



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

**Diseño de una red inalámbrica para un sistema de gestión de energía
doméstico.**

TRABAJO DE TITULACIÓN.

AUTOR: Poma Sarango, Cristina Isabel

DIRECTOR: Castillo Calvas, Tuesman Daniel, Ing., M. Sc, A

LOJA-ECUADOR

2019



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

2019

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Magister.

Tuesman Daniel Castillo Calvas.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: **Diseño de una red inalámbrica para un sistema de gestión de energía doméstico**, realizado por **Poma Sarango Cristina Isabel**, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, marzo de 2019

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo **Cristina Isabel Poma Sarango** declaro ser autora del presente trabajo de fin de titulación: “Diseño de una red inalámbrica para un sistema de gestión de energía doméstico”, de la Titulación de Electrónica y Telecomunicaciones, siendo **Tuesman Daniel Castillo Calvas** director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f.....

Autor: Poma Sarango Cristina Isabel

Cédula: 1105904070

DEDICATORIA

Este trabajo es dedicado a Dios por permitirme culminar esta meta, a mis padres Nelson y Carmen por siempre apoyarme y guiarme en cada momento de mi vida tanto en mi formación personal como académica.

A mis hermanos Diego, Darío, Elías, María, Marco, Salomé y Paola, mi hija Samanta y amigos por todo el apoyo brindado y por el inmenso cariño y aprecio que les tengo.

AGRADECIMIENTO

A Dios principalmente por guiarme y ser el sustento de mi vida.

A mis padres, hermanos y mi hija por el apoyo incondicional, paciencia y confianza brindada durante todo este proceso de formación académica.

A los docentes de la Titulación de Electrónica y Telecomunicaciones por los conocimientos impartidos dentro y fuera del aula, de manera especial a mi director Ing. Tuesman Castillo quien, con su paciencia, confianza y tiempo supo guiarme en el desarrollo de este proyecto.

ÍNDICE DE CONTENIDOS

CARÁTULA.....	i
APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN.....	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
LISTA DE FIGURAS	viii
LISTA DE TABLAS	ix
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN	3
OBJETIVOS.....	5
CAPÍTULO I.....	6
REVISIÓN DE LA LITERATURA	6
1.1 HEMS	7
1.2 Tecnologías y protocolos de comunicación	8
1.3 Topologías	9
1.4 Microcontroladores.....	10
1.5 Sensor.....	11
1.6 Técnicas de modulación.....	11
CAPÍTULO II.....	13
ARQUITECTURA DE WSN	13
2.1 WSN.....	14
2.1.1 Características.....	14
2.1.2 Limitaciones.....	15
2.2 Elementos de la red.....	15
2.2.1 Nodos	15
2.2.2 Arquitectura de la red.....	16
2.3 Internet de las cosas.....	18
2.4 Consumo de energía	19
2.5 Software para WSN.....	19
2.5.1 Network Simulator 3.....	20

2.5.2 OMNET ++.....	21
2.5.3 Network Simulator 2.....	21
2.5.4 OPNET	21
2.5.5 Comparación de herramientas para la simulación.....	22
CAPITULO III.....	24
PROTOCOLOS Y TECNOLOGÍAS DE COMUNICACIÓN PARA WSN	24
3.1 Tecnologías de comunicación inalámbrica.....	25
3.1.1 Wi-Fi (IEEE 802.11)	25
3.1.2 Bluetooth (IEEE 802.15.1)	26
3.1.3 ZigBee (IEEE 802.15.4)	26
3.2 Protocolos para WSN	27
3.2.1 OLSR.....	27
3.2.2 AODV	28
3.2.3 DSR.....	28
3.2.3 DSDV.....	28
CAPITULO IV	29
DESCRIPCIÓN DE LA SIMULACIÓN DE REDES DE SENSORES	29
4.1 Simulador de red ns-3	30
4.2 Diseño de la red	30
4.3 Simulación.....	31
4.3.1 Descripción de los parámetros de simulación	32
4.3.2 Especificaciones del software	35
4.3.3 Funcionamiento y estructura de ns-3	36
4.3.4 Módulo de energía en ns-3.26	39
4.3.5 Resultados de la simulación.....	41
4.3.6 PROPUESTA.....	47
CONCLUSIONES.....	50
RECOMENDACIONES	51
REFERENCIAS.....	52
ANEXOS.....	59
ANEXO A	60
ANEXO B	66
ANEXO C.....	68
ANEXO D.....	69
ANEXO E	71

LISTA DE FIGURAS

Figura 1. Arquitectura de HEMS	7
Figura 2. Arquitectura de el enchufe inteligente Edimax SP-2101W	11
Figura 3. Representación de una WSN.....	14
Figura 4. Estructura básica de un nodo sensor	16
Figura 5. Topología punto a punto	16
Figura 6. Topología de red en estrella.....	17
Figura 7. Topología malla	18
Figura 8. Consumo de energía de un nodo sensor.	19
Figura 9. Placa del circuito del medidor Inteligente	31
Figura 10. Escenario A, casa de dos pisos	32
Figura 11. Escenario B, edificio de 3 pisos.	34
Figura 12. Funcionamiento de ns-3.....	36
Figura 13. Módulos existentes en ns-3.....	36
Figura 14. Framework de Energía.....	39
Figura 15. Despliegue del escenario A.....	41
Figura 16. Visualización durante la transmisión y recepción de datos por medio de PyViz. .	42
Figura 17. Consumo de energía del escenario A con Wi-Fi durante 1 hora.....	42
Figura 18. Consumo de energía de Wi-Fi con algunos protocolos, en un intervalo de 1h. ...	43
Figura 19. Simulación del escenario B.....	44
Figura 20. Visualización de la transmisión y recepción de datos en un escenario B	44
Figura 21. Consumo de energía del escenario B en un período de 1 hora.....	45
Figura 22. Consumo de energía durante la transmisión y recepción por un tiempo de 60s. .	45
Figura 23. Resultados de consumo energético del escenario A y B con ZigBee	46
Figura 24. Consumo energético de las tecnologías Wi-Fi y Zigbee en cada escenario.	47

LISTA DE TABLAS

Tabla 1. Comparación de algunas plataformas de red.	22
Tabla 2. Comparativa entre tecnologías de comunicación Inalámbrica.	26
Tabla 3. Descripción de los parámetros de simulación del escenario A	32
Tabla 4. Descripción de los parámetros de simulación del escenario B	34
Tabla 5. Especificación del software, dependencias y hardware.	35
Tabla 6. Consumo de energía en los estándares Wifi y ZigBee	48
Tabla 7. Plataformas hardware IoT	49

RESUMEN

Este trabajo presenta el diseño y simulación de una red inalámbrica para monitoreo y control de energía en el hogar. El rendimiento y eficiencia energética en hogares inteligentes reducen la demanda de energía a la red eléctrica y minimizan costos. El consumo de energía en una red inalámbrica de sensores (WSN) es el principal limitante en el rendimiento de la red. Se simula en ns-3 una red WSN con tecnología Wi-Fi, algunos protocolos diseñados para redes ad-hoc como OLSR, DSDV, DSR y AODV, para analizar el consumo de energía con el fin de recomendar la red inalámbrica con el mejor rendimiento energético.

PALABRAS CLAVE: HEMS, WSN, consumo de energía, ns-3, framework de energía.

ABSTRACT

This work presents the design and simulation of a wireless network for monitoring and control of energy in the home. Energy efficiency and performance in smart households reduce energy demand to the power grid and minimize costs. The power consumption in a wireless sensor network (WSN) is the main limiting in the performance of the network. This work simulates in ns-3 a WSN network with Wi-Fi Technology, some protocols designed for ad-hoc networks such as OLSR, DSDV, DSR and AODV, to analyze energy consumption in order to recommend the wireless network with the best energy efficiency.

KEYWORDS: HEMS, WSN, power consumption, ns-3, energy framework.

INTRODUCCIÓN

Las redes de sensores inalámbricas (WSN, Wireless Sensor Networks) desde años atrás hasta la actualidad han sido el centro de atención de investigadores y usuarios en general, debido a la diversidad de aplicaciones de diferentes campos de la industria tales como el campo militar, ambiental, medicina, de energía, domótica entre otros. La investigación sobre las redes de sensores inició en la década de 1980, cuando la Agencia de Proyectos de Investigación Avanzada de Defensa de los Estados Unidos (DARPA) llevó a cabo el programa de redes de sensores distribuidos (DSN) para el ejército de los EE. UU (Chee-Yee Chong & Kumar, 2003) .

El Internet de las Cosas (IoT), Big Data y tecnologías inteligentes son estrategias tecnológicas de mayor influencia en investigaciones e incluso en la interconexión de redes, entre ellas la WSN, requiere alta eficiencia energética y un amplio rango de cobertura. Este sistema tiene la capacidad de conectar dispositivos inteligentes, pequeños actuadores y personas, brindando la capacidad de monitorear y controlar el estado en el que se encuentren ciertos dispositivos sin importar su ubicación geográfica (Ling et al., 2017). Tecnologías que permiten el uso de diferentes escenarios como Smart cities o ciudades Inteligentes, se encuentran en constante desarrollo y crecimiento. Se derivan diferentes escenarios como casas, transporte, tráfico, salud inteligente entre otros, controlados y monitoreados por dispositivos electrónicos inteligentes que generan datos, permitiendo al usuario conocer en tiempo real diferentes características como la temperatura, el control de luz, entre otros (Hammi, Khatoun, Zeadally, Fayad, & Khoukhi, 2018).

La WSN tiene algunas restricciones para adaptarse en aplicaciones de tiempo real, como energía limitada, capacidad de computación y recursos de memoria limitados que van asociados con otros problemas como baja velocidad de reloj y la velocidad del procesador, además aumentan el consumo de energía y su costo en cuanto a implementación. De esta manera las diferentes aplicaciones de redes de sensores tienen requisitos diferentes, que se los debe cumplir para lograr un buen funcionamiento (Kodali & Aravapalli, 2014).

Los nodos sensores en WSN disipan energía al detectar, procesar, transmitir o recibir datos, la alimentación de energía de la red es generalmente a base de baterías, es la característica que convierte la energía limitada en el principal inconveniente de la red. Si la ubicación de los nodos lo permite, estos pueden alimentarse también de la red eléctrica (Nayak & Devulapalli, 2016; Souza & Minet, 2011). Todo lo que compone un nodo sensor consume energía, por ende, es importante el ahorro de energía, un aspecto que se debe tomar en cuenta en el diseño e implementación de este tipo de red.

El diseño de una red inalámbrica para un sistema de gestión de energía doméstico es crucial ya que permitiría tener el control del consumo de energía de cada dispositivo existente, además es importante analizar el consumo dentro de la red, es decir se deben seleccionar los componentes correctos y empleando técnicas de diseño eficientes. El sistema de gestión de energía doméstico (HEMS, llamado así por sus siglas en inglés Home Energy Management System) es una solución viable para conocer y gestionar su consumo, por esta razón es importante analizarlo. Para diseñar la arquitectura de la red se debe analizar que sea eficiente y funcional, es importante la búsqueda, análisis y comparación de información sobre tecnologías de comunicación inalámbrica como Zigbee, Wifi, Bluetooth, incluso el microcontrolador como Atmega328 u otros modelos, además de los protocolos y técnicas de modulación que sean apropiados para esta red.

