ser aplicables a zonas de pequeño a medio alcance, ya que sin contar con una externa que pueda sustituir a la antena PCB del ESP32, se dificulta en gran medida la transmisión y recepción de datos. Por otra parte, RF24 Mesh presenta distancias entre 293.60 m y 843.19 m con promedios en latencia entre 102.89 ms y 258.02 ms. Este tipo de redes puede ser aplicable para zonas de pequeño, mediano y largo alcance, siempre y cuando se configure de manera correcta la potencia de transmisión y su tasa de datos, así como, la configuración de la ganancia de recepción.
- La presencia de varios dispositivos altera el rendimiento de la red, ya que la red debe mantenerse sincronizada con el constante envío y recepción de datos sobre la estructura de la red, notificando si algún nuevo dispositivo se ha enlazado o desaparecido, la elección del nodo de cabecera, hacia qué nodo se encuentra conectado y generar una base de datos sobre esta red. El tamaño de la red se encontrará limitado por la memoria de nuestros dispositivos, por lo que, a mayor cantidad de dispositivos, menor será el espacio de la memoria que se encuentre disponible para el procesamiento de la información enviada. En cuanto a optimización y resiliencia, un claro ejemplo sería *Figura 45*, donde los nodos se encuentran en constante ordenamiento de nodos para determinar la mejor ruta del envío de los datos.
- En cuanto a diferencias en topologías, painlessMesh ofrece arreglos en su topología mucho más amplios que RF24 Mesh, ya que este último solo puede ordenar su topología en una

de árbol, limitando la cantidad de nodos hijos en 5 dispositivos y solo hasta 4 niveles de red posteriores al del nodo de cabecera. Sin embargo, RF24 Mesh permite un mejor manejo en cuanto a niveles de transmisión, ganancia de recepción y tasa de transmisión, características que painlessMesh no cuenta.

- Las interferencias producidas a lo largo de la obtención de datos de todos los escenarios presentados pertenecieron a agentes externos como redes Wi-Fi o Bluetooth presentes en la localidad, torres de alta tensión, la presencia de árboles, bancas con techado, el paso de vehículos semi pesados y las múltiples reflexiones producidas por el agua. Todos estos factores producen alteraciones dentro de la propagación de las señales, reflejándose en el aumento de los niveles de latencia de manera puntual o a lo largo del tiempo.

4.1.2 Recomendaciones

- Para futuras implementaciones o estudios, se debería de realizar implementaciones de seguridad para los datos, así como la aplicación de otras librerías que permitan establecer redes Mesh para ampliar la gama de posibilidades en función a las necesidades del usuario.
- Para implementaciones se recomienda hacer uso de plataformas que no requieran de tanto tiempo para poder realizar la escritura de datos, como Node-Red, el cual recibe datos en tiempo real y permite una mayor organización de datos y presentaciones por pantalla para el monitoreo de la red, así como de los datos recibidos.
- Se sugiere la implementación de paneles solares para que los nodos se mantengan energizados, ahorrando costos energéticos para la carga de las baterías usadas por cada uno de los dispositivos.
- Para futuras investigaciones, se recomienda el estudio del impacto de la temperatura y de la humedad en el rendimiento de la red. Así como, el rendimiento de la red en largos periodos de tiempo.

Referencias

- [1] D. Rico-Bautista, L. Sánchez-Espinosa y E. Y. Portillo-Ballesteros, «Redes Mesh, una alternativa a problemas de cobertura de red: una revisión de literatura,» Diciembre 2014. [En línea]. Available: <https://revistas.ufps.edu.co/index.php/ingenio/article/view/2031>. [Último acceso: 13 Mayo 2023].
- [2] Z. Nurlan, T. Z. Kokenovna y M. Othman, «Resource Allocation Approach for Optimal Routing in IoT Wireless Mesh Networks,» 27 Octubre 2021. [En línea]. Available: <https://ieeexplore.ieee.org/document/9591636>. [Último acceso: 13 Mayo 2023].
- [3] CISCO Systems, “Cisco Annual Internet Report (2018–2023) White Paper”, Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. [Último acceso: 26 Julio 2023]
- [4] Monika Kumari, Dr. Vivek Kumar, Mr. Dayal Chandra Sati, “Establishing a Wireless-Local-Area-Network (WLAN) Connectivity between Multiple Nodes using ESP-Mesh Network Topology for IoT Applications”, IRJET vol. 7, Agosto, 2020.
- [5] Song Wen, Wang Fei, Dai Jianbo, “An Emergency Communication System Based on WMN in Underground Mine”, International Conference on Computer Application and System Modeling (ICCSM), Taiyuan, China, Octubre, 2010.
- [6] Cámara Nacional de Acuicultura, «Cámara Nacional de Acuicultura,» 19 Noviembre 2017. [En línea]. Available: <https://www.cna-ecuador.com/el-camaron-se-convierte-en-el-primero-producto-de-exportacion-no-petrolera-del-pais/>. [Último acceso: 3 Julio 2023]
- [7] S. Abad-Rosales, M. Betancourt-Lozano, F. Vargas-Albores, A. Roque*, “Interaction of physical, chemical and biological factors in shrimp culture”, Enero 2011.
- [8] M. Kumari, D. V. Kumar y M. D. C. Sati, «Establishing a Wireless-Local-Area-Network (WLAN) Connectivity between Multiple Nodes using ESP-Mesh Network Topology for IoT Applications,» *International Research Journal of Engineering and Technology*, vol. 7, nº 08, Agosto 2020.
- [9] G. Sacarelo, G. Romero, P. Chávez y D. Ochoa, «Diseño y evaluación de una red de sensores para analizar entornos acuáticos,» *FIEC, ESPOL*, 2014.
- [10] T. D. Le y D. H. Tan, «Design and Deploy a Wireless Sensor Network for Precision Agriculture,» ”, *2nd National Foundation for Science and Technology Development*

Conference on Information and Computer Science, (NICS), Ho Chi Minh City, Vietnam, October, 2015.

