
Modelado y validación de topologías de comunicación para redes de sensores en escenarios de monitoreo remoto

Angela Marina Quezada Orozco



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Modelado y validación de topologías de comunicación para
redes de sensores en escenarios de monitoreo remoto**

Trabajo de graduación presentado por Angela Marina Quezada Orozco
para optar al grado académico de Licenciada en Ingeniería Mecatrónica

Guatemala,

2025

Vo.Bo.:

(f) 
M.Sc. Miguel Enrique Zea Arenales

(f) 
M.Sc. Carlos Alberto Esquit Hernández

Fecha de aprobación: Guatemala, 28 de noviembre de 2025.

Quiero expresar mi más profundo agradecimiento a todas las personas que han sido parte de este camino y que, de una u otra forma, me han acompañado hasta alcanzar esta meta.

A mi mamá Ivonne, gracias por tu amor inmenso y por cada sacrificio que hiciste para que yo pudiera cumplir mis sueños. Gracias por siempre creer en mí, por enseñarme a ser fuerte y por estar conmigo en los momentos más difíciles. Gracias por tus palabras, tu ejemplo y tu apoyo incondicional.

A mi papá Víctor, gracias por todas las enseñanzas que me dejaste, por mostrarme el valor del esfuerzo y por recordarme siempre que podía alcanzar todo lo que me propusiera. Aunque ya no estés, tu recuerdo me acompañó y me dio fuerzas para llegar hasta aquí.

A mis hermanas Marcela y Pamela, gracias por estar conmigo en cada etapa, por su cariño y por hacer más llevaderos los momentos difíciles. Gracias por cada palabra de ánimo, por cada risa compartida y por recordarme que nunca camino sola.

A mi abue Hilda y a mi abuelito Rolando, gracias por su guía, sus palabras de ánimo, cuidados y apoyo. Gracias por estar siempre presentes, por su cariño y por enseñarme, con su ejemplo, a ser fuerte y a seguir adelante.

A Jorge, gracias por tu compañía a lo largo de la carrera, por tu apoyo constante y por cada palabra que me ayudó a continuar. Gracias por estar conmigo en los días complicados, por escucharme y por recordarme siempre que era capaz.

A mis amigos de la universidad, gracias por todos los momentos compartidos.

También quiero agradecer a todos los maestros que se esforzaron en impartir sus clases con dedicación y que hicieron que disfrutara la carrera. Agradezco las oportunidades que me brindaron y todas las enseñanzas que aportaron a mi formación.

Finalmente, expreso mi sincero agradecimiento a mi asesor, Miguel Zea, por su dedicación, orientación y disposición para guiarme a lo largo de la tesis.

Prefacio	I
Índice de figuras	IV
Índice de cuadros	V
Resumen	VI
Abstract	VII
1. Introducción	1
2. Antecedentes	2
3. Justificación	4
4. Objetivos	5
4.1. Objetivo general	5
4.2. Objetivos específicos	5
5. Alcance	6
6. Marco teórico	8
6.1. Redes de sensores inalámbricos	8
6.2. Topologías de red	9
6.3. Tecnologías de comunicación	9
6.4. Teoría algebraica de grafos aplicada a redes de sensores inalámbricos	11
7. Desarrollo teórico y experimental de la topología para grafos no dirigidos y dirigidos	15
7.1. Construcción inicial para grafos no dirigidos	15
7.2. Análisis manual de grafos no dirigidos	16
7.3. Implementación de grafos no dirigidos en MATLAB	17
7.4. Análisis de casos de desconexión para grafo preliminar no dirigido	19

7.5.	Iteración del grafo no dirigido, aplicación de triangulación	20
7.6.	Análisis de casos de desconexión para grafo triangulado no dirigido	20
7.7.	Construcción inicial para grafos dirigidos	21
7.8.	Implementación de algoritmo de Kosaraju en <i>MATLAB</i> para análisis de grafos dirigidos	22
7.9.	Comparación entre tipos de grafos y relación con topologías de comunicación	24
8.	Tecnologías de comunicación	26
8.1.	Comparación entre tecnologías de comunicación	26
8.2.	Entorno de desarrollo ESP-NOW	27
8.3.	Estructura del proyecto para pruebas de latencia ESP-NOW	28
8.4.	Flujo de mensajes <i>multi-hop</i> en <i>ESP-NOW</i>	28
8.5.	Propagación de mensajes en la red en ESP-NOW	29
8.6.	Resultados de pruebas de latencia en ESP-NOW	30
8.7.	Implementación final ESP-NOW	32
8.8.	Resultados implementación final ESP-NOW	33
8.9.	Entorno de desarrollo LoRa	34
8.10.	Implementación de modo <i>single-hop</i> LoRa	34
8.11.	Resultados de modo <i>single-hop</i> LoRa	35
8.12.	Implementación de modo <i>multi-hop</i> LoRa	35
8.13.	Resultados de modo <i>multi-hop</i> LoRa	37
8.14.	Pruebas de alcance LoRa	37
9.	Conclusiones	40
10.	Recomendaciones	41
11.	Referencias	42
12.	Anexos	45
12.1.	Laplacianas por caso de eliminación (grafos no dirigidos)	45
12.2.	Scripts de grafos no dirigidos en MATLAB	47
12.3.	Resumen de resultados para el grafo no dirigido preliminar	48
12.4.	Resumen de resultados para el grafo no dirigido triangulado	49
12.5.	Scripts de grafos dirigidos MATLAB	49
12.6.	Resumen de resultados para el grafo dirigido	50
12.7.	Scripts pruebas de latencia	50
12.8.	Resultados de pruebas de latencia	50
12.9.	Script implementación final ESP-NOW	55
12.10.	Resultados implementación final ESP-NOW	55
12.11.	Scripts implementación LoRa	55

Índice de figuras

1.	Ejemplos de topologías de red	10
2.	Ejemplos de grafos no dirigidos y grafos dirigidos	13
3.	Grafo preliminar para el lago de Atitlán	16
4.	Grafo original y grafo reducido tras la eliminación del nodo 3	19
5.	Proyección geográfica y grafo triangulado	20
6.	Grafo resultante tras la eliminación de nodo 3	21
7.	Grafo dirigido original y grafo reducido sin el nodo 3	23
8.	Propagación de mensaje en red completa (5-1)	30
9.	Propagación de mensaje en red sin nodo 3 (5-1)	31
10.	Implementación física de la red de nodos sobre mapa de prueba	33
11.	Resultados de latencia para modo <i>single-hop</i>	36
12.	Resultados de latencia para modo <i>multi-hop</i>	38
13.	Implementación física de la red LoRa sobre mapa de prueba	38
14.	Resultados de pruebas de alcance en terreno con obstáculos	39
15.	Resultados de pruebas de alcance en terreno con línea de vista	39
16.	Resultados para red completa	55
17.	Resultados sin nodo 6 y 8	55

1.	Resumen por tamaño de subgrafo (<i>Remaining</i>)	24
2.	Comparación entre grafos analizados y topologías clásicas de comunicación . .	25
3.	Comparación entre tecnologías de comunicación	27
4.	Prueba de latencia en red completa	31
5.	Prueba de latencia en red sin nodo 3	32
6.	Peores 5 casos entre conexos (valor mínimo de λ_2)	48
7.	Mejores 5 casos entre conexos (valor máximo de λ_2)	48
8.	Peores 5 casos entre conexos (valor mínimo de λ_2) para el grafo triangulado .	49
9.	Mejores 5 casos entre conexos (valor máximo de λ_2) para el grafo triangulado	49
10.	Resumen por tamaño de subgrafo	50
11.	Prueba de latencia en red sin nodo 2	51
12.	Prueba de latencia en red sin nodo 4	51
13.	Prueba de latencia en red sin nodo 5	51
14.	Prueba de latencia en red sin nodo 6	52
15.	Prueba de latencia en red sin nodo 7	52
16.	Prueba de latencia en red sin nodo 8	52
17.	Prueba de latencia en red sin nodos 3 y 4	53
18.	Prueba de latencia en red sin nodos 3 y 5	53
19.	Prueba de latencia en red sin nodos 5 y 6	53
20.	Prueba de latencia en red sin nodos 4, 5, 7 y 8	54
21.	Resultados de latencia para la red sin nodos 4, 5, 6, 7 y 8	54

Las redes de sensores inalámbricos ampliamente utilizados en aplicaciones de monitoreo remoto donde la continuidad de la comunicación es esencial. En estos entornos, la pérdida de nodos podía comprometer la transmisión de datos y reducir la confiabilidad del sistema, por lo que resultaba necesario evaluar arquitecturas que mantengan operación aún ante fallas.

Se desarrolló y analizó una red WSN tolerante a fallos combinando teoría de grafos y experimentación con tecnologías ESP-NOW y LoRa. El análisis matemático incluyó grafos dirigidos, no dirigidos y grafos de visibilidad para caracterizar la robustez estructural y el efecto de remover nodos sobre la conectividad. En las pruebas experimentales se evaluaron la latencia y la comunicación multisalto, observándose que ESP-NOW ofreció baja latencia pero fue sensible a fallos en nodos centrales, mientras que LoRa presentó una mayor latencia pero conservó enlaces estables incluso con múltiples pérdidas. Los resultados permitieron identificar configuraciones más resistentes y establecer lineamientos para el diseño de redes de monitoreo remoto robustas y tolerantes a fallos.

Palabras clave: redes de sensores inalámbricos, tolerancia a fallos, ESP-NOW, LoRa, grafos, conectividad, latencia, multisalto.

Wireless sensor networks (WSNs) were widely used in remote monitoring applications where continuous communication was essential. In such environments, node failures could compromise data transmission and reduce system reliability, making it necessary to evaluate architectures capable of maintaining operation even under fault conditions.

This work developed and analyzed a fault-tolerant WSN by combining graph-theoretical modeling with experimental evaluation using ESP-NOW and LoRa technologies. The mathematical analysis included directed graphs, undirected graphs, and visibility graphs to characterize structural robustness and quantify the impact of node removal on network connectivity. Experimental tests assessed latency and multi-hop communication, showing that ESP-NOW provided low latency but was sensitive to failures in central nodes, whereas LoRa exhibited higher latency but preserved stable links even under multiple node losses. The results helped identify more resilient configurations and provided guidelines for the design of robust and fault-tolerant WSNs for remote monitoring applications.

Keywords: wireless sensor networks, fault tolerance, ESP-NOW, LoRa, graphs, connectivity, latency, multi-hop.

En la actualidad, las redes de sensores inalámbricos (*Wireless Sensor Networks, WSN*) constituyen una de las tecnologías más relevantes para el monitoreo de variables ambientales, industriales y estructurales en tiempo real. Su capacidad de integrar sensado, procesamiento y comunicación en sistemas distribuidos las convierte en una herramienta fundamental para aplicaciones en agricultura, salud, industria y monitoreo ambiental. No obstante, el diseño de estas redes enfrenta desafíos significativos en escenarios reales, especialmente en contextos donde las condiciones geográficas y sociales limitan la infraestructura de comunicación y donde los fenómenos naturales pueden comprometer la continuidad de los sistemas.

En países como Guatemala, la alta vulnerabilidad ante sismos, erupciones volcánicas, deslizamientos o lluvias extremas resalta la necesidad de contar con soluciones tecnológicas resilientes, capaces de garantizar la transmisión de datos incluso bajo condiciones adversas. En este contexto, el presente trabajo se centra en el diseño y validación de una red de comunicación inalámbrica y tolerante a fallos, orientada a aplicaciones de monitoreo remoto mediante *WSN*.

La investigación se delimita a tres ejes principales. Por un lado, el análisis matemático de distintas topologías de red utilizando herramientas de teoría algebraica de grafos, lo cual permite evaluar la conectividad y la robustez de las arquitecturas ante la eliminación de nodos. Por otro lado, la comparación de diferentes tecnologías de comunicación (ESP-NOW, nRF24L01 y LoRa), considerando diferentes factores, con el fin de identificar la opción más adecuada para escenarios de difícil acceso como el lago de Atitlán. Para finalmente realizar pruebas a escala del trabajo realizado.

Metodológicamente, el trabajo combina el modelado teórico y el análisis computacional con pruebas experimentales. Inicialmente, se construyeron grafos representativos de la red y se evaluaron métricas como la conectividad algebraica (valor de Fiedler, λ_2) y las componentes fuertemente conexas mediante el algoritmo de Kosaraju. Posteriormente, se implementaron prototipos físicos a escala con microcontroladores ESP32 programados bajo ESP-NOW, ejecutando pruebas de latencia y propagación de mensajes *multi-hop* para validar la tolerancia a fallos en condiciones simuladas de pérdida de nodos.

En los últimos años, las *Wireless Sensor Networks* (WSN, Redes de Sensores Inalámbricos) han cobrado mayor relevancia debido a su capacidad para monitorear entornos complejos en tiempo real.

Las WSN. se encuentran formadas por nodos de sensores distribuidos espacialmente, permitiendo observar en tiempo real diversas variables del entorno como temperatura, humedad, calidad del aire, calidad del agua o la presencia de incendios forestales. Las WSN se utilizan en múltiples aplicaciones como la agricultura, el seguimiento de animales, la detección de terremotos y deslizamientos, y la supervisión de ecosistemas acuáticos y atmosféricos[1].

En el estudio efectuado por Ragnoli et al. [2], se implementó un sistema autónomo de monitoreo de deslizamientos de tierra y caída de rocas en la isla de Pantelleria (Italia), utilizando una WSN basada en la tecnología LoRa. El sistema estaba compuesto por 12 nodos sensores que recopilaban datos sobre aceleración, temperatura, humedad, presión, altitud, y calidad del aire. Estos nodos se comunicaban con dos *gateways* mediante una topología en estrella, donde los nodos envían la información directamente al *gateway* más cercano utilizando comunicación de salto único (*single-hop*). El diseño empleó un despliegue disperso, con sensores ubicados en diferentes niveles y zonas del terreno rocoso, fijados sobre barreras de protección y conectados a través de enlaces de radio de largo alcance.

En la revisión realizada por Ragnoli et al. [3], se recopilan y analizan diversas implementaciones de sistemas de monitoreo de deslizamientos de tierra y caída de rocas basados en WSN. El estudio identifica que las arquitecturas más comunes en este tipo de aplicaciones son las de topología en estrella y en malla. En terrenos con conectividad limitada o con nodos distribuidos en zonas extensas, las configuraciones *multi-hop* o de malla parcial resultan más eficientes para mantener la comunicación entre nodos y garantizar tolerancia a fallos. También se destaca el uso de tecnologías como LoRa, ZigBee, NB-IoT y wifi, las mismas son seleccionadas según la topología, la necesidad de ancho de banda y el consumo energético deseado.