En (Pérez, 2019) se diseñó un sensor para medir el consumo energético de cada dispositivo existente en el hogar y se plantea integrarlo a una red de sensores con el fin de gestionar la energía permitiéndole al usuario en tiempo real conocer que dispositivo es de mayor o menor consumo, que genere un historial energético de todo un día, mes o incluso año. Esto le permitirá al usuario tomar medidas de ahorro, es decir, pueda autorregularse teniendo en cuenta que si consume más tiene que pagar más, y viceversa, el objetivo es bajar el precio regulando el consumo. En el presente proyecto se plantea realizar un análisis, comparación, de arquitectura de la red, protocolos, tecnologías que adopte un sistema de gestión de energía basado en una red inalámbrica de sensores y actuadores empleados para optimizar su consumo de energía. El diseño de la red debe cumplir con los objetivos planteados, para ello se ha decidido emplear el simulador de red Network Simulator (ns-3) para la simulación de diferentes arquitecturas de red para WSN, analizando y comparando con el fin de escoger el diseño que se acerque o cumpla con lo requerido. Se espera obtener datos, información que sirva como herramienta para el diseño de futuras redes, en especial para redes de sensores para HEMS.

OBJETIVOS

Objetivo General

- Diseñar una red inalámbrica para un sistema de gestión de energía doméstico.

Objetivos Específicos

- Analizar el estado del arte de sistemas de gestión de energía (Home Energy Management Systems).
- Diseño de la arquitectura de la red y formato de datos.
- Comparación de varias tecnologías de comunicación inalámbrica, protocolos y técnicas de modulación en cuanto a consumo de energía.

CAPÍTULO I

REVISIÓN DE LA LITERATURA

Los sistemas de gestión de energía en el hogar actualmente se están vinculando con el concepto de red inteligente e Internet de las Cosas. Este sistema por lo general busca optimizar el consumo de energía en el hogar, a través de una unidad de gestión de energía y fuentes de energía renovables como la solar, eólica, entre otras, es importante recalcar que al minimizar el consumo de energía también mejora la economía. Estas nuevas tecnologías junto con HEMS permiten tener el control y administrar de manera efectiva la energía en el hogar.

1.1 HEMS

La demanda de energía en el hogar ha abierto diversas líneas de investigación con el fin de mejorar el rendimiento energético en edificios por medio de una red inteligente. En (Zhou et al., 2016) HEMS es un “sistema óptimo que proporciona servicios de gestión de energía con el fin de controlar y gestionar de forma eficiente la generación, el almacenamiento y el consumo de electricidad en las casas inteligentes”, este sistema es muy utilizado en diferentes aplicaciones, investigaciones centradas en la gestión de redes inteligentes en el hogar. En este documento mencionan algunas de las funciones del sistema que permiten controlar, administrar, monitorear, recopilar y guardar información relacionada con el consumo de energía. Además, proporciona una arquitectura de HEMS inteligente general tal como se observa en la Figura 1, conformada por un controlador inteligente centralizado que está relacionado con los dispositivos de almacenamiento de energía, sensores como el medidor inteligente, fuentes de energía renovables y los electrodomésticos.

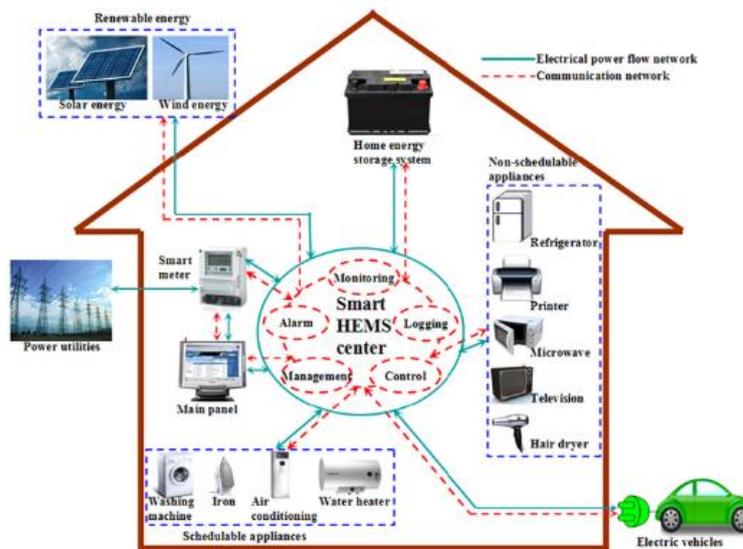


Figura 1. Arquitectura de HEMS
 Fuente: (Zhou et al., 2016)
 Elaboración: (Zhou et al., 2016)

En (Inoue et al., 2003) se propone una arquitectura de red para HEMS que resuelve problemas convencionales como la necesidad de un nuevo cableado para los dispositivos de la red y demás factores que han impedido la construcción de HEMS avanzados. Las funciones que cumple este sistema son controlar los dispositivos de forma automática y proporcionar información para el ahorro de energía.

En (Ahmed, Al Hassan, Mohamed, & Shareef, 2018) se considera una herramienta de respuesta a la demanda (DR) que cambia y reduce la demanda para mejorar el consumo de energía en el hogar. También describe el diseño de programas de consumo óptimos al considerar varios factores, como los costos de energía, las preocupaciones ambientales, los perfiles de carga y la comodidad del usuario. El sistema HEMS está logrando bastante aceptación debido a las preocupaciones relacionadas con el calentamiento global y la escasez de energía, ya que disminuye su consumo en especial en las horas pico.

El objetivo de las investigaciones relacionadas con este sistema es que los hogares y edificios inteligentes mejoren su rendimiento energético. Pretendiendo que el uso de fuentes de energía renovable sean prioridad y no opción en cada hogar o edificio, reduciendo así la demanda energética.

1.2 Tecnologías y protocolos de comunicación

La WSN se compone de varios dispositivos electrónicos que cumplen diversas funciones, permitiéndole al usuario o administrador observar y reaccionar ante cualquier evento o fenómeno que se presenta en el entorno en el que esté. Para que esta red cumpla sus funciones con normalidad emplean protocolos o estándares, tecnologías, topologías, entre otros componentes. Por ende, es importante analizar características de cada componente de red que permitan minimizar el consumo de energía de la red y que estén orientado a WSN.

En cuanto a los protocolos de control de acceso en (Yang et al., 2018) se presenta un nuevo protocolo de esquema híbrido CSMA / CA y TDMA (CTh-MAC) en la capa MAC con la finalidad de mejorar el rendimiento y minimizar el consumo de energía de una red inalámbrica WSN móvil utilizando un algoritmo de predicción de posición. Al compararlo con los protocolos existentes, el protocolo propuesto puede reducir de manera eficiente y confiable el consumo de energía, mejorando el rendimiento de la red especialmente para WSN móvil de alta velocidad en el paradigma de comunicación de muchos a uno. Al evaluar el protocolo mediante los resultados de simulación se demuestra su rendimiento.

En (Wei Ye, Heidemann, & Estrin, 2002) se propone S-MAC, un protocolo de control de acceso al medio (MAC) diseñado para redes inalámbricas de sensores. Se espera que este tipo de red se implementen de forma Ad-hoc, con nodos individuales que por largos periodos de tiempo permanezcan inactivos, pero luego repentinamente cuando detecten algo se activen. Por estas características de las redes y aplicaciones se desarrolla una MAC diferente de los tradicionales, como IEEE 802.11, la conservación de la energía y la autoconfiguración son los principales objetivos de este protocolo. SMAC emplea técnicas novedosas para reducir el consumo de energía y admitir la autoconfiguración en los nodos.

En (Callaway et al., 2002) se presenta el estándar IEEE 802.15.4 y sus aplicaciones de red doméstica. Las características más relevantes del estándar son flexibilidad de red, bajo costo y bajo consumo de energía. Otra forma muy particular para implementar en este tipo de red es emplear protocolos de transporte confiable, almacenar en caché es un método muy bueno para garantizar la confiabilidad basada en paquetes, además de disminuir el consumo de energía en los nodos intermedios debido a las retransmisiones de extremo a extremo (Alipio et al., 2017) del sensor.

Existe una gran variedad de protocolos de comunicación con esquemas orientados a la comunicación entre máquinas (M2M), uno de los más destacados es MQTT, Message Queue Telemetry transport es un protocolo de transporte de mensajes de publicación/suscripción de Client Server. Es liviano, abierto, simple y su diseño permite que sea fácil de implementar. Esto permite que la comunicación en contextos de M2M e Internet de las cosas (Akbar, Amron, Mulya, & Hanifah, 2017). Para WSN ha sido empleado para solucionar la eficiencia energética superando la pérdida de paquetes.

Existen varios tipos de tecnologías de comunicación como Wifi, Bluetooth, 3G, 4G, ZigBee, entre otras que permiten la transferencia remota de datos, el procesamiento y el control de la información. En (Batista, Melício, Matias, & Catalão, 2013) propone la tecnología ZigBee para monitorear y controlar sistemas fotovoltaicos y eólicos, pero también para la gestión de energía en el hogar. En (Collotta & Pau, 2015) se propone Bluetooth para el control de energía de cada dispositivo existente en el hogar, incluido el control de luminarias, todo con el fin de ahorrar energía.

1.3 Topologías

Las topologías de red son importantes para dirigir cada paquete de datos dentro de la red, en (Meena & Manikandan, 2017) se presenta un modelo de topología efectivo para conservar la energía de los nodos individuales en la red de sensores inalámbricos. Presenta

diferentes tipos de topologías como: Bus, anillo, estrella y su evaluación en términos de energía de recepción, energía residual, energía de estado inactivo, cantidad de paquetes enviados o recibidos en la red. El consumo de energía se analiza desde varios puntos de vista como el tamaño del paquete, el rango, el número de nodos de la red, la capacidad de la red entre otros. En este documento se utiliza el simulador de red ns-2 para simular los protocolos OLSR, AODV, DSDV, DSR en las topologías antes mencionadas.

En (Lee & Kao, 2016) se establece que el control de topología en redes inalámbricas de sensores iguala la carga de comunicación en los dispositivos de sensores y aumenta la vida útil de la red. El diseño jerárquico o basado en clúster es un enfoque para conservar la energía de las redes de sensores. Basado en una jerarquía de tres capas, este documento ha propuesto un enfoque de agrupamiento semidistribuido de los nodos. Los resultados de la simulación muestran que el enfoque propuesto es más eficiente que otros algoritmos distribuidos. Para realizar esta investigación se ha tomado en cuenta que la conservación de la energía se ha convertido en un reto para garantizar la conectividad de la red y extender la vida útil de los nodos. Además, de reconocer que generalmente la tarea de comunicación consume la mayor cantidad de energía durante la operación de la red. Se propone varios protocolos de enrutamiento para obtener una comunicación eficiente energéticamente para las WSN.

En (Moridi et al., 2018) se investiga varios arreglos de nodo de sensor en redes ZigBee para sistemas de comunicación y monitoreo en un espacio subterráneo. El rendimiento de las topologías se analiza en 12, 20 y 30 escenarios de 40 y 50 nodos con un despliegue de nodos estacionarios en entornos subterráneos. Al evaluar los resultados se comprueba que la topología malla predomina en el diseño de WSN considerando mayor rendimiento, proporción de entrega de paquetes y seguridad de red. Al analizar la topología malla se observa que crea una red de monitoreo y comunicación más confiable con una buena calidad de servicio en entornos subterráneos y túneles.

1.4 Microcontroladores

El microcontrolador es empleado “para procesar información del mundo natural, decidir un curso de acción basado en la información recopilada y luego emitir señales de control para implementar la decisión” (Steven, 2012). El microcontrolador arduino es una opción viable, barata y de fácil uso, para ser utilizado en proyectos o aplicaciones de WSN.