- [11] L. Sanchez, «Banco de pruebas de una Red Inalámbrica Mesh basada en el protocolo IEEE 802.11s,» *Universidad de Zaragoza*, 2010.
- [12] BlueSensor, «BlueSensor,» [En línea]. Available: <https://bluesensordata.com>.
- [13] E. DiPascale, I. Macaluso, A. Nag, M. Kelly y L. Doyle, «The Network as a Computer: A Framework for Distributed Computing Over IoT Mesh Networks» *IEEE Internet of Things Journal*, vol. 5, June, 2018.
- [14] M. L. Sichitiu, «Wireless mesh networks: opportunities and challenges,» *World Wireless Congress*, Raleigh, NC, USA, 2005.
- [15] Fernandez-Hernandez, Jesus & Berrocal, José Luis & Figuerola, Carlos & Zazo, Angel; «Redes privadas virtuales, » *Monografía para titulación, Universidad Autónoma del Estado de Hidalgo, Pachuca de Hidalgo, México*. [En línea]. Available: https://www.researchgate.net/publication/39697459_Redес_privadas_virtuales
- [16] Polaridad.es, «Conoce el alcance del ESP32 en proyectos de electrónica,» [En línea]. Available: <https://polaridad.es/conoce-el-alcance-del-esp32-en-proyectos-de-electronica/>.
- [17] R. Mischianti, «DOIT ESP32 DEV KIT v1 high resolution pinout and specs,» February 2021. [En línea]. Available: <https://mischianti.org/2021/02/17/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs>.
- [18] ESPRESSIF, «ESP32,» [En línea]. Available: <https://www.espressif.com/en/products/socs/esp32>.
- [19] Arduino, «Painless Mesh,» [En línea]. Available: <https://www.arduino.cc/reference/en/libraries/painless-mesh/>
- [20] painlessMesh, «PainlessMesh,» GitLab [En línea]. Available: <https://gitlab.com/painlessMesh/painlessMesh>.
- [21] gmag11, «Painless Mesh,» 2019. [En línea]. Available: <https://github.com/gmag11/painlessMesh>.
- [22] N. Mechatronics. «MÓDULO RF 2.4 GHZ – NRF24L01+PA+LNA CON ANTENA,» [En línea]. Available: <https://naylampmechatronics.com/inalambrico/113-modulo-rf-nrf24l01-pa-lna-antena.html>.

- [23] N. Semiconductor, «nRF24L01+,» [En línea]. Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus_Preliminary_Product_Specification_v1_0.pdf.
- [24] L. M. Engineers, «How nRF24L01+ Wireless Module Works & Interface with Arduino,» [En línea]. Available: <https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>.
- [25] D. Arduino, «Cómo funciona el módulo inalámbrico nRF24L01 y su interfaz con Arduino,» [En línea]. Available: <https://descubrearduino.com/nrf24l01/>.
- [26] Arduino, «RF24» [En línea]. Available: <https://www.arduino.cc/reference/en/libraries/rf24/>
- [27] Arduino, «RF24 Network» [En línea]. Available: <https://www.arduino.cc/reference/en/libraries/rf24network/>
- [28] «Newly Optimized RF24Network Layer». GitHub. Available: <https://nrf24.github.io/RF24Network/index.html>
- [29] Arduino, «RF24 Mesh» [En línea]. Available: <https://www.arduino.cc/reference/en/libraries/rf24mesh/>
- [30] «RF24Mesh – Automated Networking for nrf24L01 radios». GitHub. Available: <https://nrf24.github.io/RF24Mesh/>
- [31] Y. Liu, K. Tong, X. Qiu, Y. Liu y X. Ding, «Wireless Mesh Networks in IoT Networks,» *International Workshop on Electromagnetics: Applications and Student Innovation Competition*, London, England, May, 2017.
- [32] ESPRESSIF, «ESP-WIFI-MESH,» [En línea]. Available: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/esp-wifi-mesh.html>.
- [33] U. Bhagat, N. Gujar and S. Patel, «Implementation of IOT in development of intelligent campus lighting system using mesh network, » 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2018, pp. 251-256, doi: 10.1109/ICSSIT.2018.8748314.
- [34] A.Guevara-Villacres, H. Morán-Guevara, D. Hernández-Rojas y J. Cartuche-Calva. «Implementation of a low-cost-ad-hoc IoT mesh network for home automation systems,» Trabajo de titulación Tecnología de la Información. Universidad Técnica de Machala. [En línea]. Available: <http://repositorio.utmachala.edu.ec/handle/48000/21294>

[35] Matlab. «*ThingSpeak*, » MathWorks: Centro de ayuda. [En línea]. Available: <https://thingspeak.com/>