En el trabajo de Prabha et al. [4], se abordó la resolución de ecuaciones de grafos rela-

cionadas con redes de sensores inalámbricos, utilizando estructuras matemáticas derivadas de la teoría de grafos, como los *detour distance graphs*, *detour antipodal graph* y *line graphs*. Estas estructuras permiten modelar eficientemente las relaciones entre nodos sensores, optimizando aspectos críticos como el consumo energético y la precisión en la localización.

En el estudio de Musznicki et al. [5], se desarrolló una arquitectura de modelado para redes de sensores urbanos basada en datos abiertos, con el objetivo de representar y analizar la conectividad en redes heterogéneas mediante el uso de algoritmos orientados a *space-time graphs*. A partir de datos reales sobre la ubicación de dispositivos conectados en cuatro ciudades polacas, los autores generaron topologías estáticas y dinámicas que permitieron estudiar el comportamiento de la red ante cambios en el tiempo y el espacio. El enfoque propuesto facilitó el análisis de métricas clave como la conectividad, el grado de los nodos, y la robustez de los enlaces.

Las redes de sensores inalámbricos han cobrado una gran relevancia en el monitoreo remoto de variables ambientales, industriales y estructurales, especialmente en zonas de difícil acceso o con infraestructura limitada. Sin embargo, uno de los principales desafíos que enfrentan es garantizar una comunicación confiable y de largo alcance, que además sea capaz de mantener la operación ante posibles fallos en la red o desconexión de nodos.

Este desafío adquiere una mayor relevancia en países como Guatemala, donde no se cuenta con infraestructuras de respuesta rápida frente a desastres naturales o ante cambios súbitos en la calidad del aire o del agua. La vulnerabilidad del territorio ante fenómenos como erupciones volcánicas, sismos, deslizamientos y lluvias extremas hace indispensable el desarrollo de tecnologías que permitan la recolección y transmisión de datos en tiempo real, incluso en condiciones adversas.

Además, debido a estas mismas condiciones geográficas y sociales, es probable que algunos nodos de la red se pierdan o se vean comprometidos, ya sea por daños físicos ocasionados por la actividad volcánica o sísmica, o por robos y vandalismo. Por esto es esencial que la arquitectura de red diseñada sea tolerante a fallos, es decir, que pueda adaptarse a la pérdida de uno o varios nodos sin comprometer la funcionalidad global del sistema.

Estas necesidades serán abordadas a través del diseño y validación de una red de comunicación de largo alcance y tolerante a fallos, enfocada en aplicaciones de monitoreo remoto mediante redes de sensores. Dicha solución se desarrollará mediante el análisis de topologías, el estudio de tecnologías y protocolos de comunicación inalámbrica, considerando aspectos como su alcance y confiabilidad. Finalmente, se realizará una validación experimental por medio de un prototipo físico a escala, que permita verificar el desempeño de la red propuesta bajo condiciones simuladas de pérdida de nodos, replicando escenarios reales como interrupciones por desastres naturales o robos de dispositivos.

4.1. Objetivo general

Diseñar y validar una red de comunicación de largo alcance y tolerante a fallos, orientado a redes de sensores utilizadas en aplicaciones de monitoreo remoto.

4.2. Objetivos específicos

- Investigar y analizar matemáticamente distintas topologías de comunicación aplicables a redes de sensores para determinar su viabilidad en entornos de monitoreo remoto y su tolerancia a fallos.
- Investigar tecnologías y protocolos de comunicación inalámbrica aplicables al diseño de redes de comunicación entre sensores.
- Realizar validaciones físicas a escala que verifiquen el desempeño del sistema de comunicación propuesto.

El presente trabajo se centra en el diseño, modelado y validación de una red de comunicación inalámbrica de largo alcance y tolerante a fallos, orientada a aplicaciones de monitoreo remoto mediante redes de sensores. El alcance abarca tanto el análisis matemático de topologías de red con teoría algebraica de grafos, como la implementación experimental de un prototipo físico a escala utilizando microcontroladores ESP32 y el protocolo ESP-NOW.

Dentro de este marco, el trabajo comprende:

- El análisis y comparación de topologías de red mediante grafos dirigidos y no dirigidos, evaluando su robustez frente a la eliminación de nodos a través de métricas como la conectividad algebraica (λ_2).
- La implementación de algoritmos como Kosaraju para verificar la fuerte conectividad en grafos dirigidos.
- La comparación entre tecnologías de comunicación inalámbrica (ESP-NOW, nRF24L01 y LoRa), considerando alcance, consumo energético y tolerancia a fallos.
- El desarrollo de un prototipo físico a escala que simula la propagación de mensajes en una red multi-hop, incluyendo pruebas de latencia y propagación de mensajes bajo escenarios de pérdida de nodos.

Sin embargo, el trabajo no incluye:

- La implementación de la red en un entorno real, como el lago de Atitlán u otras zonas de estudio.
- El uso de sensores físicos para la recolección de datos ambientales, ya que el enfoque se limita a la validación de la arquitectura de comunicación.
- El análisis de consumo energético detallado en condiciones de campo, restringiéndose a la validación de comunicación y latencia en laboratorio.

- La integración de *gateways* o infraestructuras de red externas, más allá del prototipo experimental.

Se utilizó ChatGPT como apoyo para organizar y sintetizar información. El texto final fue revisado y validado por la autora [6].

El proceso de depuración de código contó con el apoyo de GitHub Copilot [7].

El diseño de una red de comunicación tolerante a fallos requiere comprender tanto el contexto operacional de las redes de sensores inalámbricos (*Wireless Sensor Networks*, WSN), como los distintos esquemas de topologías de red que determinan la manera en que los nodos se organizan y se comunican entre sí. Posteriormente, es necesario el análisis de tecnologías de comunicación, ya que su alcance, consumo y robustez condicionan la viabilidad del sistema propuesto. Finalmente, se presenta la teoría algebraica de grafos, la cual proporciona las herramientas matemáticas para modelar la conectividad, evaluar la resiliencia de la red e identificar nodos críticos, fundamentos esenciales para el análisis y validación experimental.

6.1. Redes de sensores inalámbricos

Las redes de sensores inalámbricos WSN constituyen un modelo tecnológico que combina capacidades de sensado, procesamiento y comunicación en sistemas distribuidos, posibilitando la recolección de datos en tiempo real desde entornos físicos y remotos [8].

Una WSN está compuesta por nodos sensores que integran cuatro módulos principales: unidad de sensado, unidad de procesamiento, transceptor y fuente de energía. Cada nodo cumple tres funciones esenciales: captar información del entorno, procesarla localmente y transmitirla hacia un nodo *sink* que funciona como punto central de destino para los datos [8].

El desarrollo de las WSN surge como alternativa a los sistemas cableados, superando limitaciones de instalación y ofreciendo escalabilidad. Con los avances en microelectrónica, telecomunicaciones y sistemas embebidos, su aplicación se ha expandido a diferentes campos de aplicaciones [9] [1].

En el ámbito ambiental, se utilizan para detección de incendios forestales, monitoreo de la calidad del aire y del agua, o seguimiento de parámetros climáticos. En la industria, se emplean en supervisión de procesos, detección de fallos en maquinaria y gestión de in-

ventarios. En el sector militar, cumplen funciones de vigilancia, clasificación de objetivos y detección de intrusos. Finalmente, en el ámbito de la salud permiten el monitoreo remoto de pacientes a través de sensores portátiles [8].

6.2. Topologías de red

En términos de arquitectura, una WSN puede organizarse en distintas topologías de red, que se pueden observar en la Figura 1. Algunas de estas se detallan a continuación [10]:

- Topología en bus: todos los nodos están conectados a un único canal de comunicación. Es sencilla de implementar y adecuada para un número limitado de nodos, pero se vuelve ineficiente y propensa a colisiones cuando la red crece.
- Topología en estrella: los nodos sensores se comunican únicamente con un nodo central o coordinador. Es eficiente en consumo de energía y fácil de gestionar, pero su principal limitación radica en que si el nodo central falla, la red completa queda inoperante.
- Topología en anillo: cada nodo se conecta exactamente con dos vecinos formando un ciclo cerrado. Permite comunicación secuencial, pero introduce retardos y sufre vulnerabilidad: la caída de un nodo afecta la continuidad del anillo.
- Topología en árbol: también conocida como estrella jerárquica, organiza los nodos en niveles, con un nodo raíz que conecta con varios nodos intermedios y estos, a su vez, con otros subordinados. Ofrece escalabilidad y facilidad de gestión, aunque la pérdida de un nodo intermedio puede dejar incomunicada toda una rama de la red.
- Topología en malla (*mesh*): todos los nodos pueden comunicarse entre sí, de manera directa o a través de múltiples saltos (*multi-hop*). Su principal ventaja es la redundancia y tolerancia a fallos, pues la red puede redirigir automáticamente la información si un nodo falla. Sin embargo, esta topología demanda mayor consumo energético debido a las retransmisiones [10].

La elección de la topología depende de los requisitos de la aplicación, como alcance, robustez, latencia y consumo energético.

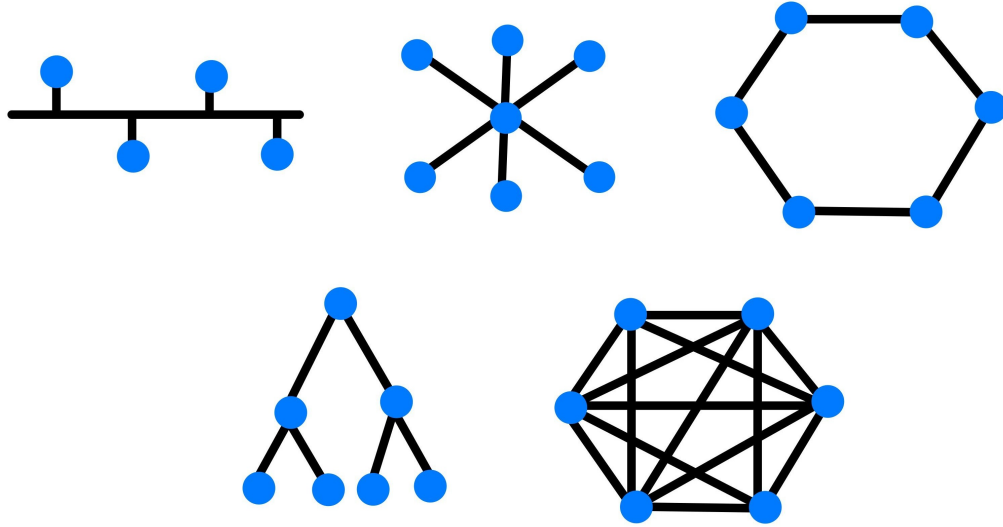
6.3. Tecnologías de comunicación

6.3.1. ESP-NOW

ESP-NOW es un protocolo de comunicación inalámbrica desarrollado por *Espressif Systems*, diseñado para los microcontroladores ESP32 y ESP8266. A diferencia de wifi convencional, no requiere un punto de acceso ni un proceso de *handshake*, lo que permite conexiones rápidas, de bajo consumo energético y sin infraestructura adicional [11].

El protocolo ESP-NOW opera en la banda ISM de 2.4 GHz y permite transmitir paquetes de hasta 250 bytes, soportando tanto comunicación unidireccional (*one-to-many*, *many-to-one*) como bidireccional entre pares. Sus modos de transmisión incluyen *unicast*, *multicast*

Figura 1. Ejemplos de topologías de red



Nota. En la parte superior izquierda se observa la topología en bus, en la parte superior central la estrella, en la parte superior derecha el anillo, en la parte inferior izquierda el árbol y en la parte inferior derecha la malla. Elaboración propia.

y *broadcast* [11] [12]. En términos de alcance, se han reportado distancias de hasta 90 m en exteriores y aproximadamente 15 m en entornos interiores, condicionadas por la presencia de obstáculos.

6.3.2. Nordic NRF24L01

El NRF24L01 es un transceptor inalámbrico de 2.4 GHz desarrollado por *Nordic Semiconductor*, utilizado ampliamente en sistemas de comunicación de corto y mediano alcance debido a su bajo consumo energético, costo reducido y robustez en la transmisión de datos [13].

Este ofrece tasas de transmisión configurables de 250 kbps, 1 Mbps y 2 Mbps. Su potencia de salida es programable entre 0 dBm y -18 dBm, mientras que la sensibilidad de recepción alcanza -85 dBm a 1 Mbps y -82 dBm a 2 Mbps, permitiendo mantener enlaces confiables en aplicaciones de corto y mediano alcance [14]. En cuanto a cobertura, su alcance típico es de aproximadamente 75 metros en interiores (limitado por la presencia de obstáculos como muros de concreto) pudiendo extenderse hasta 1 km en línea de vista (*Line Of Sight, LOS*) cuando se emplean antenas externas y tasas de datos bajas (250 kbps) [15] [13]. Además, el módulo soporta diversas topologías de comunicación, entre ellas la estrella [15], árbol y malla [13].

6.3.3. LoRa

LoRa (*Long Range*) es una tecnología de comunicación inalámbrica desarrollada por *Semtech*, basada en modulación *Chirp Spread Spectrum*, (CSS). Esta modulación permite alcanzar altas sensibilidades de recepción y enlaces de varios kilómetros con bajo consumo energético [16]. Su baja tasa de transmisión (0.3–50 kbps) está optimizada para sensores que envían datos periódicos de baja carga [17].

En su forma más simple, LoRa se implementa en una topología estrella, donde los nodos finales transmiten directamente hacia un *gateway*. Generando dos restricciones:

- *Single-hop*: la comunicación se limita a un único salto, lo que dificulta cubrir áreas extensas o con obstáculos.
- Dependencia del *gateway*: si un nodo no alcanza al *gateway*, queda fuera de la red [18].

Para superar estas limitaciones, se han desarrollado protocolos de enrutamiento que permiten operar LoRa en topologías de malla. Adaptaciones de protocolos como HWMP y AODV han demostrado ser viables en entornos agrícolas y urbanos, extendiendo la cobertura mediante comunicación *multi-hop* y aumentando la tolerancia a fallos [18].

Se reporta que los enlaces pueden extenderse hasta 22 km en condiciones de propagación en línea de vista sobre superficies marinas son plenamente factibles para cualquier combinación de parámetros de LoRa y en ambas bandas ISM (868 MHz y 434 MHz). Asimismo, es posible establecer comunicación en trayectos obstruidos de hasta 28 km en la banda de 434 MHz, aunque en este caso se requiere el uso de antenas voluminosas y de mayor costo [19].

6.4. Teoría algebraica de grafos aplicada a redes de sensores inalámbricos

La teoría de grafos proporciona un marco matemático para modelar redes de comunicación, incluidas las redes de sensores inalámbricos. Formalmente, un grafo se define como $G = (V, E)$, donde V es el conjunto de vértices y E el conjunto de aristas que representan relaciones o enlaces entre nodos [20]. En la Figura 2 se observan ejemplos de los dos tipos de grafos a tratar.