El microcontrolador ATmega1281 es parte de la serie Atmel AVR y a menudo está equipado con nodos sensores para su uso en WSN. Posee una CPU de 8 bits, frecuencia de reloj de

16 MHz, 128 Kbytes de ROM y 8 Kbytes de SRAM. Emplea la arquitectura de Harvard, cuyos datos y programas son independientes. (Toriumi, Kakizaki, & Iwamura, 2011). Este microcontrolador trabaja con Wasmote de Libelium (Hacks, 2016).

1.5 Sensor

Un nodo sensor de una WSN sirve para recolectar información o tomar datos de un medio físico de su entorno, se necesita de sensores para medir el parámetro que se necesite. Un sensor es un dispositivo que permite detectar variables de instrumentación físicas o químicas, y transformarlas en variables eléctricas (Bosmediano & Ochoa, 2017).

En (Ling et al., 2017) se describe el estudio de un sistema de conexión inteligente, la arquitectura del enchufe inteligente Edimax SP-2101W (ver Figura 2), es un sensor de energía que cumple diferentes funciones, una de ellas es la función de gestión de energía para controlar con precisión el consumo de energía, todo controlado desde su teléfono. Este tipo de sensor está ganando popularidad en la domótica, permitiendo a los usuarios monitorear a sus hogares de forma remota. Además, describe algunas pautas para asegurar que terceras personas no vulneren la seguridad de este sensor.

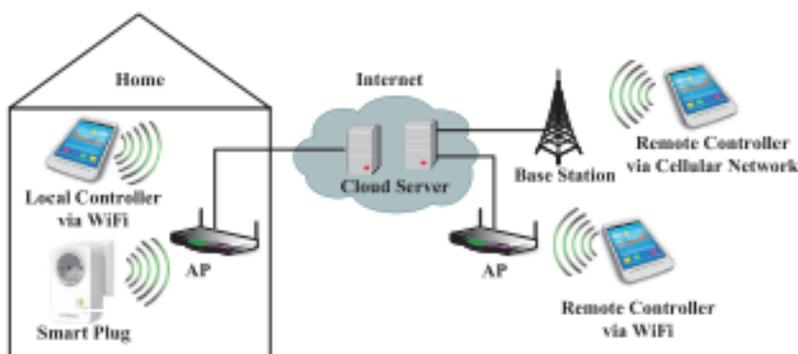


Figura 2. Arquitectura de el enchufe inteligente Edimax SP-2101W

Fuente: (Ling et al., 2017)

Elaboración: (Ling et al., 2017)

1.6 Técnicas de modulación

Existen diferentes tipos de modulación, pero solo algunas son utilizadas para minimizar el consumo de energía y a su vez la calidad de servicio, para ello maximizan la vida útil de la red considerando su eficiencia energética, en (Fateh & Govindarasu, 2015) se emplea la técnica de modulación MQAM (Modulación de amplitud en cuadratura M-aria) para el cálculo de la energía de transmisión de un nodo que conforma la red inalámbrica.

Según (Vinodhini & Priya, 2017) la medida de distancia entre los nodos es importante, debe ser más preciso. La técnica de modulación es necesaria para modular el valor de la distancia, debería ser la mejor en cuanto a consumo de energía e interferencia. Este documento estudia el rendimiento de diferentes técnicas de modulación, como MQAM, MPSK y MFSK.

Según (Vasanthi, Kumar, & Rama Rao, 2013) las técnicas de modulación deben ser eficientes energéticamente ya que son una parte fundamental para la comunicación de sensores inalámbricos, donde la eficiencia energética afecta directamente la vida útil de las redes. Los componentes indispensables para redes WSN son los transceptores de bajo consumo. En este documento simulan la técnica de modulación AQ-DBPSK y analizan su rendimiento para la comunicación de sensores inalámbricos en entornos interiores bajo el canal de ruido Gaussiano blanco aditivo y las condiciones de desvanecimiento de Raleigh.

CAPÍTULO II

ARQUITECTURA DE WSN

Es importante describir a la WSN debido a que en esta investigación juega un papel importante en el diseño de la red, además se vincula con redes inteligentes. La WSN contiene diferentes elementos y características que permiten establecer la interconexión de dos o varios nodos. Estos elementos influyen en el rendimiento de la red, convirtiéndola en una red robusta, además de soportar un mayor volumen de tráfico.

2.1 WSN

Una red de sensores inalámbricos es un sistema consta de un gran número de nodos sensores de bajo costo y bajo consumo de potencia, que son requeridos para operar en un entorno de interés por una duración máxima sin intervención humana, cuyo objetivo es monitorear, analizar y obtener datos a través del despliegue de los nodos como se observa en la Figura 3. Cada nodo que conforma esta red es un dispositivo miniatura y autónomo por lo que luego de ser configurados se los puede ubicar en cualquier lugar o área que se desee evaluar (Fateh & Govindarasu, 2015; Khajuria & Gupta, 2015).

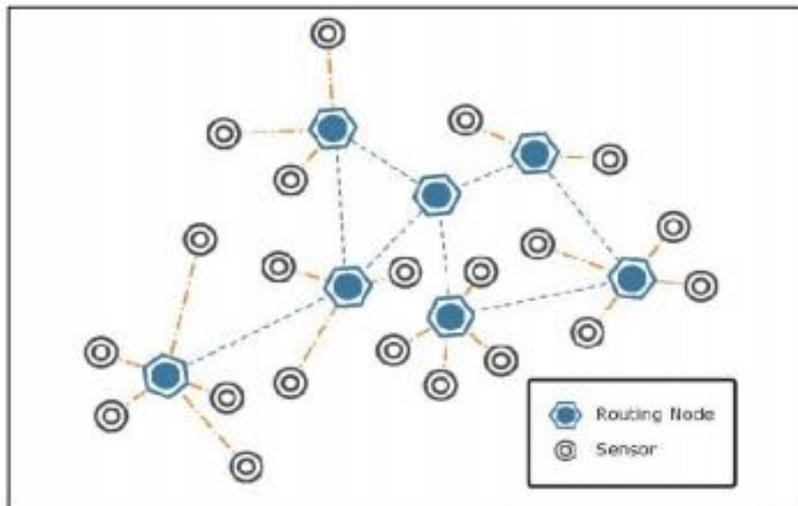


Figura 3. Representación de una WSN

Fuente: (Castro & Losilla, 2014)

Elaboración: (Castro & Losilla, 2014)

2.1.1 Características

- Unidades autónomas que constan de uno o varios sensores, una fuente de energía, microprocesadores y transceptores de radio. Es decir que detectan y procesan datos (Jun Zheng & Jamalipour, 2009).

- Posee sensores pequeños, baratos e inteligentes y de bajo consumo de energía, que pueden estar desplegados en una estructura de una red o en un área física (Mao, Fidan, & Anderson, 2007).
- Reducen el costo y el tiempo de la implementación de la red.
- Pueden ser implementadas en cualquier entorno, en especial en donde es de fácil acceso, esto facilita el desarrollo de aplicaciones en diferentes áreas de investigación.
- Los nodos sensores permiten interactuar entre un entorno doméstico y el usuario final (Sohraby, Minoli, & Znati, 2007; Jun Zheng & Jamalipour, 2009) .

2.1.2 Limitaciones

- Cada nodo tiene capacidades limitadas en cuanto al procesamiento y almacenamiento de información (Jun Zheng & Jamalipour, 2009).
- Vida útil limitada, debido a que cada nodo de la red emplea baterías, por ende su capacidad de energía es limitada (Jun Zheng & Jamalipour, 2009).
- Seguridad limitada.

2.2 Elementos de la red

Una red inalámbrica WSN se compone de varios elementos como los nodos sensores, sensores y actuadores, microcontroladores, tecnologías de comunicación, protocolos o estándares, Gateway, fuentes de alimentación entre otros.

2.2.1 Nodos

Los nodos denominados así por su tamaño pequeño por ende se las puede ubicar en cualquier entorno. Cumplen con requerimientos de autonomía, bajo consumo de energía, bajo costo y robustez. Cada nodo cumple con funciones de recolección de datos, de procesamiento, de comunicación lo que les permite interactuar con los demás nodos que conforman la red, debido a esto contiene elementos como: microcontrolador, una fuente de energía, un elemento radio-transceptor, una memoria externa y uno o más sensores, tal como se observa en la Figura 4. Todos estos elementos van añadidos a un módulo pequeño de bajo consumo de energía y bajo costo de implementación (Jun Zheng & Jamalipour, 2009).

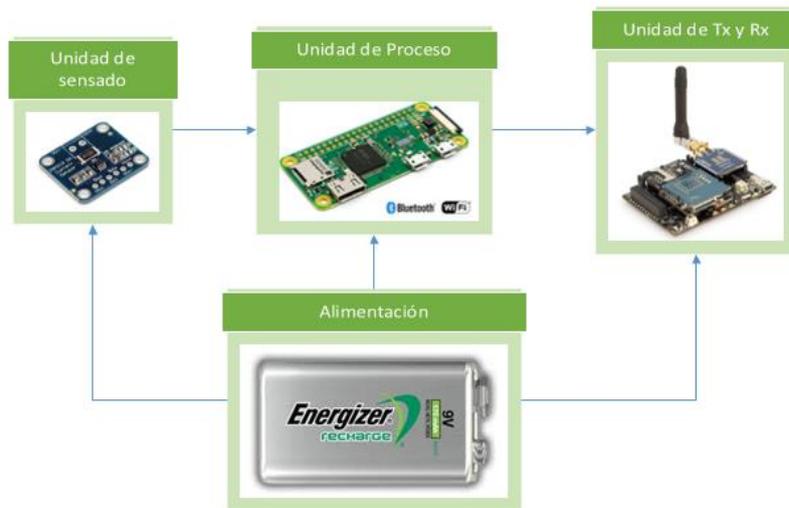


Figura 4. Estructura básica de un nodo sensor
 Fuente: (Jun Zheng & Jamalipour, 2009)
 Elaboración: Autor

2.2.2 Arquitectura de la red

La arquitectura de red es esencial para el diseño de WSN, en especial para su consumo de energía. La WSN por lo general trabaja con diferentes topologías, pero es importante recalcar que las principales son: punto a punto, estrella, malla y anillo.

- Topología punto a punto: es una de las más simples, establece la conexión de dos nodos, uno cumple la función de coordinador o nodo central y el otro de nodo sensor. Tal como se observa en la Figura 5.

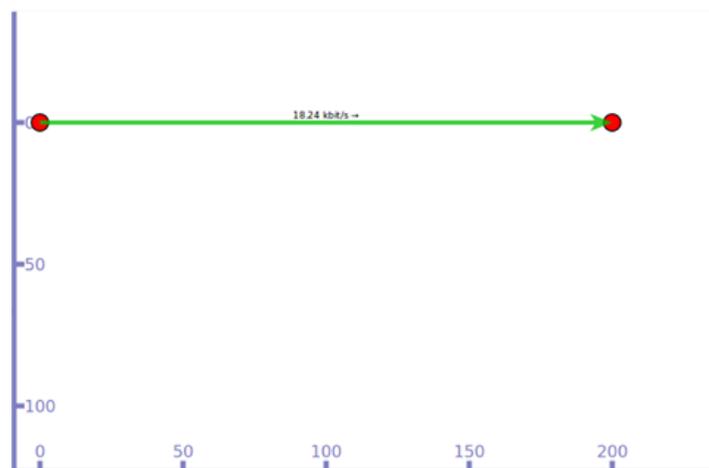


Figura 5. Topología punto a punto
 Fuente: Autor
 Elaboración: Autor

- Topología en estrella: Consta de un nodo central o estación base al que van conectados dos o más nodos, la función que cumple este nodo es recolectar la información que envía cada nodo como se observa en la Figura 6. Este decir cada nodo intercambia información e interactúa directamente con la estación base (Arias et al., 2008).

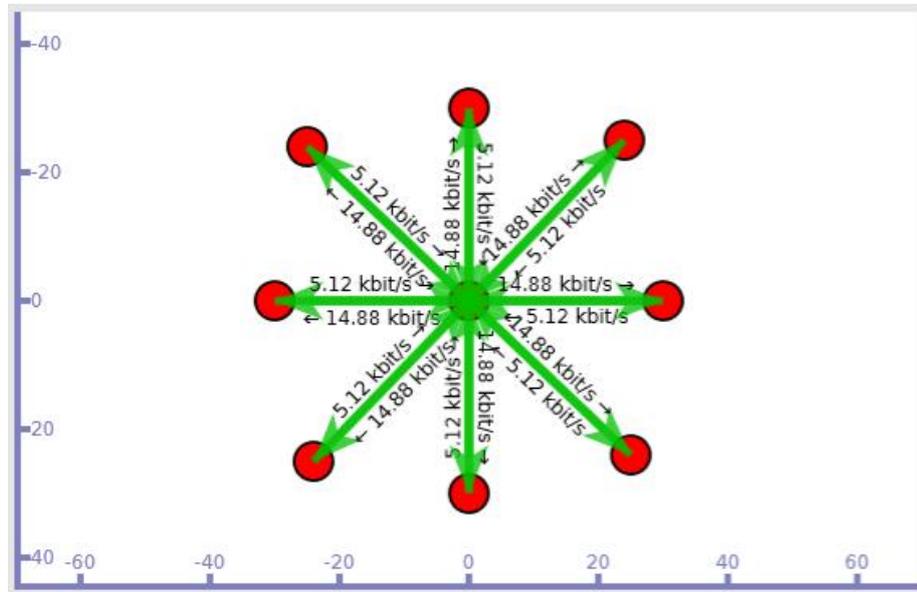


Figura 6. Topología de red en estrella
Fuente: (Arias et al., 2008)
Elaboración: Autor

Esta topología es muy simple y de bajo consumo debido a que los nodos establecen comunicación únicamente con el nodo central, y solo si este necesita información del nodo establecen conexión, lo que permite minimizar el consumo de energía. Un limitante es la difícil escalabilidad, es decir, que solo un número limitado de nodos se puede conectar, además de ello cada nodo se debe encontrar dentro del rango de cobertura del nodo central. Otra desventaja es que la red al emplear esta topología no es robusta como otro tipo de redes debido a la dependencia del nodo central o Gateway (Arias et al., 2008).

- Topología malla: conocida como red mesh, permite que cada nodo que pertenece a la red inalámbrica pueda intercambiar información con cualquier nodo de la misma, siempre y cuando esté a su alcance, es decir, no existe restricción alguna en cuanto a la transmisión y recepción de datos, emplea algoritmos de ruteo. Su principal ventaja es la redundancia y la escalabilidad de la arquitectura tal como se muestra en la Figura 7. Una desventaja es que aumenta el tiempo de entrega de la información, y con ello aumenta el consumo de energía (Arias et al., 2008).

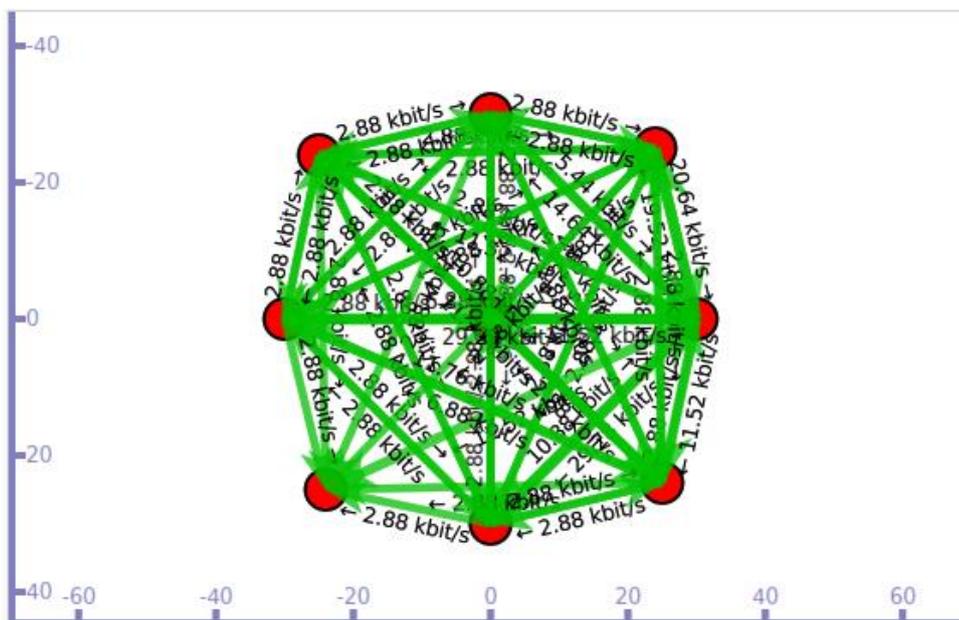


Figura 7. Topología malla
Fuente: (Arias et al., 2008)
Elaboración: Autor

- Las redes en modo ad hoc emplean un medio inalámbrico que permite que los nodos se comuniquen directamente (punto a punto) de forma libre y sin la necesidad de un punto de acceso. No tiene infraestructura, solo cuenta con estaciones base o puntos de acceso, es decir, son independientes. La estación base y el punto de acceso se comunican mediante funciones de enrutamiento, es por ello que requieren de protocolos (García, 2015).

2.3 Internet de las cosas

El internet de las cosas (IoT) son sistemas de infraestructura que permiten la interconexión de “cosas” u objetos cotidianos que generan, intercambian y procesan datos por medio de la red, es decir, cada objeto tiene dispositivos inteligentes integrados para establecer la conexión con la red y también interactuar con cualquier otro dispositivo o persona. Las “cosas” además de dispositivos electrónicos que se utilizan a diario pueden ser cualquier electrodoméstico o aparato industrial (Salazar & Silvestre, 2000).

Es definido como “Una red abierta y compleja de objetos inteligentes que tienen la capacidad de auto-organización, compartir información, datos y recursos, reaccionar y actuar frente a situaciones y cambios en el entorno”. También se la considera como una red global, a la que cualquier objeto se pueden conectar y comunicar de manera inteligente (Guang, Logenthiran, & Abidi, 2017).

2.4 Consumo de energía

“Es la cantidad de energía utilizada en un sistema” (Collins, 2012). La cantidad de energía que emplea un nodo o la red WSN en general para cumplir con sus funciones. Las redes WSN son de bajo consumo, pero a pesar de ello es el principal limitante. En la Figura 4 se puede observar que los elementos que conforman cada nodo consumen energía al cumplir cada una de sus funciones. En la Figura 8 se observa el consumo de energía de un nodo sensor al adquirir, transmitir y recibir datos e incluso cuando el nodo esté en reposo. Una solución para ayudar a minimizar el consumo de energía, es que el nodo de forma recurrente pase al estado “sleep”, es decir, se mantiene inactivo para ahorrar energía.

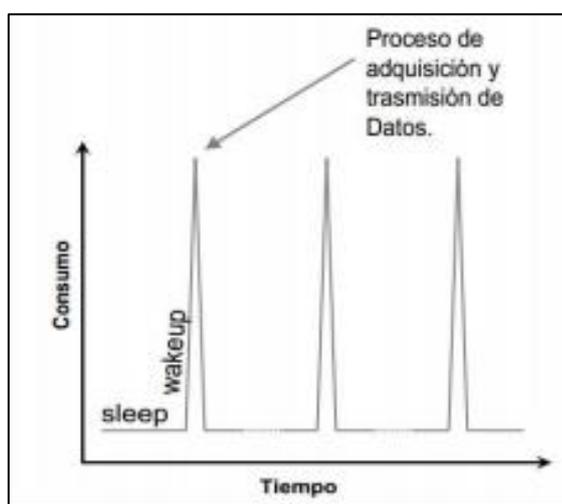


Figura 8. Consumo de energía de un nodo sensor.

Fuente: (Castro & Losilla, 2014)

Elaboración: (Castro & Losilla, 2014)

2.5 Software para WSN

Algunas herramientas o plataformas de red han permitido dar diferentes soluciones a diferentes aplicaciones, investigaciones o problemas que se relacionen con la red. Sirven para comprobar teorías sobre el comportamiento de las redes o simplemente para probar su funcionamiento, es por ello que es de vital importancia el uso de un simulador eficiente que permita encontrar una o varias soluciones.

Existen algunas plataformas de software para WSN que abordan y brindan múltiples servicios, permitiendo simular el comportamiento de dispositivos en tiempo real, algunas de estas herramientas son pagadas, pero también hay alternativas de licencia libre o gratuitas. Debido a esto se debe conocer a detalle las características y funcionalidades de las plataformas más utilizadas en el campo investigativo.

Un problema recurrente en una red inalámbrica es su consumo de energía, ya que cada nodo de la red funciona con baterías, es el punto clave de investigaciones sobre WSN y para ello se debe elegir el simulador de red adecuado que permita obtener resultados que validen, comparen, analicen y sobre todo se realicen evaluaciones experimentales de la red en general. Se describen y comparan cuatro de los simuladores de red más populares que ofrecen soporte para redes inalámbricas, y se elige el simulador que parece ser el más apropiado para el desarrollo de la simulación:

2.5.1 Network Simulator 3

Conocido como ns-3, no es compatible con su versión anterior ns-2, es un simulador de red de eventos discretos con licencia GNU GPLv2 y distribuido bajo licencia libre, fue diseñado con fines de investigación y uso educativo. Es desarrollado en C++ aunque permite el interfaz con lenguajes de alto nivel como Python, es decir, se puede realizar simulaciones empleando el lenguaje de programación C++ o Python, compuesto por cientos de archivos agrupados en módulos (Mattei, 2014).

Este simulador de red es empleado como un emulador de red en tiempo real. Existen algunas versiones de este software, actualmente el simulador cuenta con la versión 3.29 (Patel & Kamboj, 2015; Tazaki et al., 2011). Por medio de esta plataforma se puede estudiar, analizar el rendimiento y comportamiento de la red bajo determinadas condiciones. Es utilizado en investigaciones relacionadas con redes inalámbrica, en especial con redes ad-hoc, debido a que permite simular grandes escenarios con diversos usuarios y simula protocolos unicast y multicast. Se lo puede instalar en sistemas operativos Linux, en Windows por medio de Cygwin. Presenta algunas características que lo diferencian de otros simuladores de red, que se mencionan a continuación (García, 2015; Patel & Kamboj, 2015):

- Contiene múltiples bibliotecas que pueden combinarse entre sí.
- Contiene un núcleo sólido de simulación adaptable a las necesidades que el usuario desee realizar en la red sea inalámbrica o cableada, cada modelo de simulación es realista lo que permite que ns-3 emule la red en tiempo real.
- Este simulador de red está en constante innovación al igual que la tecnología, se adapta a nuevos modelos, dispositivos y redes. Soporta simulaciones de redes o tecnologías como Ipv4, Ipv6, Wireless (Adhoc, Wifi, Wimax, etc) y algoritmos de enrutamiento como AODV, OLSR, entre otros.

2.5.2 OMNET ++

Omnet ++ (Objective Modular Network Testbed en C++) es un simulador de red con de eventos discretos modulares orientados a objetos, con lenguaje de programación C++. Pese a que no es un simulador de red, ha obtenido gran popularidad debido a que es muy utilizado en diversos ámbitos como el modelado de redes, protocolos o arquitecturas de hardware (Xiaodong Xian, Weiren Shi, & He Huang, 2008).