La conexidad evalúa la capacidad de la red para mantener comunicación entre nodos [21].

6.4.1. Grafos no dirigidos

Un grafo no dirigido está compuesto por aristas del tipo $\{v_i, v_j\}$, que representan relaciones bidireccionales. Son utilizados para modelar enlaces simétricos donde la comunicación es posible en ambas direcciones [22]. En grafos no dirigidos, el grafo es conexo si existe al menos un camino entre cada par de vértices [21].

El análisis algebraico de grafos utiliza representaciones matriciales que permiten estudiar sus propiedades estructurales [21].

Matriz de adyacencia

La matriz de adyacencia $A \in \mathbb{R}^{n \times n}$ se define como:

$$A_{ij} = \begin{cases} 1, & \text{si existe arista entre } v_i \text{ y } v_j, \\ 0, & \text{en otro caso.} \end{cases}$$

En grafos no dirigidos, A es simétrica [22].

Matriz de grado

La matriz de grado D es una matriz diagonal cuyas entradas son:

$$D_{ii} = \deg(v_i) = \sum_j A_{ij}$$

Esta matriz representa a la cantidad de vecinos que tiene cada grafo [22].

Matriz laplaciana

La laplaciana de un grafo no dirigido se define como:

$$L = D - A$$

La matriz L es simétrica y semidefinida positiva. El número de valores propios nulos de L corresponde al número de componentes conexas del grafo [21].

El valor de *Fiedler* corresponde al segundo autovalor más pequeño λ_2 de la matriz laplaciana L . Este valor, también denominado conectividad algebraica [21], cumple que:

- $\lambda_2 > 0 \iff$ el grafo es conexo.
- Cuanto mayor es λ_2 , mayor es la robustez estructural del grafo frente a fallos y cortes de enlaces.

6.4.2. Grafos dirigidos

Un grafo dirigido (dígrafo) cada arco es un par ordenado (v_i, v_j) , indicando que la relación existe únicamente en la dirección $v_i \rightarrow v_j$ [22].

En grafos dirigidos, la conexidad se distingue entre:

- Conexidad fuerte: para todo par (u, v) existen caminos $u \rightarrow v$ y $v \rightarrow u$.
- Conexidad débil: el grafo subyacente (ignorando las direcciones) es conexo [23].

Matriz de incidencia

Para un dígrafo con m arcos, la matriz de incidencia $B \in \mathbb{R}^{n \times m}$ se define como:

$$B_{pk} = \begin{cases} -1, & \text{si } v_p \text{ es el origen del arco } e_k, \\ +1, & \text{si } v_p \text{ es el destino del arco } e_k, \\ 0, & \text{en otro caso.} \end{cases}$$

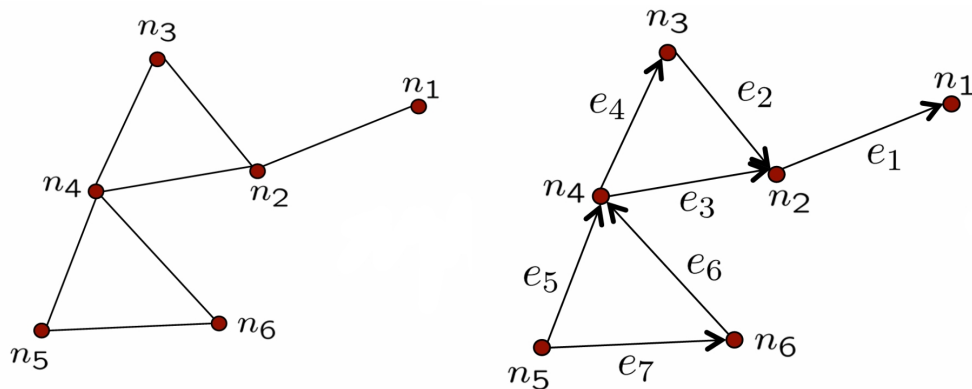
En grafos no dirigidos, orientando arbitrariamente cada arista, se cumple $L = BB^T$ [22] [20] [21].

6.4.3. Algoritmo de Kosaraju

El algoritmo de Kosaraju permite encontrar las componentes fuertemente conexas (SCC) en grafos dirigidos con complejidad $\mathcal{O}(n + m)$ de la siguiente manera [24]:

1. Realiza un recorrido de búsqueda en profundidad (*Depth-First Search*, DFS) sobre G , registrando el orden de finalización de cada vértice.
2. Calcula el grafo transpuesto G^T invirtiendo todas las aristas.
3. Ejecuta DFS sobre G^T en el orden inverso de finalización. Cada árbol DFS obtenido corresponde a una SCC .

Figura 2. Ejemplos de grafos no dirigidos y grafos dirigidos



Nota. En la parte izquierda se observa un ejemplo de grafos dirigidos y en la parte derecha un ejemplo de grafo dirigido [22].

6.4.4. Grafo de visibilidad

Un grafo de visibilidad es una representación en la que cada elemento de un conjunto se convierte en un vértice, y dos vértices están conectados solo si sus entidades son mutuamente visibles. En este contexto, visibilidad significa que el segmento que une a ambos puntos no es bloqueado por ningún obstáculo [25].

6.4.5. Grafo de visibilidad de disco unitario

Un grafo de visibilidad de disco unitario es una variante del grafo de visibilidad que incorpora una limitación de distancia. En este modelo, dos vértices se conectan únicamente si sus entidades son mutuamente visibles y además están lo suficientemente cerca. Esto significa que, aunque no exista ningún obstáculo entre ellas, dos entidades no se consideran visibles si la distancia entre ambas es mayor que una unidad [25].

Desarrollo teórico y experimental de la topología para grafos no dirigidos y dirigidos

El presente capítulo abordó el análisis estructural de la red utilizando herramientas de teoría algebraica de grafos con el objetivo de evaluar su robustez y tolerancia a fallos. Se estudiaron distintas configuraciones de grafos para identificar nodos críticos, rutas alternativas y condiciones bajo las cuales la red podía fragmentarse ante la pérdida de elementos. Este análisis permitió cuantificar la conectividad mediante el valor de Fiedler, evaluar escenarios de eliminación de nodos y determinar qué topologías ofrecían un comportamiento más estable.

7.1. Construcción inicial para grafos no dirigidos

El punto de partida del trabajo consistió en revisar los conceptos básicos de teoría de grafos aplicados a redes de sensores inalámbricas. Se continuó con la elaboración de un grafo preliminar de manera manual. Para ello se utilizó la herramienta en línea *FreeMapTools* [26], que permite medir distancias geográficas sobre un mapa. A partir de la visualización del lago de Atitlán, se midieron las distancias aproximadas entre distintos puntos del área de estudio, lo que permitió ubicar nodos representativos y establecer las conexiones iniciales entre ellos usando criterios de proximidad. Los nodos fueron ubicados en áreas estratégicas alrededor del lago, como muelles, zonas frecuentadas para actividades recreativas y puntos intermedios, con el propósito de asegurar un monitoreo representativo de la calidad del agua en diferentes sectores.

La distribución real de los nodos en el terreno como de la red formada entre ellos se puede observar en la Figura 3.

Figura 3. Grafo preliminar para el lago de Atitlán



Nota. A la izquierda se presenta el grafo preliminar construido en la herramienta en línea con las distancias entre cada nodo [26]. Mientras que a la derecha se muestra el grafo resultante con las conexiones definidas, elaboración propia.

7.2. Análisis manual de grafos no dirigidos

Con el grafo preliminar definido, se procedió a realizar un análisis manual de sus características. El propósito de esto fue comprender de manera práctica cómo se representan y se manipulan las relaciones entre nodos en una red por medio de matrices.

7.2.1. Construcción de matrices

En primer lugar, se construyó la matriz de adyacencia (A), que registra las conexiones entre pares de nodos, asignando el valor de 1 cuando existe un enlace y 0 en caso contrario. A partir de ella, se obtuvo la matriz de grado (Δ), donde cada valor en la diagonal corresponde al número de vecinos asociados a un nodo. Finalmente, se calculó la matriz laplaciana (L) como la diferencia entre la matriz de grado y la de adyacencia ($L = \Delta - A$).

Red completa

Para la red completa se obtuvieron las matrices de adyacencia, grado y laplaciana, mostradas en las ecuaciones (1) y (2).

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad \Delta = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (1)$$

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \quad (2)$$

En la matriz de grado puede observarse que el nodo 3 es el más conectado de la red, con un total de cuatro vecinos, mientras que otros nodos poseen únicamente dos conexiones.

Eliminación de nodos

En el análisis de la red completa se observó que el nodo 3 era el que concentraba la mayor cantidad de conexiones, lo cual lo convertía en un punto crítico dentro de la topología. Por esta razón, se decidió presentar de forma detallada el caso de su eliminación.

Para este caso se obtuvieron las matrices mostradas en las ecuaciones (3) y (4).

$$A_{-v_3} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad \Delta_{-v_3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (3)$$

$$L_{-v_3} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \quad (4)$$

Los demás casos de desconexión de nodos individuales se incluyen en el Anexo 12.1, donde se muestran las correspondientes matrices laplacianas.

7.3. Implementación de grafos no dirigidos en MATLAB

Dado que los cálculos manuales se volvían poco prácticos por la cantidad de nodos, se desarrolló un código en la herramienta *MATLAB* para automatizar el procedimiento. El mismo se puede encontrar en el repositorio del Anexo 12.2.

El programa permitió:

- Definir la matriz de adyacencia inicial.
- Eliminar uno o varios nodos y recalcular la matriz laplaciana correspondiente.

- Obtener el polinomio característico calculado por medio de la determinante de la matriz

$$p(\lambda) = \det(L - \lambda I)$$

- Determinar el valor de *Fiedler* de cada caso para evaluar la conectividad de la red.

Para el caso de la eliminación del nodo 3 en el grafo no dirigido preliminar se obtuvo un valor de *Fiedler* de 0.19806. Los valores de *Fiedler* para la eliminación individual del resto de nodos se puede observar en el Anexo 12.1.

7.3.1. Evolución de scripts para grafos no dirigidos

Con el fin de automatizar y escalar el análisis, se desarrollaron tres versiones de *scripts* en *MATLAB*, cada una ampliando capacidades respecto a la anterior:

Versión 1: *Script* básico

Permitió definir la matriz de adyacencia A , eliminar un conjunto de nodos y calcular la matriz de grado Δ , la laplaciana $L = \Delta - A$, el polinomio característico $p(\lambda) = \det(L - \lambda I)$ y el valor de *Fiedler*.

Versión 2: *Script* con eliminación de nodos parametrizable

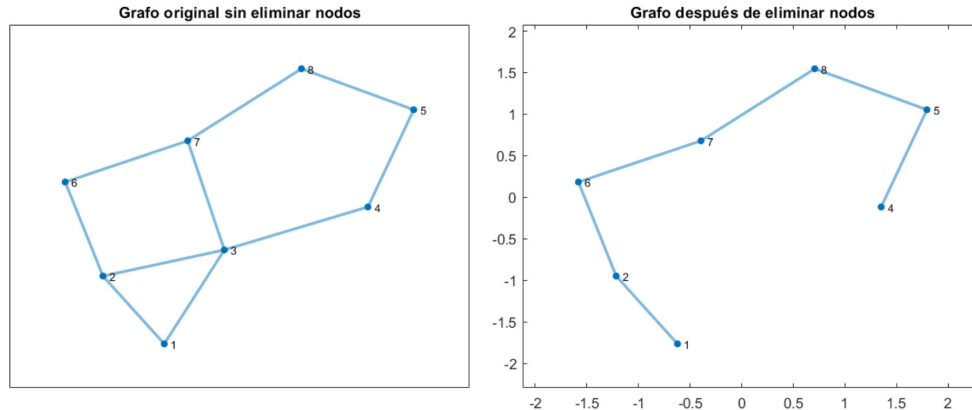
En este *script* se eliminó la necesidad de editar el código fuente. Se solicita por consola los nodos a eliminar, ejecutando automáticamente el cálculo de A , Δ , L y $p(\lambda)$. Adicionalmente se grafica el grafo orginal (red completa) y el grafo reducido (después de la eliminación de nodos) manteniendo la posición de cada nodo para facilitar la comparación visual. Por ejemplo al eliminar el nodo 3 de la red, las gráficas resultantes son las que se observan en la Figura 4.

Versión 3: *Script* para evaluación de todos los casos

Con el objetivo de evaluar la resiliencia de la red, se desarrolló una tercera versión del código que analiza todas las combinaciones posibles de eliminación de nodos hasta un límite $n - 1$ donde n es el número de nodos en el grafo. Para cada subgrafo resultante, se calcula el laplaciano L y se obtiene el valor de *Fiedler* (λ_2). Los resultados se organizan automáticamente en archivos `.xlsx/.csv`, lo que permite identificar los escenarios en donde el grafo deja de ser conexo y los escenarios más críticos (menores valores de λ_2). Además, el *script* genera resúmenes de los resultados en consola, mostrando los casos más débiles y más robustos en términos de conectividad, junto con la cantidad de casos conexos y no conexos.

Estos *scripts* pueden encontrarse en el repositorio del Anexo 12.2.

Figura 4. Grafo original y grafo reducido tras la eliminación del nodo 3



Nota. A la izquierda se presenta el grafo original con todos los nodos de la red, mientras que a la derecha se muestra el grafo reducido obtenido después de la eliminación del nodo 3. Elaboración propia.

7.4. Análisis de casos de desconexión para grafo preliminar no dirigido

En el análisis realizado con la tercera versión del código, se evaluaron un total de 255 combinaciones posibles de escenarios de desconexión, correspondientes a todas las formas de eliminar entre 0 y $n - 1$ nodos en un grafo de $n = 8$. Sin embargo, dentro de este conjunto se incluyen 8 casos en los que se eliminan exactamente 7 nodos, quedando únicamente un vértice aislado. En tales situaciones, la matriz laplaciana se reduce a una dimensión 1×1 , cuyo único autovalor es cero. Esto implica que no existe un segundo autovalor definido (λ_2), por lo que la conectividad algebraica no puede calcularse. En consecuencia, dichos casos se reportan como inválidos (con valores nulos) y el número de escenarios “válidos” con al menos dos nodos es de 247.

De estos 247 escenarios válidos, 100 casos (40.49 %) resultaron conexos, mientras que 147 casos (59.51 %) no mantuvieron conectividad lo que indica una alta vulnerabilidad de la red frente a fallos múltiples.

Adicionalmente, analizando los escenarios de desconexión más críticos, se encontró que el caso más crítico correspondió a la eliminación del nodo 3, que presentó un valor de $\lambda_2 = 0.19806$, confirmando la importancia de este nodo en la red por la cantidad de nodos que tiene.

Por otro lado, se analizaron también los escenarios de mayor robustez (valores más altos de λ_2). Estos escenarios correspondieron a configuraciones con un número reducido de nodos remanentes. Entre ellos destacan la eliminación de $\{4, 5, 6, 7, 8\}$, que dejó únicamente tres nodos conectados con $\lambda_2 = 3$.

El resumen de los resultados obtenidos en *MATLAB* se encuentra en el Anexo 12.3.

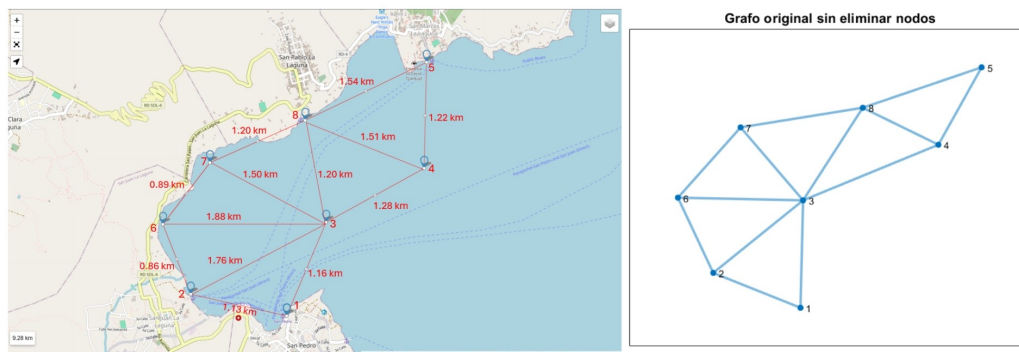
7.5. Iteración del grafo no dirigido, aplicación de triangulación

Debido a que en el análisis preliminar se observaron más casos de grafos desconectados que conexos, se consideró necesario modificar la configuración de la red. Para ello se implementó un proceso de triangulación, que consistió en añadir aristas adicionales con el fin de incrementar la robustez del grafo sin llegar a saturarlo mediante una conectividad completa entre todos los nodos. La lógica de este procedimiento fue asegurar que cada nodo contara con más de un vecino, reduciendo así la probabilidad de que una única desconexión provocara la pérdida de conectividad global.

Asimismo, se buscó acortar la longitud de los caminos entre pares de nodos, ya que en el grafo original, aunque todos los nodos tenían al menos dos vecinos algunos trayectos resultaban excesivamente largos.

Como resultado de este ajuste, se generó una segunda versión del grafo con triangulaciones adicionales. En la Figura 5 se muestra, a la izquierda, la proyección de los nodos sobre el mapa del lago de Atitlán, mientras que a la derecha se presenta el grafo resultante tras aplicar las triangulaciones adicionales.

Figura 5. Proyección geográfica y grafo triangulado



Nota. A la izquierda se observa la distribución espacial de los nodos en el lago de Atitlán [26]. Mientras que a la derecha se muestra el grafo modificado mediante triangulaciones, elaboración propia.

7.6. Análisis de casos de desconexión para grafo triangulado no dirigido

Tras la aplicación de triangulaciones al grafo preliminar, se repitió el barrido de escenarios de desconexión utilizando la tercera versión del código. En este caso igualmente se evaluaron los 247 escenarios válidos. De este conjunto, 137 escenarios (55.47%) resultaron conexos, mientras que 110 escenarios (44.53%) no mantuvieron conectividad. En comparación con el grafo preliminar, el número de configuraciones conexas aumentó de manera significativa, mejorando la robustez de la red.

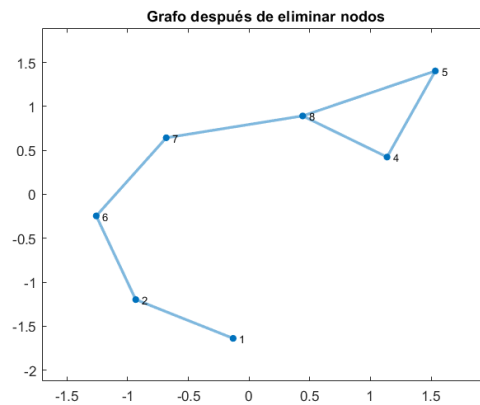
En los escenarios más críticos identificados para el grafo triangulado, el nodo 3 aparece

de forma recurrente como el elemento cuya eliminación provoca la mayor reducción de la conectividad algebraica. El caso más sensible fue la desconexión individual del nodo 3, con un valor de $\lambda_2 = 0.225$, este aumentó respecto al del grafo preliminar, lo que indica una menor probabilidad de desconexión. Otros escenarios críticos analizados fueron $\{3, 4\}$ y $\{3, 5\}$, ambos con $\lambda_2 = 0.26795$.

En cuanto a los escenarios de mayor robustez, se observó que, al igual que en el grafo preliminar, correspondieron a subgrafos con un número reducido de nodos, alcanzando nuevamente $\lambda_2 = 3$.

La gráfica de la red sin el nodo 3 puede observarse en la Figura 6. El resumen de los resultados obtenidos en *MATLAB*, se encuentra en el Anexo 12.3.

Figura 6. Grafo resultante tras la eliminación de nodo 3



Nota. Representación gráfica del sistema después de la eliminación del nodo 3 en la red original. El grafo conserva su estructura de conectividad parcial, manteniendo enlaces entre los nodos remanentes. Elaboración propia.

7.7. Construcción inicial para grafos dirigidos

A partir de la configuración final del grafo no dirigido (versión triangulada) manteniendo el mismo conjunto de ocho nodos, se definieron direcciones sobre un subconjunto de aristas con el objetivo de construir un grafo dirigido. La orientación de los enlaces se representó mediante una matriz de incidencia.

La matriz de incidencia obtenida para el grafo dirigido es:

$$B = \begin{bmatrix} -1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & +1 & 0 \\ +1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & 0 & 0 & +1 \\ 0 & -1 & +1 & 0 & +1 & -1 & 0 & +1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & +1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 & +1 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & +1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \end{bmatrix} \quad (5)$$

7.8. Implementación de algoritmo de Kosaraju en *MATLAB* para análisis de grafos dirigidos

El punto de partida fue la implementación del algoritmo de Kosaraju en lenguaje C++ [24]. Dicho código se utilizó como referencia para la lógica del procedimiento, y sobre esta base se elaboró una adaptación a *MATLAB*.

7.8.1. Evolución de scripts para grafos dirigidos

Versión 1: Adaptación inicial

La primera versión del código corresponde a la adaptación del algoritmo de Kosaraju a *MATLAB*. En esta etapa no se empezaba a partir de una matriz de incidencia, sino que los vecinos de cada nodo se describían de forma manual dentro del programa. Con esta lista de vecinos, el código ejecutaba recorridos en profundidad (*DFS*) para verificar si todos los nodos eran alcanzables desde cualquier punto. Esta primera versión permitió comprobar la validez del algoritmo de GeeksforGeeks en *MATLAB* y validar su funcionamiento.

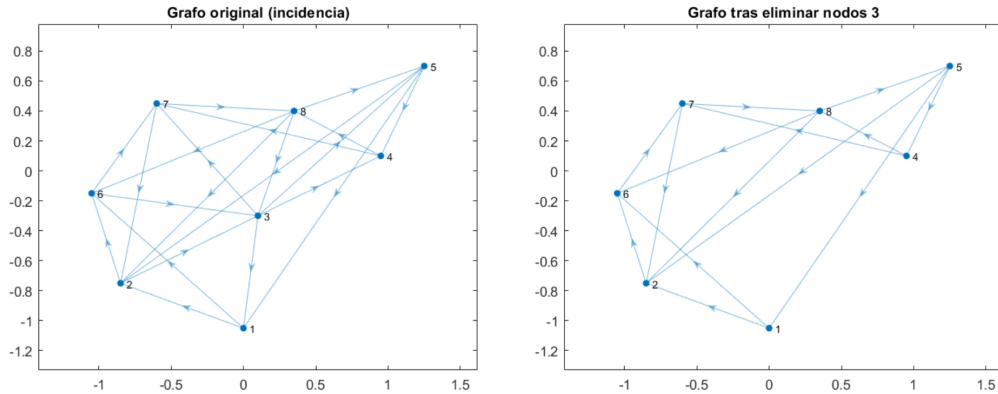
Versión 2: *Script* con implementación de matriz de incidencia y eliminación de nodos parametrizable

En esta segunda versión, el grafo dirigido se describe mediante su matriz de incidencia B . A partir de B , se construyen las aristas (s, t) y se genera el grafo dirigido. El conjunto de nodos a eliminar se ingresa de forma parametrizable en la terminal. El *script* filtra las aristas que permanecen dentro del conjunto de nodos remanentes y remapea índices para la representación gráfica del nuevo grafo, mostrando de igual manera el grafo dirigido original.

La verificación de conectividad fuerte del *script* se basa en verificar si, tras eliminar determinados nodos, el grafo reducido mantiene la propiedad de que todos los nodos pueden alcanzarse mutuamente a través de trayectorias dirigidas. Para ello, una vez generado el subgrafo con los nodos restantes, se recorre su estructura y se identifican sus componentes. Si el conjunto de nodos forma una sola componente fuertemente conexa, el subgrafo se clasifica como fuertemente conexo, caso contrario, se considera no conexo.

Por ejemplo, tras la eliminación del nodo 3 de la red, las gráficas resultantes son las observadas en la Figura 7.

Figura 7. Grafo dirigido original y grafo reducido sin el nodo 3



Nota. A la izquierda se presenta el grafo dirigido original, mientras que a la derecha se observa el subgrafo resultante tras la eliminación del nodo 3 mediante la matriz de incidencia. Elaboración propia.

Versión 3: *Script* para evaluación de todos los casos

En la tercera versión, el algoritmo recorre todas las combinaciones posibles de eliminación de nodos hasta $n - 1$ siendo un total de 247 casos, construye el subgrafo resultante y evalúa si este se mantiene fuertemente conexo.

Los resultados se consolidan en una tabla que incluye, para cada escenario: el conjunto de nodos eliminados, el número de nodos remanentes, la cantidad de aristas, el estado de fuerte conectividad y el número de componentes fuertemente conexas, que puede ser observada en el Cuadro 1. Además, el *script* genera resúmenes automáticos en consola (por número de nodos eliminados y por tamaño del subgrafo) y exporta la información completa a archivos en formato `.xlsx` y `.csv` para facilitar su posterior análisis.

Cuadro 1. Resumen por tamaño de subgrafo (*Remaining*)

Remaining	Total de casos	SC	No-SC
2	28	0	28
3	56	9	47
4	70	16	54
5	56	22	34
6	28	20	8
7	8	8	0

Nota. El cuadro resume los resultados del análisis de fuerte conectividad para el grafo dirigido, agrupados según el número de nodos remanentes en cada subgrafo. Se observa que, a medida que aumenta el tamaño del subgrafo, incrementa también la probabilidad de mantener fuerte conectividad.

Estos *scripts* pueden encontrarse en el repositorio del Anexo 12.5, en conjunto con el archivo de Excel de evaluación de todos los casos.

7.8.2. Análisis de casos de desconexión para grafo dirigido

A partir del barrido completo de combinaciones de eliminación de nodos, se evaluaron en total 247 escenarios. Los resultados globales muestran que únicamente 76 subgrafos (30.77 %) se mantuvieron fuertemente conexos, mientras que la mayoría, 171 subgrafos (69.23 %), perdieron esta propiedad. Demostrando que la red dirigida presenta una alta vulnerabilidad estructural frente a la eliminación de nodos.

De manera complementaria, el análisis por tamaño de subgrafo indica que los subgrafos con mayor número de nodos presentaron una fracción reducida de casos fuertemente conexos, mientras que al disminuir el número de nodos, la conectividad fuerte desaparece casi por completo. Estos resultados, se encuentran sintetizados en el Anexo 12.6.

7.9. Comparación entre tipos de grafos y relación con topologías de comunicación

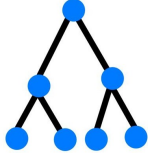
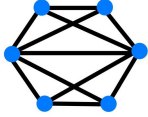
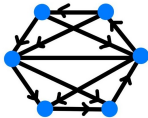
Realizando un análisis comparativo entre ambos tipos de grafos, se observa que los grafos no dirigidos ofrecen una mayor tolerancia a la eliminación de nodos. Esto se debe a que la conectividad no depende de la dirección de los enlaces, sino únicamente de su existencia, lo que permite que entre dos nodos existan múltiples trayectorias alternas. Gracias a esta característica la red puede absorber fallos sin perder inmediatamente la conectividad global. No obstante, para mantener dicha conectividad en escenarios de fallas múltiples fue necesario reforzar la topología mediante triangulaciones, incrementando así la redundancia estructural.

En contraste, los grafos dirigidos mostraron una vulnerabilidad más marcada. La fuerte conectividad exige que para cada par de nodos exista no solo un camino de ida, sino también uno de regreso. Esta condición más estricta hace que la probabilidad de perder conectividad

sea considerablemente mayor. En los escenarios analizados, más del 69% de los subgrafos perdió esta propiedad. Como consecuencia, aún con la misma cantidad de nodos, los grafos dirigidos presentan un menor nivel de robustez frente a fallos.

Adicionalmente, los grafos trabajados se comparan con topologías de red, con el fin de identificar similitudes estructurales y evaluar las ventajas y limitaciones de cada representación. Esta relación se sintetiza en el Cuadro 2.

Cuadro 2. Comparación entre grafos analizados y topologías clásicas de comunicación

Grafo analizado	Topología	Características principales	Figura
Grafo no dirigido preliminar	Árbol expandido	Presenta rutas claras entre nodos, pero algunos trayectos son largos y dependen de nodos críticos, aumentando la vulnerabilidad a fallos.	
Grafo no dirigido triangulado	Malla parcial	Cada nodo cuenta con múltiples conexiones alternativas sin llegar a una conectividad completa, lo que mejora la tolerancia a fallos sin saturar la red.	
Grafo dirigido	Malla dirigida	Representa escenarios con comunicación unidireccional, donde los caminos de ida no garantizan trayectorias de regreso. Es más vulnerable a fallos múltiples.	

La comparación realizada en el Cuadro 2 evidencia que los grafos no dirigidos, particularmente en su versión triangulada, presentan mayor correspondencia con estructuras de malla parcial, lo que favorece la tolerancia a fallos. Por otro lado, el grafo dirigido es una representación de redes en mallas dirigidas en un solo sentido, resultando más vulnerables a la pérdida de conectividad.

Dado que en las aplicaciones de redes de sensores para monitoreo remoto, uno de los objetivos principales es garantizar la tolerancia a fallos, los grafos no dirigidos constituyen una opción más adecuada para este tipo de sistemas. Por lo que siempre que las tecnologías de comunicación seleccionadas lo permitan, se procurará trabajar con representaciones no dirigidas para maximizar la robustez de la red. Más adelante se revisan tecnologías de comunicación, comparando sus propiedades, así como las topologías que cada una permite implementar, con el fin de identificar qué tecnologías ofrecen el mejor desempeño en una red tolerante a fallos.