Esta herramienta proporciona una arquitectura de varios módulos o componentes, consta de módulos simples y módulos compuestos. Estos módulos se programan en C++ y luego se ensamblan en módulos y modelos más sofisticados empleando un lenguaje de alto nivel (NED). Este software se puede ejecutar en Windows, Linux, Mac OS X y otros sistemas similares a Unix. Tiene el soporte a tecnologías inalámbricas, es de uso investigativo, posee una interfaz gráfica potente (Xiaodong Xian et al., 2008).

OMNeT++ es un simulador de red que le permite al usuario tener el control completo desde su entorno de programación. Además, es gratuito debido a que está compuesto por un conjunto de librerías, posee un entorno que permite la ejecución distribuida y paralela de la simulación.

2.5.3 Network Simulator 2

Es un simulador de red versión 2 fue desarrollado en C++ basado en eventos discretos. Es un software libre y orientado a objetos, es decir, existe una gran comunidad de colaboradores que permiten que el software mejore y desarrolle. Es utilizado para modelar el comportamiento básico de la red en general, todo esto es posible gracias a la ayuda de OTcl se puede describir y especificar características de la red importantes para la simulación, como la topología de red (Patel & Kamboj, 2015; Xiaodong Xian et al., 2008).

Esta plataforma es utilizada tanto en el campo investigativo como educativo a pesar que no cuenta con herramientas de visualización, es posible simular gran variedad de redes IP, cableadas como inalámbricas. Se puede implementar protocolos de red como TCP, UDP y protocolos multicast o protocolos de capa MAC para redes de área local (Patel & Kamboj, 2015; Xiaodong Xian et al., 2008).

2.5.4 OPNET

Esta plataforma de software permite simulaciones prácticas de redes, que incluye módulos de visualización y recopilación de datos de rendimiento. Permite el diseño de diferentes

topologías de redes inalámbricas o cableadas. El usuario puede simular, analizar y diseñar cualquier tipo de red, protocolos, dispositivos y aplicaciones. Herramienta de diseño y planificación de múltiples redes y proveedores, además de ello es para uso comercial (Patel & Kamboj, 2015).

2.5.5 Comparación de herramientas para la simulación

Existen algunas plataformas de red que son utilizadas para analizar redes inalámbricas orientadas a IoT como las definidas en el ítem anterior. En la Tabla 1 destaca algunas características importantes, con el fin de elegir un simulador de red. Se analiza y se elige bajo ciertos parámetros como la popularidad, es decir si es bastante utilizado por los investigadores, o conocer su disponibilidad, si cuenta con alguna interfaz gráfica, si es escalable, un parámetro importante si permite analizar el consumo de energía, si cuenta con soporte a redes inalámbricas, y demás parámetros que ayudan a tomar esta decisión:

Tabla 1. Comparación de algunas plataformas de red.

Característica / Software de red	OMNet++	ns-2	ns-3	OPNET
Analizador de Interface	GUI	NAM	NetAnim, Pyviz	GUI
Tipo de licencia	Libre	Libre	Libre	Comercial
Lenguaje de programación	C++	C++. Otcl	C++, Python	C, C++
Protocolos inalámbricos	802.11a/b/g, 802.16e	802.11a/b/g/e, 802.16	802.11a/b/g/e, 802.16, LTE	802.11a/b /e/g, 802.16
Modos de operación	Infraestructura y ad-hoc	Infraestructura y ad-hoc	Infraestructura y ad-hoc	Infraestructura y ad-hoc
Sistema Operativo	Windows, Linux, Mac	Linux, Mac, Windows, Free BSD	Linux, Mac, Windows, Free BSD	Windows
Uso investigativo	Alto	Alto	Alto	Alto
Presentación de resultados	Aceptable	No tiene	Excelente	Aceptable

Escalabilidad	Amplia	Reducida	Amplia	Amplia
Analizar el consumo de energía	Eficiente	Reducido	Eficiente	Eficiente
Soporte de redes inalámbricas	Medio	Bueno	Alto	Bueno

Fuente: (Patel & Kamboj, 2015; Xiaodong Xian et al., 2008)
 Elaboración: Autor

El software que cumple con la mayoría de las características es OPNET, pero es de licencia tipo comercial. Se descarta ns-2 debido a que es ineficaz para analizar el consumo de energía y por falta de la herramienta de visualización para la evaluación de los resultados. OMNet++ y ns-3 cumplen con todas las características de la Tabla 1, pero se eligió ns-3 debido a que es eficiente, flexible, de código abierto y es ampliamente utilizado, por estas razones se concluye que es la solución más adecuada para el desarrollo de este proyecto. Además, se planea superar su falta de herramientas de visualización mediante Excel.

CAPITULO III

PROTOCOLOS Y TECNOLOGÍAS DE COMUNICACIÓN PARA WSN

Las tecnologías de comunicación y protocolos para una red inalámbrica han tenido un gran crecimiento y desarrollo, brindan a objetos funciones de adquisición y procesamiento de información. Facilitan la transmisión de información, ya sea datos, voz o video de manera confiable, viable y eficiente a corto y largo alcance. Existen algunas tecnologías y protocolos orientadas a este tipo de red inalámbrica.

3.1 Tecnologías de comunicación inalámbrica

Es importante recalcar que son esenciales para la WSN, proveen a cada dispositivo o nodo la capacidad de permanecer conectados de forma indefinida o por un tiempo establecido. Se define su cobertura, el alcance y precisión del sistema. Este tipo de tecnologías comúnmente es de fácil instalación, permite movilidad, son escalables, adaptables, poseen altas tasas de transmisión entre otras características. Debido a todo esto algunas de estas tecnologías están incrustadas en los diferentes dispositivos y accesorios. Brindan soluciones inalámbricas para casas, edificios o ciudades, facilitándoles la comunicación sin importar la distancia. Existen diversas tecnologías inalámbricas, las más populares en WSN son: Bluetooth, Zigbee y Wi-Fi. A continuación, se describen y detallan algunas características de las tecnologías más utilizadas, su elección es debido a su bajo consumo energético y bajo costo de implementación (ver Tabla 2):

3.1.1 Wi-Fi (IEEE 802.11)

Tecnología fue diseñada por IEEE ofrece una conexión sin cables, a varios dispositivos dentro del área establecida. Esta incorporado en casi todos los dispositivos electrónicos debido a que es compatible con una amplia gama de aplicaciones e investigaciones realizadas en diferentes campos de la industria, se basa en un estándar de red inalámbrica IEEE 802.11x, esta familia de protocolos de paquetes define la transmisión de datos y administra el acceso a la red de área local empleando señales de radio. Esto permite a los usuarios moverse alrededor del área de cobertura y a su vez mantener la conexión de la red. Transmite hasta 100Mb/s, lo que permite compartir archivos de gran tamaño (García, 2015; Guang et al., 2017).

Wi-Fi está conformado por una familia de estándares o normas definidos para redes inalámbricas basados en especificaciones técnicas IEEE 802.11. Existen varias versiones de este estándar desde el año 1997 hasta la actualidad, brindando un ancho de banda, una cobertura y además garantiza la compatibilidad entre dispositivos, es decir es eficiente (Rodríguez & Padilla, 2015).

3.1.2 Bluetooth (IEEE 802.15.1)

Esta tecnología se refiere a una especificación para la comunicación inalámbrica de datos y voz de corto alcance, baja potencia, bajo costo y con dispositivos baratos. Bluetooth en la actualidad está en la mayoría de los dispositivos inteligentes, periféricos de datos, periféricos de audio y aplicaciones integradas, por ejemplo, en autos, facilitando el envío y recepción tanto de datos como voz sin la necesidad de cables y conectando los dispositivos a través de una conexión ad hoc por radio. Esta especificación es parte del estándar IEEE 802.15.1, posee un radio de cobertura de aproximadamente 10m, sin embargo, puede alcanzar rangos de hasta 100m con un consumo de corriente de 40mA. (McDermott-Wells, 2005).

3.1.3 ZigBee (IEEE 802.15.4)

Protocolo desarrollado con el fin de abordar las necesidades de fácil aplicación para redes inalámbricas de área personal de baja velocidad de datos, de bajo costo y bajo consumo de energía. Esta especificación es parte del estándar IEEE 802.15.4 definido por IEEE, es una solución inalámbrica que permite interconectar dispositivos remotos, también permite crear redes inalámbricas grandes que no requieran una gran cantidad de transmisión de datos. Facilita la comunicación a aplicaciones que requieren una comunicación segura, con una baja tasa de transmisión y por ende bajo consumo energético. Es un estándar abierto, con alta fiabilidad, baja capacidad de transmisión lo que permite que se diseñen sistemas de red de bajo costo y consumo. Soporta tres topologías de red: estrella, árbol y malla (J Zheng & Lee, 2002).

En la Tabla 2 se describen algunas características de las tecnologías antes mencionadas, conociendo parámetros relevantes para el desarrollo de la simulación:

Tabla 2. Comparativa entre tecnologías de comunicación Inalámbrica.

Parámetros/ Estándar	Bluetooth IEEE 802.15.1	Zigbee IEEE 802.15.4	Wi-Fi IEEE 802.11
Alcance	1-100m	1-100m	10-300m
Velocidad Tx	<1Mbps	<250kbps	11-54Mbps
Número de nodos	8	255/65535	+100

Consumo de Tx	40mA	30mA	400mA
Consumo en reposo	0.2mA	3 μ A	20mA
Costo	Medio	Bajo	Alto
Duración de batería	1 semana	100 días	12-48 horas
Latencia	14ms	240 μ s	1ms
Modulación	FHSS	DSSS	DSSS Y OFDM
Topología	Albol, Estrella, Mesh	Punto a punto, estrella, Mesh, Arbol	Punto a punto, Punto a multipunto, Mesh
Frecuencia	2.4GHz	2.4GHz	0.868GHz-0.915GHz-2.4GHz- 5GHz

Fuente: (Alipio et al., 2017; Edwards, 2012)
 Elaboración: Autor

3.2 Protocolos para WSN

Existen diferentes protocolos que cumplen diferentes funciones en la red WSN, en especial en redes inalámbricas sin infraestructura o redes ad-hoc, desde el enrutamiento hasta la transmisión y recepción de datos, este proyecto trabaja con algunos de ellos: OLSR, DSDV, AODV y DSR. En diferentes investigaciones se han analizado y comparado protocolos de enrutamiento relacionados específicamente con redes ad-hoc evaluando su desempeño.

Se analizó el comportamiento de la red en cuanto a consumo de energía con los protocolos antes mencionados. Comparando la eficiencia energética que existe con cada protocolo en una red WSN fija, ubicada en el hogar. Según (Feeney & Nilsson, 2001) para evaluar el consumo de energía de una red por medio de protocolos es conveniente analizar el comportamiento de consumo de energía de las interfaces inalámbricas. Se analiza una serie de datos presentados como ecuaciones lineales para calcular la energía consumida en el envío, recepción y descarte de paquetes de datos para protocolos en redes adhoc.

3.2.1 OLSR

Es un protocolo de enrutamiento proactivo, se basa en las tablas de enrutamiento, es decir cada nodo de la red intercambia información de manera frecuente. Este protocolo busca

minimizar el número requerido de paquetes para ser transmitidos y también reducir su tamaño (Singh, Barkhodia, & Walia, 2012).

3.2.2 AODV

Este protocolo es reactivo considerado un protocolo de enrutamiento eficiente y soporta mecanismos de enrutamiento unicast y multicast basado en el algoritmo vector distancia. El algoritmo que emplea AODV puede cambiar la situación del enlace, es por ello que utiliza la técnica de bajo demanda para buscar rutas y así establecer el enlace. Reduce la sobrecarga de tráfico de control a costa de una mayor latencia en el descubrimiento de nuevas rutas (Singh et al., 2012).

3.2.3 DSR

Es un protocolo reactivo, basado en enrutamiento de origen debido a que es un protocolo simple y eficiente, apropiado para redes inalámbricas ad hoc. El nodo de origen toma las decisiones en cuanto a enrutamiento o estado del enlace. Cumple dos funciones con el fin de construir las rutas, el descubrimiento y mantenimiento de la ruta (Singh et al., 2012).

3.2.3 DSDV

Es parte de los protocolos que utilizan algoritmos de vector distancia, es proactivo y requiere que cada nodo de la red transmita periódicamente actualizaciones de enrutamiento. Permite que cada nodo posea una tabla de enrutamiento en la que indica la cantidad de saltos que necesita, tomando en cuenta su destino. Es un protocolo eficiente y fiable, de bajo coste (Rodriguez & Padilla, 2015).

CAPITULO IV

DESCRIPCIÓN DE LA SIMULACIÓN DE REDES DE SENSORES

4.1 Simulador de red ns-3

El simulador ns-3 es elegido para simular escenarios de red con el fin de obtener graficas que permitan analizar diferentes características de red, por ejemplo, el consumo energético que existe en una red inalámbrica con ciertas variables, permitiendo conocer su comportamiento y rendimiento bajo ciertas condiciones al añadir ciertos protocolos o tecnologías. Es una herramienta de fácil uso y flexible, adecuado para simular escenarios de redes inalámbricas.

En este documento se detallan simulaciones centradas en el principal limitante de la WSN, el consumo de energía del dispositivo, un motivo por el que se eligió ns-3 es porque posee dos modelos de energía uno para el consumo de energía del dispositivo y otro para la fuente de energía. El modelo para el consumo de energía del dispositivo captura cómo se consume la energía de cada componente del nodo de la red, en cambio el modelo de fuente de energía simula el comportamiento de fuentes de energía como las baterías. Para tener un consumo real de energía es importante incluir modelos de consumo de energía de dispositivos para varios componentes de un nodo, como radios, procesadores o sensores (Wu, Nabar, & Poovendran, 2011).

Según (Anonymous, 2010; García, 2015; Ingalaturre, López, & Villajosana, 2018; Patel & Kamboj, 2015) este simulador puede emular redes en tiempo real, además, de adaptarse y cambiar constantemente a nuevos módulos, redes y dispositivos, se adapta a las necesidades que requiere analizar el usuario en cuanto al comportamiento de la red.

Esta plataforma de red permite simular modelos de redes de alto desempeño, es decir soporta simular redes inalámbricas que incluyen cada parámetro relevante que permite simular su funcionamiento para ello emplean:

- Sobre Ipv4
- Sobre Ipv6
- Wireless (Ad-hoc, WiFi, WiMax, etc)
- Algoritmos de enrutamiento

4.2 Diseño de la red

Para el diseño de la red se toma en cuenta la distancia de cada nodo, el tipo de microcontrolador, tecnologías a utilizar, protocolos y topologías, sensores para medir el

consumo de energía, en el trabajo de “DISEÑO Y CONSTRUCCIÓN DE UN HUB DE MEDICIÓN DE CONSUMO DE ENERGÍA ELÉCTRICA PARA USO DOMÉSTICO”, desarrollado bajo la dirección del docente Tiesman Daniel Castillo, se presenta un prototipo de medidor inteligente para el hogar con el fin de recolectar datos sobre el consumo de energía para luego ser subidos a la red (Peréz, 2019).

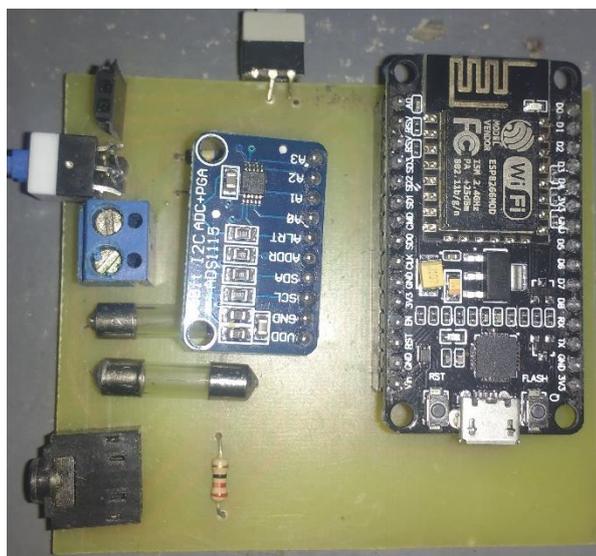


Figura 9. Placa del circuito del medidor Inteligente
Fuente: (Peréz, 2019)
Elaboración:(Peréz, 2019)

La función principal de este medidor (ver Figura 9) es cuantificar el consumo de energía empleando un microcontrolador nodeMCU esp8266, dos sensores uno de voltaje y el otro tipo pinza para medir la corriente. Para subir estos datos a una plataforma web lo hace a través de una red inalámbrica (Peréz, 2019).

También emplea un sistema de gestión de energía que busca la eficiencia energética en el circuito implementado para informar al cliente sobre consumo eléctrico generado por cada dispositivo electrónico que se encuentra en el hogar. Este módulo fue diseñado, implementado y validado, además los componentes electrónicos con los que se desarrolló son de bajo costo (Peréz, 2019).

4.3 Simulación

Para tener éxito en el desarrollo de la simulación se necesita cumplir con varios parámetros como los escenarios, la cantidad de nodos e incluso la distancia que se encuentra entre ellos y requisitos del software a utilizar.

4.3.1 Descripción de los parámetros de simulación

Para la simulación de las redes WSN en modo ad-hoc, se ha tomado en cuenta dos casos o escenarios. Como se va a analizar el consumo del hogar se ha tomado en cuenta una casa de dos pisos como escenario A y como escenario B un edificio de 3 pisos. Con las mismas características de topologías y protocolos detallados anteriormente, con el fin de apreciar diferencias que permitan analizar y dar una solución eficiente y fiable.

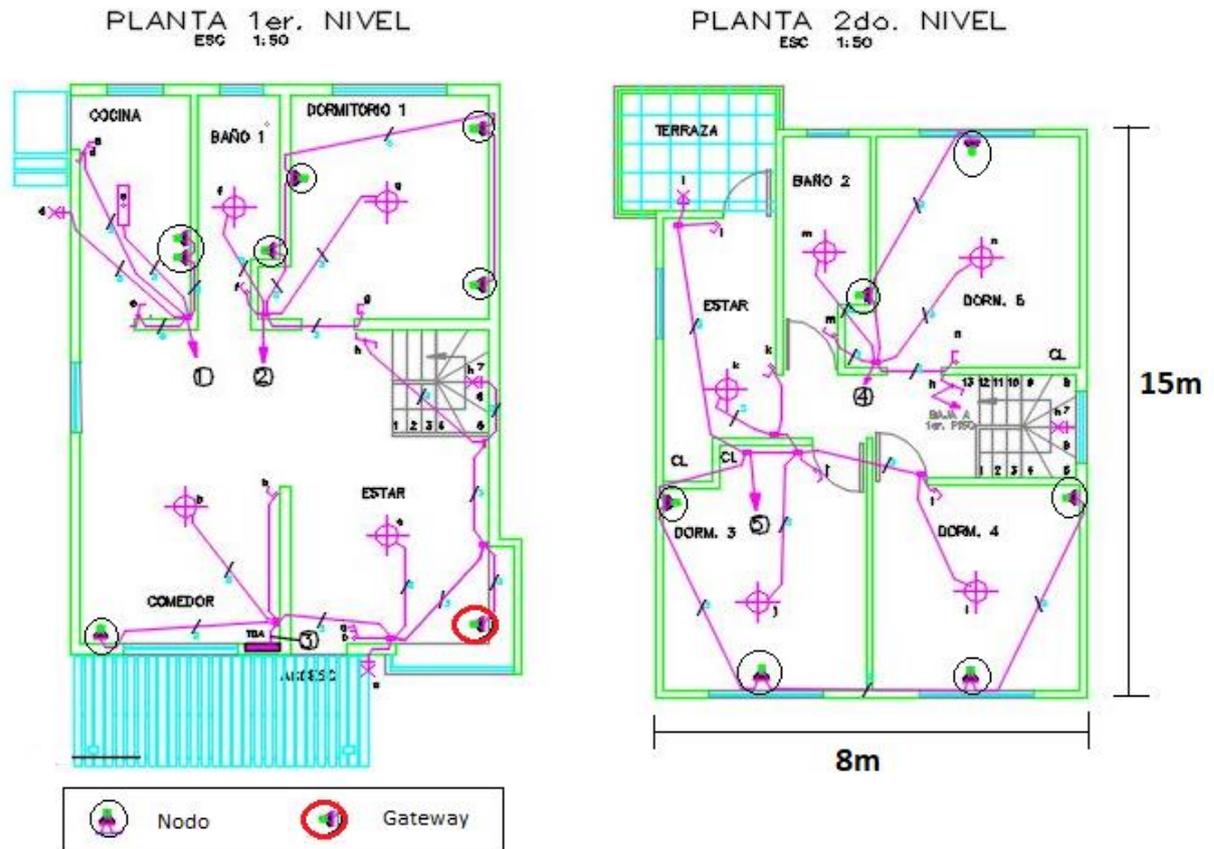


Figura 10. Escenario A, casa de dos pisos
 Fuente: (Mundaca, 2014)
 Elaboración: (Mundaca, 2014)

En la Figura 10, se observa el escenario A, un plano eléctrico de una casa de dos pisos, cada nodo va estar ubicado en cada enchufe existente en la casa los detalles que se utilizan para la simulación se muestran en la Tabla 3:

Tabla 3. Descripción de los parámetros de simulación del escenario A

Parámetro	Cantidad
Nodos	13

Gateway	1
Distancia máxima entre nodos	8x15m
Tiempo de la simulación	3600s
Tamaño de paquetes	1024 bytes
Tasa de transmisión	1Mbps
Paquetes	1000
Protocolos	OLSR-AODV-DSR-DSDV

Fuente: Autor
Elaboración: Autor

En la siguiente Figura 11 se observa el plano eléctrico de un edificio de 3 pisos conformado por 64 nodos y 1 gateway, cada nodo se encuentra ubicado en cada enchufe existente en el edificio, cada nodo tiene una distancia diferente pero la mínima es de 8m y la máxima de 16m.