En este capítulo se describió la evaluación práctica de las tecnologías de comunicación consideradas para la red de sensores. A partir de los requerimientos estructurales identificados en el análisis de grafos, se estudiaron las características de nRF24L01, ESP-NOW y LoRa, con el fin de determinar su idoneidad para redes distribuidas y escenarios de monitoreo remoto.

Además, tras este análisis se realizaron pruebas preliminares con ESP-NOW para evaluar su desempeño en distancias cortas y validar los resultados obtenidos en el capítulo anterior. Posteriormente, se empleó LoRa para desarrollar una aplicación de largo alcance integrando el grafo de visibilidad.

8.1. Comparación entre tecnologías de comunicación

Antes de seleccionar las tecnologías utilizadas en las pruebas experimentales, fue necesario analizar las alternativas de comunicación inalámbrica que pudieran adaptarse a los requerimientos identificados en el análisis de grafos. En particular, se consideraron criterios como el alcance, la latencia, el consumo energético y la topología soportada, dado que cada uno influye directamente en la capacidad de la red para mantener conectividad aún ante fallas. Con este propósito, se evaluaron tres tecnologías: ESP-NOW, nRF24L01 y LoRa, cuyos aspectos más relevantes se resumen en el Cuadro 3.

Cuadro 3. Comparación entre tecnologías de comunicación

Tecnología	Topología / Modo de comunicación	Características principales
ESP-NOW	No requiere punto de acceso ni infraestructura adicional. Soporta <i>unicast</i> , <i>multicast</i> y <i>broadcast</i> .	Conexiones rápidas y de bajo consumo. Alcance limitado: ~ 90 m en exteriores y 15 m en interiores. Opera en 2.4 GHz, vulnerable a interferencias. Escalabilidad reducida (número limitado de pares registrados).
nRF24L01	Soporta topologías estrella, árbol y malla.	Bajo costo y consumo energético reducido. Alcance de hasta 1 km en LOS con antenas externas. Alcance limitado en interiores (~ 75 m). Mayor degradación en entornos con obstáculos.
LoRa	Posibilidad de multi-hop mediante protocolos de enrutamiento. Topología básica de estrella con dependencia del <i>gateway</i> .	Gran alcance (hasta 22 km en LOS, 28 km con antenas de alta ganancia). Alta sensibilidad de recepción y bajo consumo. Baja tasa de transmisión (0.3–50 kbps). Mayor latencia que ESP-NOW o nRF24L01.

La comparación presentada en el Cuadro 3 muestra que cada tecnología ofrece distintas ventajas en términos de alcance, consumo energético, latencia y el tipo de topología que permite implementar. En el análisis de corto alcance, tanto nRF24L01 como ESP-NOW fueron evaluados considerando estas propiedades. Aunque nRF24L01 ofrece flexibilidad y configuraciones variadas de transmisión, ESP-NOW resultó más adecuado debido a su compatibilidad nativa con los módulos ESP32, lo que simplificó la implementación.

Por otro lado, LoRa destacó por su gran alcance y su comportamiento estable en entornos con interferencias, lo que la hizo adecuada para escenarios donde se requiere cubrir distancias largas con baja carga de datos.

Con base en este análisis, se eligió trabajar experimentalmente con ESP-NOW para validar el comportamiento derivado del modelo matemático del grafo no dirigido triangulado en redes de corto alcance. Posteriormente, se utilizó LoRa para evaluar la propagación de mensajes en escenarios de largo alcance, consistentes con el grafo de visibilidad.

8.2. Entorno de desarrollo ESP-NOW

Para la implementación del sistema se utilizó el microcontrolador ESP32 DEVKIT V1. Como entorno principal se empleó *Visual Studio Code*. Sobre este se instaló el *plugin PlatformIO*, el cual facilita la administración de dependencias, compilación y carga de programas en microcontroladores.

El *framework* seleccionado para la programación de los nodos fue ESP-IDF (*Espressif IoT Development Framework*), ya que provee acceso directo a las funciones nativas de los microcontroladores ESP32, incluyendo el soporte para la tecnología de comunicación ESP-NOW.

8.3. Estructura del proyecto para pruebas de latencia ESP-NOW

- `types.h`: se define la estructura `paquete_t` con los campos {origen, destino, mensaje, ttl, id_mensaje}, que soportan el reenvío *multi-hop* y la deduplicación de mensajes.
- `config_nodes.h` / `config_nodes.c`: configuración de cada nodo mediante la macro `NODE_ID`. Contienen la tabla de vecinos (MACs e IDs) y el número de conexiones, reflejando la topología de la red. La tabla de vecinos permite que cada nodo conozca a quién puede enviar mensajes directamente, facilitando el reenvío *multi-hop*.
- `comms.h` / `comms.c`: capa de comunicación con ESP-NOW. Incluye registro de vecinos, envío a todos los vecinos (`send_to_all_neighbors`) y recepción con deduplicación por `id_mensaje`. Implementa el reenvío cuando `ttl > 1` (*Time To Live*) que corresponde al número de saltos restantes que un mensaje puede realizar dentro de la red. Así como el uso de *backoff* para mitigar colisiones.
- `test_latencia.h` / `test_latencia.c`: lógica de prueba de latencia ida-vuelta. El emisor envía un mensaje de prueba, el receptor responde con un ACK, y el emisor calcula la latencia de ida y vuelta con `esp_timer_get_time()`.
- `main.c`: Inicializa *ESP-NOW*, carga la configuración de vecinos, prepara GPIOs de depuración y lanza la tarea de pruebas de latencia.

Este código puede encontrarse en el repositorio del Anexo 12.7.

8.4. Flujo de mensajes *multi-hop* en *ESP-NOW*

En lo que respecta a la metodología utilizada en el flujo de mensajes de *multi-hop* se tiene:

- **Generación del paquete en el emisor:** cuando un emisor inicia la transmisión mediante la pulsación de un botón, se genera un paquete de tipo `paquete_t`, el cual contiene los campos {origen, destino, mensaje, ttl, id_mensaje}. Este paquete se difunde inicialmente a todos los vecinos del nodo.
- **Procesamiento en nodos intermedios:** al llegar a un nodo intermedio, primero se realiza un proceso de deduplicación comparando el `id_mensaje` con un *buffer* circular que almacena los últimos identificadores recibidos. Esto permite descartar paquetes repetidos y evita la formación de bucles en la red.

Si el nodo no corresponde al destino y el contador `ttl` es mayor que uno, se decrementa dicho valor y el paquete se reenvía a todos los vecinos, excluyendo al nodo del cual se

recibió. Para reducir la probabilidad de colisiones en este reenvío, se aplica un retardo dependiente del identificador del nodo ($30 + 10 * \text{NODE_ID}$).

- **Recepción en el nodo destino:** si el paquete llega al nodo destino y corresponde a una prueba de latencia, este genera y envía una respuesta de confirmación (ACK) de regreso al emisor, siguiendo la misma lógica de *multi-hop*.
- **Cálculo de latencia:** cuando el emisor recibe el ACK, calcula la latencia de ida y vuelta a partir de la marca temporal obtenida con `esp_timer_get_time()` y reporta el resultado en los registros de salida.
- **Selección del nodo destino para las pruebas:** para mantener un mayor control sobre los resultados y facilitar la comparación entre escenarios, se decidió trabajar con un nodo destino fijo. El nodo seleccionado fue el nodo 1, al encontrarse en un extremo de la red y representar un punto de referencia adecuado para evaluar el impacto de las desconexiones.

8.5. Propagación de mensajes en la red en ESP-NOW

Para complementar la explicación del flujo de mensajes con *multi-hop*, se registró la propagación de los mensajes en la red mediante el analizador lógico *Saleae Logic Pro 16* [27], observando la activación secuencial de cada nodo al retransmitir los paquetes.

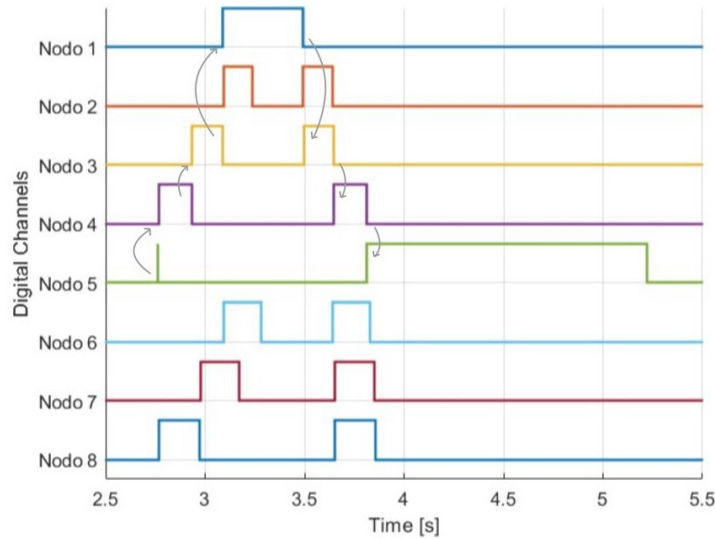
Para iniciar con las pruebas de latencia, se seleccionaron diferentes escenarios de desconexión de nodos tomando como referencia el análisis realizado sobre el grafo no dirigido triangulado con el objetivo de evaluar cómo la pérdida de nodos afecta el tiempo de propagación de los mensajes en la red. En primer lugar, se realizó la prueba sobre la red completa. Posteriormente, se analizaron siete casos individuales de desconexión, uno por cada nodo del 2 al 8.

Adicionalmente, se evaluaron tres casos considerados críticos, en los que la eliminación conjunta de ciertos nodos tenía un impacto notable sobre la conectividad global (desconexiones simultáneas de los nodos 3 y 4, y de los nodos 3 y 5). Finalmente, se incluyeron dos escenarios de interés que, a pesar de múltiples desconexiones, mantenían una cierta robustez en la comunicación (eliminación del conjunto de nodos 4, 5, 7 y 8, y de los nodos 4, 5, 6, 7 y 8). Para cada uno de los escenarios se evaluó individualmente cada nodo obteniendo el tiempo de latencia. Las capturas de cada uno de estos escenarios pueden encontrarse en el repositorio del Anexo 12.7.

Cuando el mensaje enviado por el emisor llega al receptor, este enciende su LED como confirmación de recepción. Posteriormente, al retornar la respuesta hacia el emisor, el LED del emisor parpadea indicando que el mensaje ha regresado correctamente.

En esta sección se presenta los escenarios más relevantes. En la red completa, se analizó el caso del nodo 5 enviando un mensaje al nodo 1, ya que el nodo 5 es el más lejano a 1. Como se observa en la Figura 8, el mensaje siguió un camino relativamente directo gracias a la conectividad redundante provista por las triangulaciones. En este escenario la latencia de ida fue de aproximadamente 327.47312 ms, mientras que la latencia de regreso fue de 321.5584 ms.

Figura 8. Propagación de mensaje en red completa (5-1)



Nota. Los pulsos corresponden a los GPIO de debug de cada nodo en la red completa, capturados con el analizador lógico y graficado en MATLAB. Se observa la propagación del mensaje, realizando un total de 2 saltos intermedios de ida y 2 de vuelta, señalados mediante flechas. Elaboración propia.

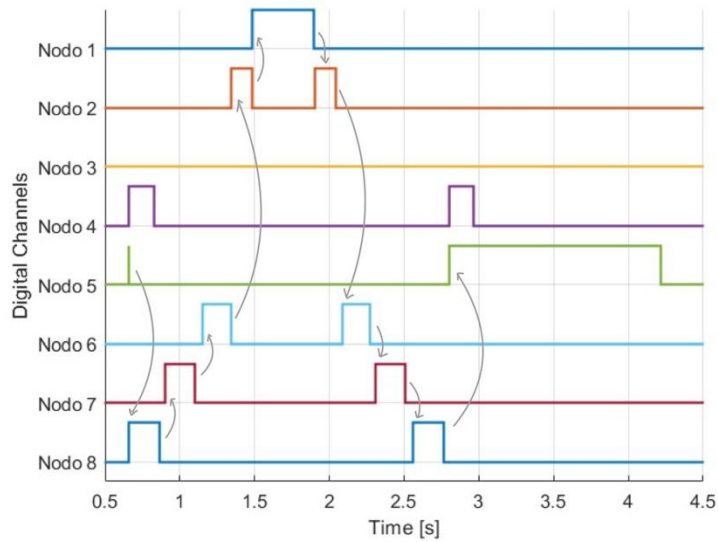
Por otro lado, en la red sin el nodo 3, el mismo escenario (mensaje del nodo 5 al nodo 1) experimenta un comportamiento diferente. Tal como se muestra en la Figura 9, el paquete se ve obligado a tomar rutas alternativas, atravesando un mayor número de nodos intermedios antes de alcanzar su destino. En este caso, la latencia de ida aumentó a 828.8352 ms, mientras que la latencia de regreso alcanzó 905.19152 ms, lo que representa un incremento notable respecto a la red completa y confirma el impacto de la desconexión de nodos críticos en el desempeño de la red.

8.6. Resultados de pruebas de latencia en ESP-NOW

En esta sección se presentan los resultados de latencia para la red completa visibles en el Cuadro 4 y para su mayor caso crítico siendo la desconexión del nodo 3 en el Cuadro 5. Este escenario fue seleccionado debido a que, según el valor de *Fiedler* en el grafo no dirigido triangulado, el nodo 3 concentra una alta cantidad de conexiones y su pérdida afecta significativamente la conectividad global.

Las demás configuraciones de prueba de latencia se detallan en el Anexo 12.8.

Figura 9. Propagación de mensaje en red sin nodo 3 (5-1)



Nota. Los pulsos corresponden a los GPIO de *debug* de cada nodo en la red con el nodo 3 faltante, capturados con el analizador lógico y graficados en *MATLAB*. Se observa la propagación del mensaje, realizando un total de 4 saltos intermedios de ida y 4 de vuelta, señalados mediante flechas. Elaboración propia.

Cuadro 4. Prueba de latencia en red completa

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	4.25648	5.30064	9.55712
3	9.01184	9.86704	18.87888
4	165.9528	168.71136	334.66416
5	327.47312	321.5584	649.03152
6	161.32176	151.61568	312.93744
7	160.4696	162.50864	322.97824
8	160.68032	169.82768	330.50800

Nota. Los valores presentados corresponden a las mediciones experimentales de latencia en la red completa utilizando ESP-NOW. Para cada nodo emisor, se registró el tiempo de ida, el tiempo de retorno del ACK y la latencia total resultante de la suma de ambos.