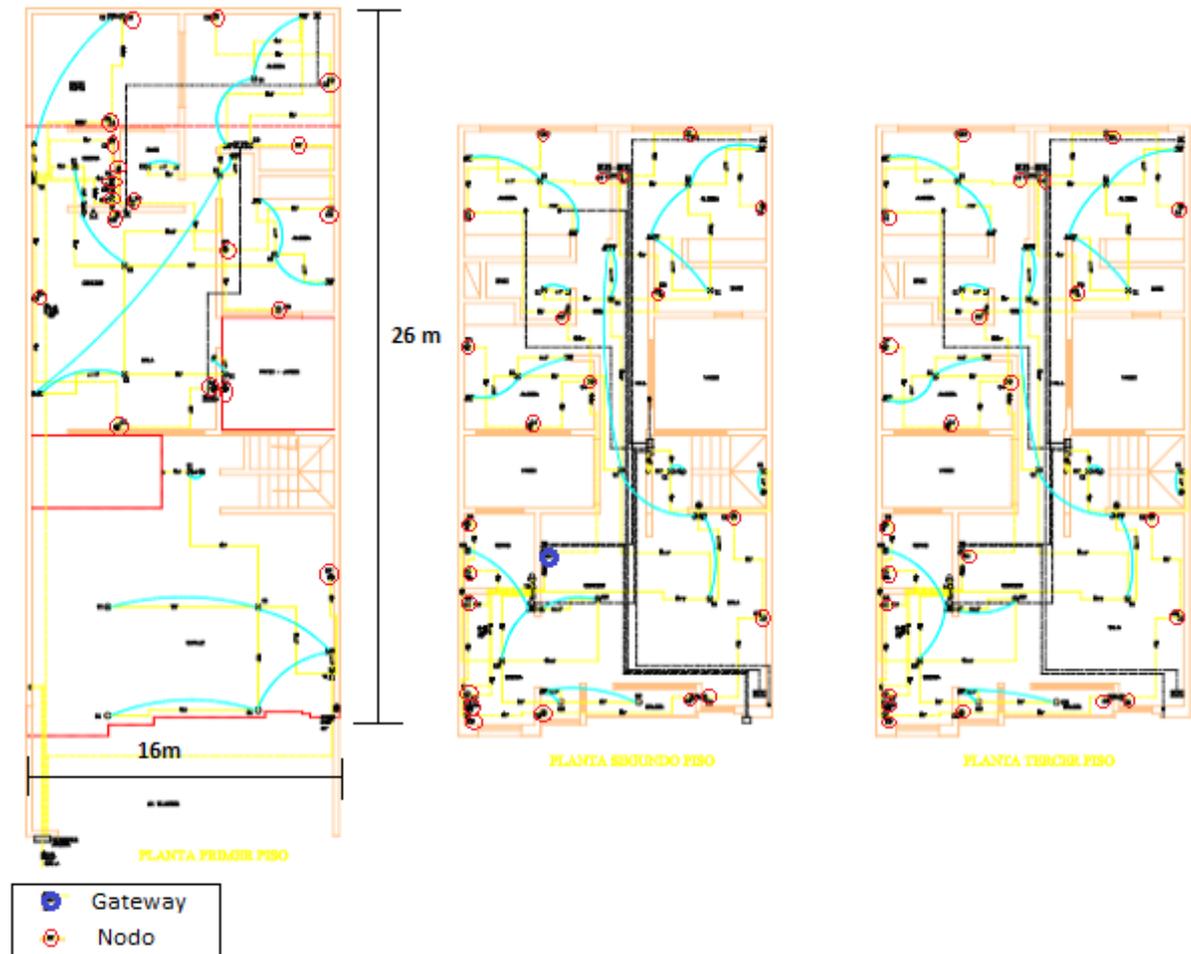


Figura 11. Escenario B, edificio de 3 pisos.
 Fuente: (Rivera & Gil, 2013)
 Elaboración: (Rivera & Gil, 2013)

En la Tabla 4 se observan las características principales que se utilizó para la simulación de este escenario en el cual se desplegarán alrededor de 64 nodos y 1 gateway en un edificio de 3 pisos.

Tabla 4. Descripción de los parámetros de simulación del escenario B

Parámetro	Cantidad
Nodos	64
Gateway	1
Distancia máxima entre nodos	16x26m
Tiempo de la simulación	3600s
Tamaño de paquetes	1024 bytes

Tasa de transmisión	1Mbps
Paquetes	1000
Protocolos	OLSR-AODV-DSR-DSDV

Fuente: Autor
Elaboración: Autor

4.3.2 Especificaciones del software

Se obtuvo resultados referentes al consumo de energía simulando una red inalámbrica WSN en modo ad hoc en la plataforma de simulación ns-3. Se describe en la Tabla 5 esta plataforma y algunas dependencias que se utilizaron para el desarrollo del proyecto.

Tabla 5. Especificación del software, dependencias y hardware.

Indicador	Software	Versión
Sistema Operativo	Ubuntu	16.04 LTS
Simulador de red	ns-3	3.26
Herramienta de análisis	NetAnim	3.107
Dependencias para ns-3		
Python, C++, Mercurial, Qt4-devel	Requerimientos de ns-3	
Hardware		
Equipo	Laptop Hp	
Memoria RAM	8GB	
Procesador	Intel Core I5-7200U CPU 2.50GHz 2.70GHz	
Disco duro	1TB	

Fuente: Autor
Elaboración: Autor

La instalación del simulador ns-3 en el sistema operativo antes mencionado es un punto clave para analizar la eficiencia energética de la red WSN en los escenarios propuestos, con el fin de cumplir con este objetivo se describe la instalación del simulador, y los módulos que este contiene junto con algunas características esenciales para la simulación:

4.3.3 Funcionamiento y estructura de ns-3

Se instaló ns-3 en Ubuntu 16.04 LTS desde su terminal, con el fin de simular los escenarios propuestos anteriormente. El simulador está conformado por diferentes partes tal como se observa en la Figura 12:



Figura 12. Funcionamiento de ns-3
Fuente: (ns-3, 2010)
Elaboración: Autor

Los módulos que forman parte del núcleo de esta herramienta de simulación son aquellos que aportan diferentes funcionalidades que cada entorno o escenario a simular lo requiera, estos módulos se observan en la Figura 13:

```
luli@luli-HP-Laptop-15-bs0xx:~/tesis/ns-allinone-3.26/ns-3.26$ ./waf
Waf: Entering directory `~/home/luli/tesis/ns-allinone-3.26/ns-3.26/build'
Waf: Leaving directory `~/home/luli/tesis/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.741s)

Modules built:
antenna          aadv             applications
bridge           buildings        config-store
core             csma             csma-layout
dsdv            dsr              energy
fd-net-device   flow-monitor     internet
internet-apps  lr-wpan          lte
mesh            mobility         mpi
netanim (no Python) network          nix-vector-routing
olsr            point-to-point  point-to-point-layout
propagation     sixlowpan        spectrum
stats           tap-bridge       test (no Python)
topology-read  traffic-control  uan
virtual-net-device visualizer        wave
wifi           wimax

Modules not built (see ns-3 tutorial for explanation):
brite           click            openflow

luli@luli-HP-Laptop-15-bs0xx:~/tesis/ns-allinone-3.26/ns-3.26$
```

Figura 13. Módulos existentes en ns-3
Fuente: Plataforma de ns-3
Elaboración: Autor

Un script desarrollado en ns-3 en la versión que se instale, contiene una serie de sentencias que son almacenadas en un archivo de texto y a la vez son interpretadas al ser ejecutadas desde el terminal del sistema operativo que se requiera en este caso Ubuntu, por medio de Waf una herramienta que permite configurar e instalar los requerimientos y librerías que requiere el software, además se puede compilar los diferentes códigos o simulaciones desarrollados en este entorno de simulación. Las diferentes partes de este script se presentan a continuación:

- **Texto estándar o informativo**

Al inicio del script se especifica algunos detalles relevantes que se deben dar a conocer como por ejemplo los derechos de autor, el tipo de código fuente, detalles como la topología y funcionamiento del script (ns-3, 2010):

```
/* Mode: C++; c-file-style:"gnu"; */
```

- **Módulos de inclusión:**

Al inicio del código es apropiado declarar los módulos de inclusión que se van a necesitar en el código y que se encuentran en el directorio de src, se los escribe de la siguiente manera (ns-3, 2010):

```
#include "ns3/core-module.h"
#include "ns3/command-line.h"
#include "ns3/network-module.h"
#include "ns3/wifi-module.h"
#include "ns3/applications-module.h"
#include "ns3/energy-module.h"
#include "ns3/internet-module.h"
#include "ns3/olsr-helper.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/mobility-helper.h"
#include "ns3/lpv4-address-helper.h"
#include "ns3/wifi-radio-energy-model-helper.h"
```

- **Descripción de nombres**

Por medio de la siguiente línea se implementa un espacio de nombres de C++ permite el manejo global de las variables descritas en el código e incluso se pueden integrar funciones de otros códigos (García, 2015; ns-3, 2010):

```
using namespace ns3;
```

- **Registro**

Utilizado la siguiente línea del código se declara un componente de registro de llamada, por ejemplo, FirstScriptExample activa y desactiva el registro de mensajes en la consola (ns-3, 2010):

```
NS_LOG_COMPONENT_DEFINE ( "FirstScriptExample");
```

Función principal

En esta parte del código se declara la función principal activando los componentes del registro, de la misma forma como se inicia un programa en C++ (García, 2015; ns-3, 2010):

```
int
```

```
    Main (int argc, char *argv[])
```

```
    {
```

```
        LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
```

```
        LogComponentEnable (
```

```
            "UdpEchoServerApplication", LOG_LEVEL_INFO);
```

```
        .....
```

- **Tipos de ayudantes de topologías**

Se definen varios tipos de ayudantes de topologías, son utilizados por el simulador para declarar variables, crear, gestionar y acceder nodos objetos, tipos de comunicación y redes de datos a ser configuradas, protocolos, crear las topologías, conectar los nodos entre sí, agrupar los dispositivos y nodos de la red (García, 2015; ns-3, 2010):

- NodeContainer
- PointToPointHelper
- NetDeviceContainer
- InternetStackHelper
- Ipv4AddressHelper

- **Aplicaciones**

Otro componente del simulador de red ns-3 es la Aplicación. Existen varios tipos de aplicaciones que se utilizan para configurar como se transmiten los datos por ejemplo UDP o TCP. Es decir, se utilizan protocolos de comunicaciones y formas de transmisión de datos, por ejemplo: OnOffApplication, UdpClientServer, PacketSink, Ping6, UdpEcho, entre otras (ns-3, 2010).

- **Simulador**

Se lo escribe al final del script es usado para ejecutar la simulación, el fin es ejecutar cada instrucción escrita o programada siguiendo el orden establecido. Esto se realiza con la siguiente línea de código (García, 2015; ns-3, 2010):

```
    Simulator::Run ();
```

Es necesario limpiar todos los procesos creados y ejecutados anteriormente para que se puedan ejecutar los procesos siguientes, para ello se escriben las siguientes líneas de código:

```
Simulator :: Destroy ();
```

```
Return 0;
```

4.3.4 Módulo de energía en ns-3.26

Recalcando la importancia de investigar el consumo de energía en los dispositivos inalámbricos que forman parte de una red WSN, el framework de energía de ns-3 está compuesto por dos clases importantes: la Fuente de energía (EnergySource) y Modelo de energía del dispositivo (DeviceEnergyModel) que están relacionadas por medio de una serie de interfaces predefinidas tal como se observa en la Figura 14 (Anonymous, 2010; Ingalaturre et al., 2018; Wu et al., 2011).

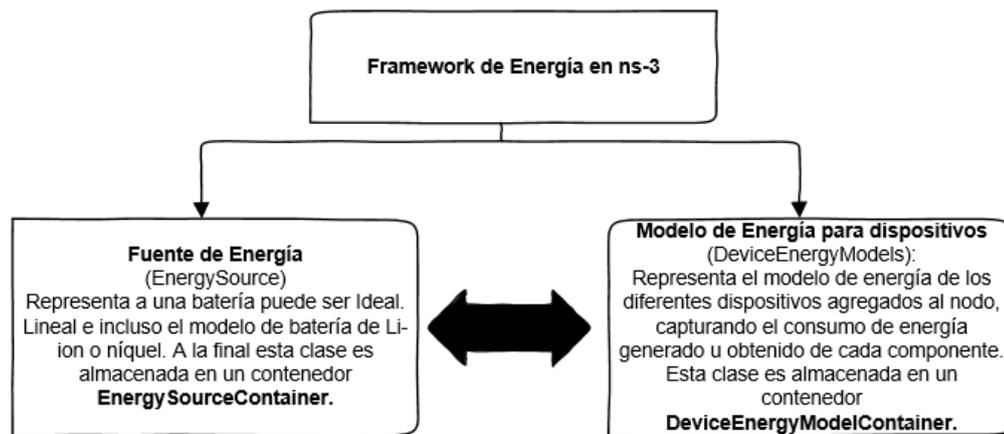


Figura 14. Framework de Energía
Fuente: (Anonymous, 2010; Ingalaturre et al., 2018; Wu et al., 2011)
Elaboración: Autor

La clase EnergySource representa una fuente de alimentación o batería en cada nodo. Un nodo puede tener una o más fuentes de energía y a su vez cada fuente puede conectarse a varios modelos de energía de los dispositivos. Su funcionalidad es brindar energía para los dispositivos del nodo. Representa diferentes tipos de fuentes de alimentación principalmente a las baterías de Ion-litio, de níquel etc., son almacenadas en un contenedor denominado EnergySourceContainer (Anonymous, 2010; García, 2015; Wu et al., 2011).