Cuadro 5. Prueba de latencia en red sin nodo 3

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.93136	1.44336	5.37472
3	-	-	-
4	898.3216	895.9176	1794.2392
5	828.8352	905.19152	1734.02672
6	151.21264	221.54016	372.75280
7	380.93168	407.35888	788.29056
8	626.73632	643.76400	1270.50032

Nota. Los valores presentados corresponden a las mediciones experimentales de latencia en la red sin el nodo 3 utilizando ESP-NOW. Para cada nodo emisor, se registró el tiempo de ida, el tiempo de retorno del ACK y la latencia total resultante de la suma de ambos. Se muestra un incremento significativo en la latencia para los nodos cuyo trayecto hacia el nodo 1 dependía del nodo 3.

Respecto a la red completa, las latencias totales permanecieron en el rango de 9 ms a 650 ms, con valores bajos en los nodos más cercanos al nodo 1, como el nodo 2 con 9.55 ms. En contraste, el nodo 5, que es el más alejado del nodo 1, alcanzó una latencia de 649 ms, demostrando cómo la distancia en la topología influye directamente en los tiempos de propagación.

En el caso de la red sin nodo 3, las latencias aumentaron de manera considerable, alcanzando valores de hasta 1794 ms para el nodo 4 y 1734 ms en el nodo 5. Sin embargo, se mantiene un tiempo reducido en el nodo 2 de 5.37 ms, debido a su cercanía directa con el nodo 1. Esto demostró la dificultad de mantener trayectorias estables sin el nodo 3, ya que la red entra en un escenario peligroso porque se interrumpen caminos claves y la latencia se incrementa drásticamente.

8.7. Implementación final ESP-NOW

Aunque la versión de pruebas de latencia y la implementación final de *broadcast* persiguen objetivos distintos, comparten una misma base de funcionamiento. En ambos casos se utiliza la misma estructura de mensajes `paquete_t`. La comunicación *multi-hop* siguió la misma lógica: cada nodo transmite a sus vecinos directos, decrementa el ttl, descarta duplicados y evita reenviar al nodo de origen inmediato. También se conservó la gestión de vecinos mediante archivos de configuración, la inicialización de ESP-NOW y GPIOs, y la estrategia de retardo dependiente del nodo para reducir colisiones.

Las diferencias principales aparecen en el propósito y alcance de cada implementación. En la primera versión, orientada a pruebas de latencia, se tenía un esquema rígido: el emisor era fijo, el campo `destino` apuntaba a un nodo específico (`TARGET_NODE_ID`), y únicamente ese nodo procesaba el mensaje y respondía con un ACK.

En cambio, la versión final de *broadcast* introdujo un mecanismo más robusto y representativo de una red real. En este diseño, el campo `destino` se fijó en `BROADCAST_ID` (255), indicando que todos los nodos debían procesar el mensaje. Cada nodo que recibía la difusión

generaba un ACK, y el emisor mantenía un registro completo mediante un `ack_bitmap`, lo que le permitió identificar con precisión qué nodos respondieron y cuáles no, mostrando así el estado de conectividad de la red en tiempo real. El mecanismo de reenvío también cambió, tanto los mensajes como los ACKs se propagaron por medio de *multi-hop* a toda la red, lo que aseguró que la confirmación llegara incluso a través de rutas indirectas. Además, el rol de receptor dejó de ser fijo; cualquier nodo podía funcionar como receptor cuando se presionaba el botón de otro nodo, lo que hizo al sistema más dinámico y flexible. Finalmente, la señalización con LEDs también evolucionó: pasó de ser un simple indicador de recepción de un ACK a mostrar el estado general de conectividad de toda la red.

Este código puede encontrarse en el repositorio del Anexo 12.9.

8.8. Resultados implementación final ESP-NOW

Los resultados de la implementación se registraron en las salidas de las terminales seriales, tal como se presenta en el Anexo 12.10. En la primera prueba, con todos los nodos conectados, el nodo 3 transmitió un mensaje y recibió confirmaciones de recepción ACK por parte de cada uno de los otros siete nodos de la red, evidenciando que la comunicación fue exitosa y que todos los dispositivos permanecían activos dentro de la topología.

En la segunda prueba, se desconectaron los nodos 6 y 8. Los resultados muestran que, tras la transmisión del nodo 3, el tiempo de espera para la recepción de sus correspondientes ACK se agotó sin obtener respuesta. Este comportamiento permitió concluir que dichos nodos se encontraban desconectados, validando la capacidad del sistema para detectar la pérdida de conectividad en la red.

Finalmente, en la Figura 10 se puede observar la implementación física realizada.

Figura 10. Implementación física de la red de nodos sobre mapa de prueba



Nota. La figura muestra la disposición experimental de la red de sensores inalámbricos utilizando nodos ESP32. Elaboración propia.

Tras validar el modelo matemático con ESP-NOW en escenarios de corto alcance, fue

necesario evaluar una tecnología adecuada para enlaces más extensos. Por ello, el trabajo continuó con LoRa y su correspondencia con el grafo de visibilidad.

8.9. Entorno de desarrollo LoRa

El desarrollo de los módulos de comunicación LoRa se realizó utilizando tarjetas Heltec WiFi LoRa 32 (V3) [28], las cuales integran un microcontrolador ESP32 junto con un transceptor SX1262. Esta plataforma permitió trabajar en modo LoRa punto-a-punto (P2P) sin la necesidad de *gateways* ni infraestructura de red LoRaWAN, utilizando únicamente comunicación directa entre nodos.

Para la programación se empleó el entorno *PlatformIO* sobre *Visual Studio Code* con el *framework* de Arduino y la librería *RadioLib* [29] (control del transceptor SX1262), *ArduinoJson* (serialización de mensajes) y *Adafruit SSD1306* (visualización en pantalla OLED), que facilitaron el manejo de los parámetros de modulación y la interfaz de usuario. Todas las tarjetas se configuraron a una frecuencia de 915 MHz, correspondiente al espectro libre permitido para la región [30] [31]. Los parámetros de operación incluyeron un factor de expansión (*Spreading Factor*) de 9, un ancho de banda de 125 kHz, una potencia de transmisión de 10 dBm y una tasa de codificación (*Coding Rate*) de 4/7 (4 bits de datos, se transmiten 7 bits totales).

Se integró un sistema de temporización mediante `esp_timer_get_time()` para registrar las latencias de envío y recepción. El procesamiento de mensajes y confirmaciones (ACKs) se realizó mediante estructuras personalizadas que incluyeron campos para el identificador del mensaje, el origen, el destino y el valor de TTL en el caso del modo multi-hop.

Finalmente, la relación entre nodos se estableció utilizando un grafo de visibilidad, dado que en LoRa P2P un nodo puede comunicarse únicamente con aquellos que se encuentren dentro de su rango efectivo de recepción en la misma frecuencia.

8.10. Implementación de modo *single-hop* LoRa

La primera implementación del sistema LoRa operó en modo *single-hop*, en el cual el nodo emisor transmite directamente a todos los nodos dentro de su rango mediante propagación de mensajes tipo *broadcast*. En este modo no existen retransmisiones intermedias, por lo que únicamente los nodos que reciben la señal directamente participan en la comunicación.

Para la transmisión, el nodo emisor envía un paquete JSON que incluye un identificador único de mensaje. Cada nodo que recibe este paquete genera un ACK dirigido al emisor. Sin embargo, como varios nodos responden a un mismo mensaje, fue necesario implementar un mecanismo para evitar colisiones: cada nodo programa su ACK con un retardo fijo basado en su identificador numérico. De esta forma, las respuestas no se traslapan en el canal de radio.

En este modo de operación, todos los nodos permanecían permanentemente en modo escucha (RX continuo), esperando la llegada de un mensaje. Cuando uno de ellos recibía

el mensaje, cambiaba momentáneamente a modo emisor (RX) para transmitir su ACK, y luego regresaba automáticamente al estado de recepción. Respecto al nodo emisor del mensaje, este permanece en modo recepción durante una ventana de escucha predefinida, registrando los ACKs conforme van llegando. Para cada ACK recibido, calcula la latencia restando el retardo artificial introducido por el nodo respondedor, lo que permite obtener un valor aproximado del tiempo real de propagación del mensaje. Los resultados se almacenan por nodo y se muestran en la pantalla OLED del dispositivo, permitiendo visualizar en tiempo real qué nodos respondieron y cuál fue su latencia. Adicionalmente, para facilitar la verificación visual del funcionamiento del sistema, cada nodo activaba brevemente el LED integrado de la placa cuando recibía un mensaje válido.

Este código puede encontrarse en el repositorio del Anexo 12.11.

8.11. Resultados de modo *single-hop* LoRa

La validación del modo *single-hop* se realizó inicialmente con dos nodos para asegurar la correcta transmisión de mensajes JSON, el funcionamiento del ciclo RX→TX→RX y la generación de ACKs. Posteriormente, se operó la red completa de ocho nodos ubicados a distancias cortas (<10 m).

En todos los escenarios, los nodos que se encontraban dentro del alcance del emisor recibieron el mensaje confirmándolo mediante la iluminación del LED integrado y la emisión del correspondiente ACK. Gracias al esquema de retardos escalonados basado en el identificador de nodo, no se observaron colisiones en las respuestas, incluso cuando los ocho nodos transmitieron ACKs para un mismo nodo.

En términos de desempeño, la latencia se mantuvo prácticamente constante, con valores típicos de 570 ms. Solo en casos aislados se observaron variaciones de ± 1 ms (569–571 ms), lo cual es consistente con pequeñas variaciones en el tiempo de aire y en el procesamiento interno del transceptor. Como se aprecia en la Figura 11, las mediciones de latencia mostraron una notable estabilidad.

El modo *single-hop* mostró varias limitaciones importantes. En primer lugar, la comunicación dependía por completo del alcance directo del emisor, por lo que los nodos fuera de rango quedaban completamente aislados. Además, el sistema no ofrecía rutas alternativas, de modo que cualquier obstáculo o degradación del enlace impedía la recepción del mensaje y su correspondiente ACK. Estas restricciones motivaron la adopción del modo *multi-hop* para extender la cobertura y mejorar la tolerancia a fallos.

8.12. Implementación de modo *multi-hop* LoRa

La implementación *multi-hop* amplió las capacidades del modo anterior permitiendo que los mensajes alcanzaran nodos fuera del rango directo del emisor mediante la retransmisión cooperativa entre nodos intermedios. Cada nodo actuó simultáneamente como receptor y como posible repetidor, lo que permitió superar las limitaciones de alcance físico y construir

Figura 11. Resultados de latencia para modo *single-hop*



Nota. La figura muestra la pantalla OLED del nodo 1 durante una transmisión en modo *single-hop*. En ella se observa la recepción exitosa de los ACKs provenientes de todos los nodos restantes de la red. Elaboración propia.

una red funcional basada en el grafo de visibilidad.

A diferencia del otro modo, donde la comunicación ocurría únicamente por difusión directa, el modo multi-hop utilizó un esquema de *flooding controlado*. Cada mensaje incluía un identificador único (MID), un valor de TTL y un registro del camino recorrido (*path*). Estos campos permitieron controlar el número máximo de saltos, asegurar la trazabilidad del mensaje y evitar retransmisiones innecesarias.

Para prevenir bucles, cada nodo implementó un mecanismo de deduplicación mediante un búfer circular que almacenaba los MIDs procesados recientemente. Si un mensaje ya había sido visto, era descartado inmediatamente. Además, el TTL se reducía en cada salto, descartándose mensajes que alcanzaban el límite permitido. Cuando un nodo recibía un mensaje de datos, ejecutaba una lógica simple:

- Si el nodo era el destino final, procesaba el mensaje y generaba un ACK.
- Si no lo era, evaluaba si debía retransmitirlo según las reglas de deduplicación (TTL y destino). La retransmisión incluía un pequeño retardo aleatorio para reducir la probabilidad de colisiones entre nodos que actuaban como relays simultáneamente.

Los ACKs no emplearon *flooding*. En su lugar, se utilizó *source routing*, es decir, cada ACK siguió la ruta inversa registrada en el *path* del mensaje original. Esto redujo significativamente el tráfico y garantizó que la confirmación regresara al emisor a través de un camino válido.

Para facilitar pruebas en laboratorio donde todos los nodos estaban físicamente dentro del mismo rango, se incorporó un modo adicional donde se realizó una “simulación de topología”, que forzaba a los nodos a aceptar mensajes únicamente de sus vecinos definidos.

La medición de latencia se realizó de manera similar al modo *single-hop*, compensando el retardo fijo programado en el nodo receptor.

Este código puede encontrarse en el repositorio del Anexo 12.11.

8.13. Resultados de modo *multi-hop* LoRa

Los resultados obtenidos muestran el impacto directo que tiene la retransmisión entre nodos sobre la latencia de la red. En comparación con el modo *single-hop*, donde todos los nodos alcanzables recibían el mensaje con una latencia prácticamente constante, la inclusión de saltos intermedios incrementó de manera significativa los tiempos de propagación.

En primer lugar, incluso para nodos que sí están dentro del rango directo del emisor, pero participan en un escenario *multi-hop*, la latencia promedio (838 ms) es de aproximadamente el doble respecto al *single-hop*. Este incremento surge porque, aunque el nodo está en rango, el sistema debe evaluar reglas adicionales del protocolo (deduplicación, TTL, retardo anti-colisión y procesamiento ampliado), lo que introduce sobrecarga temporal en el manejo del mensaje.

Por otro lado, para nodos que no tenían visibilidad directa y requerían un salto intermedio, la latencia promedio (1811 ms) resultó aproximadamente tres veces mayor que en el modo *single-hop*. Es decir, la latencia del nodo “con salto” es aproximadamente el doble de la latencia del nodo directo en *multi-hop*, lo cual tiene sentido porque el nodo intermedio añade una etapa completa de recepción, procesamiento y retransmisión tanto para el mensaje original como para el ACK de retorno.

Tal como se observa en la Figura 12, las latencias en modo multi-hop dejan de ser constantes. Para los nodos con conexión directa al emisor, la latencia presentó un promedio de 838 ms, con un valor mínimo de 775 ms y un máximo de 912 ms, evidenciando la variabilidad introducida por el procesamiento adicional y los retardos anti-colisión. En contraste, los nodos que requieren un salto intermedio alcanzaron latencias significativamente mayores, con valores de 1861 ms y 1760 ms.

En la Figura 13 se puede observar la implementación física realizada en el laboratorio.

8.14. Pruebas de alcance LoRa

Para evaluar el desempeño de LoRa en condiciones reales, se realizaron pruebas de alcance en dos escenarios representativos: un entorno con interferencias y un escenario de línea de vista despejada. En ambos casos se utilizaron las antenas incluidas de fábrica con las placas Heltec, las cuales presentan una ganancia limitada. Esto implica que la distancia máxima alcanzada en LOS no corresponde al alcance máximo del módulo SX1262, sino al máximo que pudo evaluarse dadas las condiciones del terreno y el equipamiento disponible.

En el escenario con interferencias, caracterizado por árboles, vehículos y variaciones del relieve, se obtuvo un alcance aproximado de 786 metros, manteniendo recepción estable y

Figura 12. Resultados de latencia para modo *multi-hop*



Nota. La figura muestra la pantalla OLED del nodo 6 durante una transmisión en modo *multi-hop*. En ella se observa la recepción exitosa de los ACKs provenientes de todos los nodos restantes de la red. Elaboración propia.

Figura 13. Implementación física de la red LoRa sobre mapa de prueba



Nota. La figura muestra la disposición experimental de la red de sensores inalámbricos utilizando Nodos LoRa. Elaboración propia

sin pérdida de mensajes. En contraste, en condiciones de línea de vista despejada, la comunicación alcanzó 1.37 kilómetros, casi el doble de distancia respecto al entorno obstruido. Esta diferencia confirma la sensibilidad de LoRa a la obstrucción de la línea de propagación, pero también demuestra su capacidad para operar en escenarios no ideales.

Las distancias registradas durante las pruebas de alcance pueden observarse en las Figuras 14 y 15, que muestran el comportamiento del enlace LoRa en entornos con interferencias y en condiciones de línea de vista despejada.

Figura 14. Resultados de pruebas de alcance en terreno con obstáculos



Nota. La figura muestra los registros obtenidos con dos nodos (transmisor y receptor) durante la prueba de alcance en un entorno con obstáculos. El nodo receptor fue desplazado a lo largo del recorrido mientras se verificaba la recepción del mensaje LoRa, alcanzando una distancia máxima aproximada de 786 metros.

Figura 15. Resultados de pruebas de alcance en terreno con línea de vista



Nota. La figura presenta los resultados de la prueba realizada con dos nodos (transmisor y receptor) en condiciones de línea de vista despejada. Debido a la disponibilidad del terreno, la distancia máxima que pudo evaluarse fue de aproximadamente 1.37 kilómetros. Dentro de este límite, el enlace LoRa se mantuvo estable durante toda la prueba en condiciones de línea de vista despejada.

- El análisis con teoría algebraica de grafos confirmó que las topologías no dirigidas con triangulación incrementan la robustez de la red. Se observó un aumento de la conectividad algebraica de 0.198 a 0.225, así como un incremento cercano al 15 % en los casos conexos (de 100 a 137), casi duplicando los valores obtenidos para el grafo dirigido (76 \rightarrow 137). Estos datos confirman que la triangulación mejora la tolerancia a fallos frente a la eliminación de nodos.
- La comparación de tecnologías de comunicación demostró que ESP-NOW ofreció un alcance hasta 20 % mayor que nRF24L01 en escenarios de corta distancia, además de contar con compatibilidad nativa con ESP32, lo que facilitó su implementación experimental. Por su parte, LoRa proporcionó el mejor desempeño en alcance, logrando distancias casi el doble en línea de vista (1.37 km) respecto a entornos con interferencias (0.786 km), junto con un menor consumo energético y alta tolerancia a fallos.
- Las validaciones físicas con ESP-NOW mostraron que, aunque la pérdida de un nodo crítico (nodo 3) incrementó la latencia de 141 ms a más del triple (513 ms), en el resto de escenarios la red mantuvo tiempos de propagación prácticamente constantes, reflejando resiliencia frente a fallos.
- Las validaciones físicas con LoRa mostraron que el modo broadcast directo presentó latencias uniformes de aproximadamente 570 ms, mientras que el modo multi-hop duplicó la latencia para nodos en rango directo (\sim 912 ms) y la triplicó para nodos que requerían un salto adicional (\sim 1.8 s), extendiendo significativamente la cobertura de la red.

A partir de los resultados obtenidos y considerando las limitaciones establecidas en el alcance, se plantean las siguientes recomendaciones para futuras investigaciones:

- Implementar la red en un entorno real, como el lago de Atitlán u otras zonas de estudio, con el fin de validar el desempeño de la topología y la tecnología de comunicación en condiciones ambientales adversas.
- Integrar sensores físicos de variables ambientales (temperatura, humedad, calidad del agua o del aire) para evaluar la red en escenarios de monitoreo remoto con datos reales.
- Incluir un análisis detallado del consumo energético en condiciones de campo, con el objetivo de determinar la autonomía de los nodos y la viabilidad de la red para aplicaciones de largo plazo.
- Ampliar el número de nodos y escenarios evaluados en simulaciones y pruebas físicas, con el fin de caracterizar mejor la tolerancia a fallos en redes más complejas y con topologías dinámicas.
- Considerar la implementación de mecanismos de seguridad en la transmisión de datos, ya que la robustez de la comunicación no solo depende de la tolerancia a fallos, sino también de la protección frente a posibles ataques o accesos no autorizados.

-
- [1] K. S. Adu-Manu et al., *WSN Architectures for Environmental Monitoring Applications*, 2022. DOI: 10.1155/2022/7823481.
 - [2] M. Ragnoli, A. Leoni, G. Barile, G. Ferri y V. Stornelli, «LoRa-Based Wireless Sensors Network for Rockfall and Landslide Monitoring: A Case Study in Pantelleria Island with Portable LoRaWAN Access,» *Journal of Low Power Electronics and Applications*, vol. 12, 3 sep. de 2022, ISSN: 20799268. DOI: 10.3390/jlpea12030047.
 - [3] M. Ragnoli, M. Scarsella, A. Leoni, G. Ferri y V. Stornelli, «Wireless Sensor Network-Based Rockfall and Landslide Monitoring Systems: A Review,» *Sensors*, vol. 23, 16 ago. de 2023, ISSN: 14248220. DOI: 10.3390/s23167278.
 - [4] S. C. Prabha et al., «Solutions of Detour Distance Graph Equations,» *Sensors*, vol. 22, 21 nov. de 2022, ISSN: 14248220. DOI: 10.3390/s22218440.
 - [5] B. Musznicki, M. Piechowiak y P. Zwierzykowski, «Modeling and Analyzing Urban Sensor Network Connectivity Based on Open Data,» *Sensors*, vol. 23, 23 dic. de 2023, ISSN: 14248220. DOI: 10.3390/s23239559.
 - [6] OpenAI, *ChatGPT (versión GPT-5)*, Modelo de lenguaje de gran tamaño. Disponible en: <https://chat.openai.com/>, 2025.
 - [7] GitHub, *GitHub Copilot*, Asistente de programación basado en IA. Disponible en: <https://github.com/features/copilot>, 2025.
 - [8] A. Abdulwahid y M. Salih, «Wireless Sensor Networks Applications, Challenges, and Security Requirements,» European Alliance for Innovation n.o., mar. de 2022. DOI: 10.4108/eai.7-9-2021.2314823.
 - [9] J. A. Stankovic, *Wireless Sensor Networks*.
 - [10] M. R. Sakarvadia y M. Mangal, *NETWORK TOPOLOGIES IN WIRELESS SENSOR NETWORK*, 2022. dirección: www.ijcrt.org.
 - [11] M. I. Labib, M. ElGazzar, A. Ghalwash y S. N. AbdulKader, «An efficient networking solution for extending and controlling wireless sensor networks using low-energy technologies,» *PeerJ Computer Science*, vol. 7, 2021, ISSN: 23765992. DOI: 10.7717/peerj-cs.780.

- [12] L. M. Chatue, R. C. G. N. Ewo, L. N. Nneme, N. Dada y V. W. F. Tekam, «Cryptographic Algorithm Grafted onto the ESP-NOW Protocol for Security Optimization in Wireless Sensor Network Chains,» *Journal of Information Security*, vol. 16, págs. 313-329, 02 2025, ISSN: 2153-1234. DOI: 10.4236/jis.2025.162016.
- [13] Z. Abidin, R. Falah, R. A. Setyawan y F. C. Wardana, «Wireless sensor network using nRF24L01+ for precision agriculture,» *Bulletin of Electrical Engineering and Informatics*, vol. 14, págs. 1003-1013, 2 abr. de 2025, ISSN: 23029285. DOI: 10.11591/eei.v14i2.8481.
- [14] Nordic Semiconductor, *nRF24L01 Product Specification v2.0*, Product datasheet, Nordic Semiconductor ASA, jul. de 2007. visitado 28 de sep. de 2025. dirección: https://infocenter.nordicsemi.com/pdf/nRF24L01_PS_v2.0.pdf.
- [15] A. O. Abdalla, A. Z. Abdalla, M. N. Muftah y M. S. Alsahulli, *Monitoring and Controlling Temperature Sensors by Wireless Network Using NRF24L01*, 2020. dirección: www.ijiset.com.
- [16] A. Augustin, J. Yi, T. Clausen y W. M. Townsley, «A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,» *Sensors*, vol. 16, n.º 9, pág. 1466, 2016, ISSN: 1424-8220. DOI: 10.3390/s16091466.
- [17] L. García, C. Cancimance, R. Asorey-Cacheda, C.-L. Zúñiga-Cañón, A.-J. Garcia-Sanchez y J. Garcia-Haro, «Compliant and Seamless Hybrid (Star and Mesh) Network Topology Coexistence for LoRaWAN: A Proof of Concept,» *Applied Sciences*, vol. 15, n.º 7, pág. 3487, mar. de 2025. DOI: 10.3390/app15073487.
- [18] *19th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2018): June 12–15, 2018, Chania, Greece*, IEEE Catalog Number: CFP18WOW-USB; CFP18WOW-ART, IEEE, 2018, ISBN: 978-1-5386-4725-7.
- [19] N. Jovalekic, V. Drndarevic, E. Pietrosevoli, I. Darby y M. Zennaro, «Experimental Study of LoRa Transmission over Seawater,» *Sensors*, vol. 18, n.º 9, pág. 2853, sep. de 2018, ISSN: 1424-8220. DOI: 10.3390/s18092853.
- [20] D. Sarioz, *Geometric Graph Theory and Wireless Sensor Networks*, 2012. dirección: <https://academicworks.cuny.edu>.
- [21] N. Y. B. H. B. H. K. L. M. P. S. Tokyo y E. S. B. A. F. G. K. Ribet, *Springer Graduate Texts in Mathematics 207*.
- [22] M. E. Gritslab, *Networked Control Modeling, Design, and Applications Outline: 1. Graph-Based Control 2. Multi-Agent Networks 3. Control of Robot Teams 4. Sensor Networks*. dirección: www.ece.gatech.edu/~magnus.
- [23] GeeksforGeeks. «Strongly Connected Components.» Accedido: 30 de septiembre de 2025. dirección: <https://www.geeksforgeeks.org/dsa/strongly-connected-components/>.
- [24] GeeksforGeeks. «Kosaraju's Algorithm in C++.» Accedido: 30-sep-2025. dirección: <https://www.geeksforgeeks.org/cpp/kosarajus-algorithm-in-cpp/>.
- [25] D. Ağaoğlu y O. Çağrıncı, «Unit Disk Visibility Graphs,» *arXiv preprint arXiv:2103.14092*, 2021. dirección: <https://arxiv.org/abs/2103.14092>.
- [26] FreeMapTools. «Measure Distance.» Accedido: 30-sep-2025. dirección: <https://www.freemaptools.com/measure-distance.htm>.

- [27] Saleae Inc., *Saleae Logic Analyzers*, <https://saleae.com/>, Accedido: 14 de noviembre de 2025.
- [28] Heltec Automation, *WiFi LoRa 32 V3 (HTIT-WB32LA_V3.2) Hardware Specification*, Accedido: 15-nov-2025, 2023. dirección: https://resource.heltec.cn/download/WiFi_LoRa_32_V3/HTIT-WB32LA_V3.2.pdf.
- [29] J. Gromes, *RadioLib: Universal Wireless Communication Library*, <https://github.com/jgromes/RadioLib>, Accedido: 15-nov-2025, 2025.
- [30] Superintendencia de Telecomunicaciones de Guatemala. «Tabla Nacional de Atribución de Frecuencias (TNAF).» Accedido: 15-nov-2025. dirección: <https://sit.gob.gt/gerencia-de-frecuencias/frecuencias/tabla-nacional-de-atribucion-de-frecuencias/>.
- [31] LoRa Alliance, «LoRaWAN Regional Parameters Specification, Version RP002-1.0.2,» LoRa Alliance Technical Committee, inf. téc., 2020, Accedido: 15-nov-2025. dirección: https://lora-alliance.org/wp-content/uploads/2020/11/RP_2-1.0.2.pdf.

Los anexos presentados en esta sección reúnen el material complementario que respalda el desarrollo teórico y experimental del trabajo. Se incluyen matrices, cálculos, *scripts* utilizados para el análisis de grafos no dirigidos y dirigidos, así como los resultados completos de las pruebas de latencia en ESP-NOW. Asimismo, se adjuntan los códigos implementados para cada una de las tecnologías evaluadas y los datos derivados de las simulaciones y validaciones físicas.

12.1. Laplacianas por caso de eliminación (grafos no dirigidos)

12.1.1. Caso 1: eliminación del nodo 1

Nodos restantes: $\{2, 3, 4, 5, 6, 7, 8\}$. La matriz laplaciana L_{-v_1} queda:

$$L_{-v_1} = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 3 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 2 \end{bmatrix} \quad (6)$$

Valor de Fiedler: $\lambda_2 = 0.75302$.

12.1.2. Caso 2: eliminación del nodo 2

Nodos restantes: $\{1, 3, 4, 5, 6, 7, 8\}$. La matriz laplaciana L_{-v_2} queda:

$$L_{-v_2} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 & 3 \end{bmatrix} \quad (7)$$

Valor de Fiedler: $\lambda_2 = 0.62280$.

Caso 4: eliminación del nodo 4

Nodos restantes: $\{1, 2, 3, 5, 6, 7, 8\}$. La matriz laplaciana L_{-v_4} queda:

$$L_{-v_4} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \quad (8)$$

Valor de Fiedler: $\lambda_2 = 0.38889$.

Caso 5: eliminación del nodo 5

Nodos restantes: $\{1, 2, 3, 4, 6, 7, 8\}$. La matriz laplaciana L_{-v_5} queda:

$$L_{-v_5} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \quad (9)$$

Valor de Fiedler: $\lambda_2 = 0.63383$.

Caso 6: eliminación del nodo 6

Nodos restantes: $\{1, 2, 3, 4, 5, 7, 8\}$. La matriz laplaciana L_{-v_6} queda:

$$L_{-v_6} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \quad (10)$$

Valor de Fiedler: $\lambda_2 = 0.60862$.

Caso 7: eliminación del nodo 7

Nodos restantes: $\{1, 2, 3, 4, 5, 6, 8\}$. La matriz laplaciana L_{-v_7} queda:

$$L_{-v_7} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Valor de Fiedler: $\lambda_2 = 0.26795$.

Caso 8: eliminación del nodo 8

Nodos restantes: $\{1, 2, 3, 4, 5, 6, 7\}$. La matriz laplaciana L_{-v_8} queda:

$$L_{-v_8} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 \\ -1 & -1 & 4 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 2 \end{bmatrix} \quad (12)$$

Valor de Fiedler: $\lambda_2 = 0.43741$.

12.2. Scripts de grafos no dirigidos en MATLAB

Con el fin de facilitar la consulta, los códigos utilizados para el análisis de los grafos no dirigidos se encuentran disponibles en un repositorio público de GitHub.

En este repositorio se incluyen las distintas versiones del código desarrolladas (V1, V2 y V3), así como el archivo de excel con los resultados para todos los casos de desconexión.

Al repositorio puede accederse desde el siguiente enlace: <https://github.com/AngelaQue12/Anexos-Tesis>

12.3. Resumen de resultados para el grafo no dirigido preliminar

En el Cuadro 6 se presentan los cinco escenarios más críticos identificados en el grafo no dirigido preliminar, es decir, aquellos con menor valor de Fiedler (λ_2).

Por otro lado, el Cuadro 7 muestra los cinco escenarios más robustos, caracterizados por los mayores valores de λ_2 . Estos corresponden a configuraciones en las que quedan únicamente dos o tres nodos conectados, alcanzando valores de λ_2 iguales a 2 o 3.

Cuadro 6. Peores 5 casos entre conexos (valor mínimo de λ_2)

Removed	Remaining	λ_2	Connected
{3}	7	0.19806	true
{1,3}	6	0.26795	true
{7}	7	0.26795	true
{1,7}	6	0.26795	true
{3,4}	6	0.26795	true

Nota. El cuadro presenta los cinco escenarios más críticos del grafo no dirigido preliminar, caracterizados por los valores más bajos de la conectividad algebraica λ_2 . Estos casos corresponden a configuraciones donde la eliminación de ciertos nodos reduce significativamente la robustez estructural de la red.

Cuadro 7. Mejores 5 casos entre conexos (valor máximo de λ_2)

Removed	Remaining	λ_2	Connected
{4,5,6,7,8}	3	3	true
{1,2,3,4,5,6}	2	2	true
{1,2,3,4,5,8}	2	2	true
{1,2,3,4,6,7}	2	2	true
{1,2,3,6,7,8}	2	2	true

Nota. El cuadro presenta los cinco escenarios más robustos del grafo no dirigido preliminar, caracterizados por los valores más altos de la conectividad algebraica λ_2 . Estos casos corresponden a configuraciones con pocos nodos remanentes, en las que la red mantiene una conectividad fuerte con valores de λ_2 iguales a 2 o 3.

12.4. Resumen de resultados para el grafo no dirigido triangulado

En el análisis del grafo triangulado se identificaron los escenarios más críticos y más robustos en términos de conectividad algebraica. El Cuadro 8 presenta los cinco casos con menor valor de Fiedler (λ_2), mientras que el Cuadro 9 resume los cinco casos con mayor λ_2 .

Cuadro 8. Peores 5 casos entre conexos (valor mínimo de λ_2) para el grafo triangulado

Removed	Remaining	λ_2	Connected
{3}	7	0.22538	true
{3,4}	6	0.26795	true
{3,5}	6	0.26795	true
{1,3}	6	0.32487	true
{1,3,4}	5	0.38197	true

Nota. El cuadro presenta los cinco escenarios más críticos del grafo no dirigido triangulado, caracterizados por los valores más bajos de la conectividad algebraica λ_2 . Estos casos corresponden a configuraciones donde la eliminación de ciertos nodos reduce significativamente la robustez estructural de la red.

Cuadro 9. Mejores 5 casos entre conexos (valor máximo de λ_2) para el grafo triangulado

Removed	Remaining	λ_2	Connected
{1,2,3,6,7}	3	3	true
{1,2,4,5,6}	3	3	true
{1,2,4,5,8}	3	3	true
{1,2,5,6,7}	3	3	true
{1,4,5,7,8}	3	3	true

Nota. El cuadro presenta los cinco escenarios más robustos del grafo no dirigido triangulado, caracterizados por los valores más altos de la conectividad algebraica λ_2 . Estos casos corresponden a configuraciones con pocos nodos remanentes, en las que la red mantiene una conectividad fuerte con valores de λ_2 iguales a 3.

12.5. Scripts de grafos dirigidos MATLAB

Con el fin de facilitar la consulta, los códigos utilizados para el análisis de los grafos dirigidos se encuentran disponibles en un repositorio público de GitHub.

En este repositorio se incluyen las distintas versiones del código desarrolladas (V1, V2 y V3), así como el archivo de excel con los resultados para todos los casos de desconexión.

Al repositorio puede accederse desde el siguiente enlace: <https://github.com/AngelaQue12/Anexos-Tesis>

12.6. Resumen de resultados para el grafo dirigido

El Cuadro 10 presenta el resumen de resultados del análisis de fuerte conectividad agrupados según el número de nodos remanentes en cada subgrafo. Para cada tamaño se indica el número total de configuraciones evaluadas, cuántas de ellas se mantuvieron fuertemente conexas (SC) y cuántas no lo fueron (No-SC).

Cuadro 10. Resumen por tamaño de subgrafo

Remaining	Total de casos	SC	No-SC
2	28	0	28
3	56	9	47
4	70	16	54
5	56	22	34
6	28	20	8
7	8	8	0

Nota. El cuadro muestra el resumen de resultados del análisis de fuerte conectividad agrupados según el número de nodos remanentes en cada subgrafo.

12.7. Scripts pruebas de latencia

Con el fin de facilitar la consulta, los códigos utilizados para las pruebas de latencia se encuentran disponibles en un repositorio público de GitHub en la carpeta PruebasLatencia.

Al repositorio puede accederse desde el siguiente enlace: <https://github.com/AngelaQue12/Anexos-Tesis>

12.8. Resultados de pruebas de latencia

Además del análisis de la red completa y del caso crítico sin el nodo 3, se evaluaron diversos escenarios adicionales que se incluyen de el Cuadro 11 a la Tabla 21. Estos casos corresponden a la desconexión individual de nodos 2, 4, 5, 6, 7 y 8, así como a combinaciones críticas y de interés 3-4, 3-5, 5-6, 4-5-7-8 y 4-5-6-7-8.

Cuadro 11. Prueba de latencia en red sin nodo 2

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	-	-	-
3	3.81296	36.13216	39.94512
4	165.35792	231.4184	396.77632
5	326.21472	382.40864	708.62336
6	231.17904	242.7896	473.96864
7	157.8816	244.77536	402.65696
8	156.72448	262.7432	419.46768

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin el nodo 2. Se observa que la eliminación del nodo 2 no afecta de manera crítica la conectividad de la red.

Cuadro 12. Prueba de latencia en red sin nodo 4

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.69248	6.20416	9.89664
3	3.58032	2.61984	6.20016
4	-	-	-
5	856.99392	916.55824	1773.55216
6	146.80608	198.70384	345.50992
7	379.9856	403.65232	783.63792
8	666.64592	626.07488	1292.72080

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin el nodo 4. Se observa que la eliminación de este nodo afecta significativamente a los nodos más alejados, incrementando la latencia, aunque la red mantiene conectividad.

Cuadro 13. Prueba de latencia en red sin nodo 5

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.7936	1.16864	4.96224
3	3.78976	2.58704	6.37680
4	170.8448	163.29008	334.13488
5	-	-	-
6	156.09616	154.41248	310.50864
7	161.38656	161.65904	323.04560
8	158.56176	172.58368	331.14544

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin el nodo 5. Se observa que la eliminación del nodo 5 no afecta de manera crítica la conectividad de la red.

Cuadro 14. Prueba de latencia en red sin nodo 6

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	4.008	17.16544	21.17344
3	4.27072	2.42032	6.69104
4	161.62848	174.30912	335.93760
5	325.93952	345.49152	671.43104
6	-	-	-
7	162.49248	218.07856	380.57104
8	167.36112	212.99920	380.36032

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin el nodo 6. Se observa que la eliminación del nodo 6 no afecta de manera crítica la conectividad de la red.

Cuadro 15. Prueba de latencia en red sin nodo 7

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.76736	2.35952	6.12688
3	3.41856	2.38320	5.80176
4	155.61056	159.75680	315.36736
5	322.97269	333.08336	656.05605
6	147.87232	150.79808	298.67040
7	-	-	-
8	162.68784	202.90464	365.59248

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin el nodo 7. Se observa que la eliminación del nodo 7 no afecta de manera crítica la conectividad de la red.

Cuadro 16. Prueba de latencia en red sin nodo 8

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.87888	1.19968	5.07856
3	5.90720	6.07008	11.97728
4	162.04704	164.99328	327.04032
5	330.77968	333.78976	664.56944
6	151.04608	149.80208	300.84816
7	168.58096	159.74800	328.32896
8	-	-	-

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin el nodo 8. Se observa que la eliminación del nodo 8 no afecta de manera crítica la conectividad de la red.

Cuadro 17. Prueba de latencia en red sin nodos 3 y 4

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	8.17696	1.20688	9.38384
3	-	-	-
4	-	-	-
5	895.37408	937.08752	1832.46160
6	151.56848	191.14448	342.71296
7	365.04672	403.19552	768.24224
8	645.15984	646.46864	1291.62848

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin los nodos 3 y 4. Se observa un incremento notable en la latencia para varios nodos, indicando que la eliminación simultánea de ambos afecta de manera significativa la conectividad de la red.

Cuadro 18. Prueba de latencia en red sin nodos 3 y 5

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	4.3016	2.00224	6.30384
3	-	-	-
4	897.60544	863.13856	1760.74400
5	-	-	-
6	151.51696	194.6704	346.18736
7	364.54608	401.23808	765.78416
8	640.06864	645.56576	1285.63440

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin los nodos 3 y 5. Se observa un incremento notable en la latencia para varios nodos, indicando que la eliminación conjunta de ambos afecta de manera significativa la conectividad de la red.

Cuadro 19. Prueba de latencia en red sin nodos 5 y 6

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.63248	3.48416	7.11664
3	3.92272	3.04032	6.96304
4	158.42624	175.22672	333.65296
5	-	-	-
6	-	-	-
7	160.80176	211.97168	372.77344
8	169.81632	229.43536	399.25168

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin los nodos 5 y 6. Se observa que la eliminación conjunta de estos nodos no afecta de manera crítica la conectividad de la red.

Cuadro 20. Prueba de latencia en red sin nodos 4, 5, 7 y 8

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Latencia total (ms)
2	3.7616	1.20464	4.96624
3	4.72864	5.99792	10.72656
4	-	-	-
5	-	-	-
6	150.81376	146.7864	297.60016
7	-	-	-
8	-	-	-

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin los nodos 4, 5, 7 y 8. Se observa que la red mantiene conectividad mínima, aunque con rutas limitadas entre los nodos restantes.

Cuadro 21. Resultados de latencia para la red sin nodos 4, 5, 6, 7 y 8

No. Nodo Emisor	ΔT NE→N1 (ms)	ΔT N1→NE (ms)	Suma (ms)
2	5.40368	1.85216	7.25584
3	5.54096	4.6496	10.19056
4	-	-	-
5	-	-	-
6	-	-	-
7	-	-	-
8	-	-	-

Nota. El cuadro muestra los resultados de latencia obtenidos para la red sin los nodos 4, 5, 6, 7 y 8. Se observa que únicamente los nodos más cercanos entre sí mantienen latencia baja.

De manera general, los resultados permiten observar que:

- Las desconexiones individuales no críticas muestran latencias totales moderadas, lo cual indica que la red mantiene rutas alternativas relativamente eficientes.
- En las desconexiones críticas se genera un aumento abrupto de la latencia, alcanzando valores superiores a 1700 ms en nodos alejados. Si bien su desconexión impacta incrementando las latencias y reduciendo las trayectorias disponibles, la red no colapsa por completo. Por el contrario, conserva rutas alternativas que, aunque menos eficientes, garantizan que el sistema continúe operativo.
- Finalmente en los escenarios con múltiples desconexiones reducen considerablemente la cantidad de caminos disponibles, lo que se traduce en que solo algunos emisores mantienen conectividad hacia el nodo 1, con latencias bajas únicamente en los nodos más cercanos al ser los únicos nodos remanentes.

En el repositorio público de GitHub en la carpeta PruebasLatencia se adjuntan los resultados obtenidos por medio de la aplicación de Logic Analyzer Logic 2.4.29, estos resultados incluyen las capturas de señales como el archivo de excel que recopila los tiempos de

cada prueba y un archivo de word con las imagenes que muestran el comportamiento de los mensajes para cada prueba. <https://github.com/AngelaQue12/Anexos-Tesis>

12.9. Script implementación final ESP-NOW

Con el fin de facilitar la consulta, los códigos utilizados en la implementación final de ESP-NOW se encuentran en un repositorio público de GitHub.

Al repositorio puede accederse desde el siguiente enlace: <https://github.com/AngelaQue12/Anexos-Tesis>

12.10. Resultados implementación final ESP-NOW

Figura 16. Resultados para red completa

```
I (6982) BROADCAST: NODO 3: Recibido ACK de 8 para 3 (ID:1293481925, TTL:8)
I (6982) BROADCAST: NODO 3: ACK recibido de nodo 8 (7/7)
I (7012) BROADCAST: NODO 3: ¡Todos los ACKs recibidos!
I (7012) BROADCAST: Nodo 1: CONECTADO
I (7012) BROADCAST: Nodo 2: CONECTADO
I (7012) BROADCAST: Nodo 4: CONECTADO
I (7012) BROADCAST: Nodo 5: CONECTADO
I (7012) BROADCAST: Nodo 6: CONECTADO
I (7022) BROADCAST: Nodo 7: CONECTADO
I (7022) BROADCAST: Nodo 8: CONECTADO
```

Nota. Terminal serial, prueba de funcionamiento para implementación final de ESP-NOW en la red completa. Elaboración propia.

Figura 17. Resultados sin nodo 6 y 8

```
W (33412) BROADCAST: NODO 3: Timeout esperando ACKs (5/7 recibidos)
I (33412) BROADCAST: Nodo 1: CONECTADO
I (33412) BROADCAST: Nodo 2: CONECTADO
I (33412) BROADCAST: Nodo 4: CONECTADO
I (33412) BROADCAST: Nodo 5: CONECTADO
W (33412) BROADCAST: Nodo 6: DESCONECTADO
I (33422) BROADCAST: Nodo 7: CONECTADO
W (33422) BROADCAST: Nodo 8: DESCONECTADO
```

Nota. Terminal serial, prueba de funcionamiento para implementación final de ESP-NOW sin los nodos 6 y 8 en la red. Elaboración propia.

12.11. Scripts implementación LoRa

Con el fin de facilitar la consulta, los códigos utilizados en la implementación de LoRa tanto en modo *single-hop* como *multi-hop* se encuentran en un repositorio público de GitHub.

Al repositorio puede accederse desde el siguiente enlace: <https://github.com/AngelaQue12/Anexos-Tesis>