El modelo de energía del dispositivo en el nodo consume energía de la fuente de energía, la clase DeviceEnergyModel representa el modelo de consumo de energía de uno o diversos dispositivos, se almacenan en un contenedor denominado DeviceEnergyModelContainer. En

caso de que se haya agotado la energía o presente algún problema la fuente de energía tiene la capacidad de notificar a cada modelo de energía (Anonymous, 2010; Wu et al., 2011).

Según (Ingalaturre et al., 2018) el modelo de fuente de energía puede contener un perfil de descarga lineal lo que permite reducir la carga computacional y la complejidad de codificación del modelo. Al establecer un modelo de consumo de cada dispositivo existente en el nodo se proporciona algunos valores de corriente en cada estado de operación del nodo: Tx, Rx, Standby y Sleep.

El módulo de energía además de contener estas dos clases, provee funciones de consulta a módulos externos, es decir, si un protocolo de enrutamiento o tecnología sensible a la energía desea conocer el nivel de energía del nodo lo puede hacer solicitando información a la fuente de energía. También existe un proceso para suministrar energía de forma externa por medio de las fuentes de energía renovable y a la vez almacenar esta energía en cada fuente de energía, está separado del módulo de energía con el fin de que se modele de forma individual denominado HarvestingEnergyModel. (Wu et al., 2011).

```
/** Energy Model **/  
/* energy source */  
BasicEnergySourceHelper basicSourceHelper;  
// configure energy source  
basicSourceHelper.Set ("BasicEnergySourceInitialEnergyJ", DoubleValue (5.0));  
// install source  
EnergySourceContainer sources = basicSourceHelper.Install (c);  
/* device energy model */  
DeviceEnergyModelContainer deviceModels = radioEnergyHelper.Install (devices, sources);  
WifiRadioEnergyModelHelper radioEnergyHelper;  
// configure radio energy model  
radioEnergyHelper.Set ("TxCurrentA", DoubleValue (0.0174));  
radioEnergyHelper.Set ("RxCurrentA", DoubleValue (0.0197));
```

Para modelar el consumo de energía se configura todos los parámetros y se los instala en los dispositivos virtuales. La clase BasicEnergySourceHelper permite configurar e instalar en cada nodo la fuente de energía y a la vez almacenar en un contenedor EnergySourceContainer. Se establece un valor de corriente en cada estado de operación de la red, por medio de TxCurrentA y RxCurrentA se configura la transmisión y recepción, el SwitchingCurrentA representa a la corriente consumida cuando se cambia entre los estados de transmisión y recepción, y el SleepCurrentA representa la corriente cuando el dispositivo está en modo de suspensión (García, 2015).

El modelo de energía para Wi-Fi y ZigBee integra este modelo en donde se configuran los diferentes parámetros de energía de los dispositivos y de la fuente mediante dos clases

principales heredadas, para Wifi son WifiEnergySource y WifiRadioEnergyModel, y para ZigBee ns-3 tiene integrado a Lr-wpan y su capa superior esta implementado 6LoWPAN, para Lr-wpan son Lr-wpanEnergySource y Lr-wpanRadioEnergyModel.

4.3.5 Resultados de la simulación

- **Primer escenario**

La Figura 15 muestra a la red inalámbrica diseñada para el escenario 1 conformado por 14 nodos desplegados en un área aproximada de 8x16 m, la forma un Gateway ubicado en el primer piso de la casa, que se comunica con los 13 nodos.

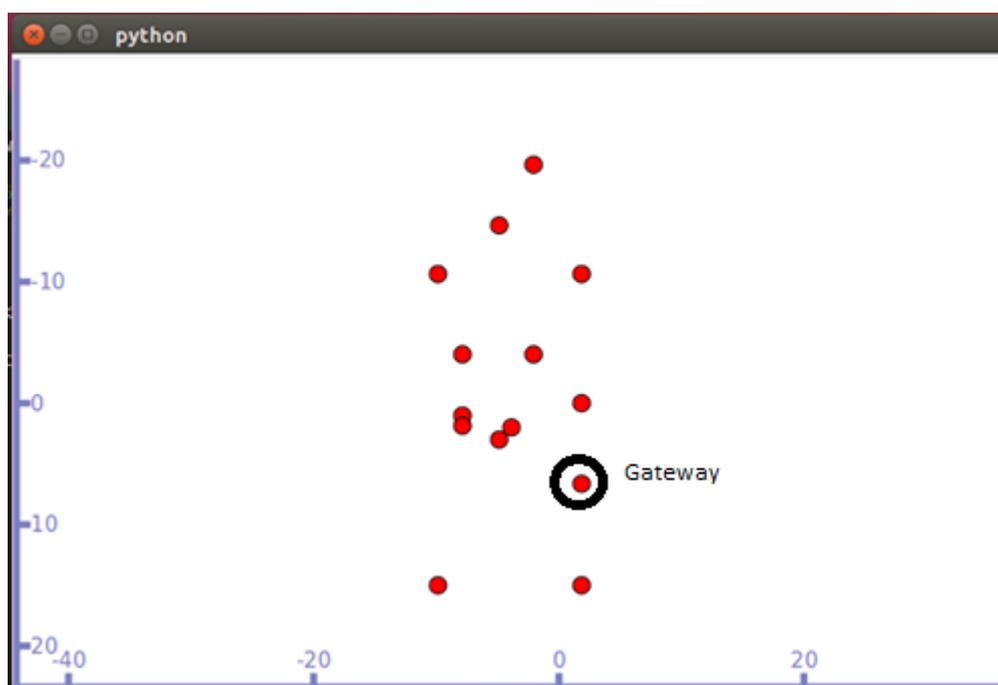


Figura 15. Despliegue del escenario A

Fuente: Autor

Elaboración: Autor

Los resultados de las simulaciones realizadas en relación al consumo de energía se obtuvieron en Joules (J) durante 1 hora. Se toma en cuenta la potencia de transmisión, recepción y cada estado de un nodo sensor. También se establece la tasa de transmisión de los datos, e incluso la información se divide en tramas o paquetes de 1024 bytes de datos cada uno y demás factores que influyen en el consumo de energía de cada nodo y de la red en general.

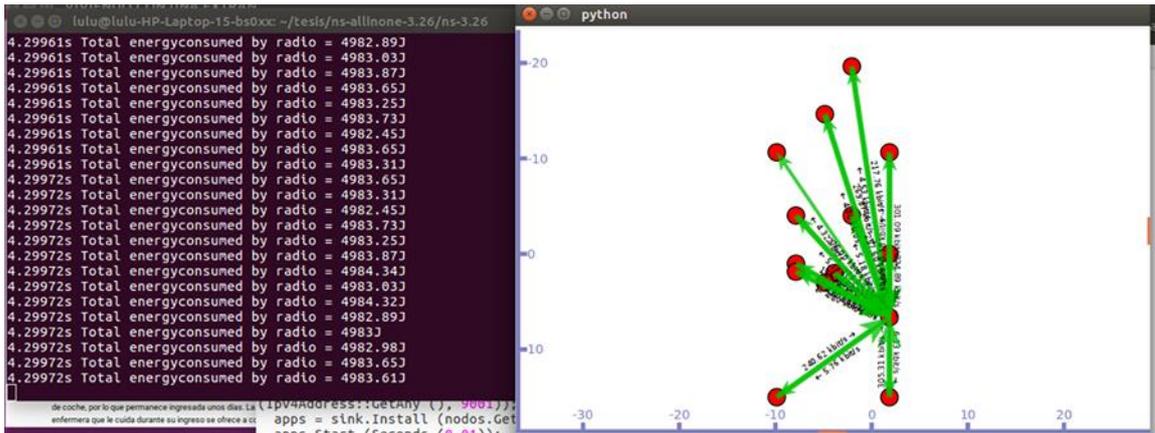


Figura 16. Visualización durante la transmisión y recepción de datos por medio de PyViz.
 Fuente: Autor
 Elaboración: Autor

PyViz es el visualizador de Python que es parte de ns-3 es el que permite observar la transmisión y recepción de datos desde los nodos al Gateway (ver Figura 16) y su respectiva confirmación. El consumo de energía es almacenado en un documento .log y además se lo puede visualizar en el terminal del Sistema Operativo.

Protocolo TCP:

En la Figura 17 se observa los resultados de la simulación en un intervalo de 60 minutos, con un consumo de energía máximo de 10.81W-h (Vatio-hora).

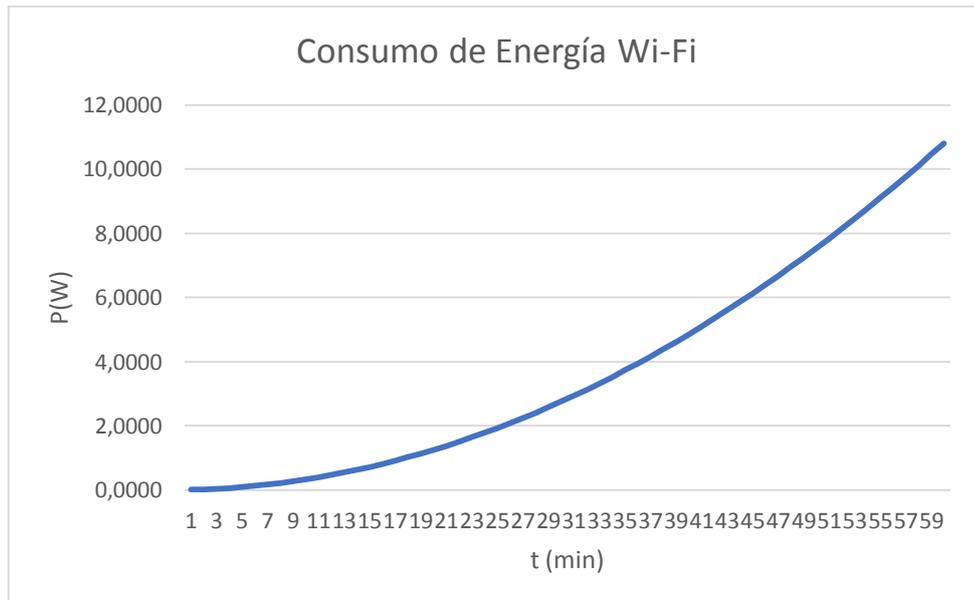


Figura 17. Consumo de energía del escenario A con Wi-Fi durante 1hora.
 Fuente: Autor
 Elaboración: Autor

Comparación de los protocolos

En la Figura 18 se muestra los resultados obtenidos de las simulaciones del escenario A, con los protocolos: AODV, OLSR, DSDV y DSR. El protocolo de menor consumo energético es OLSR, con un consumo máximo de 9.82Wh.

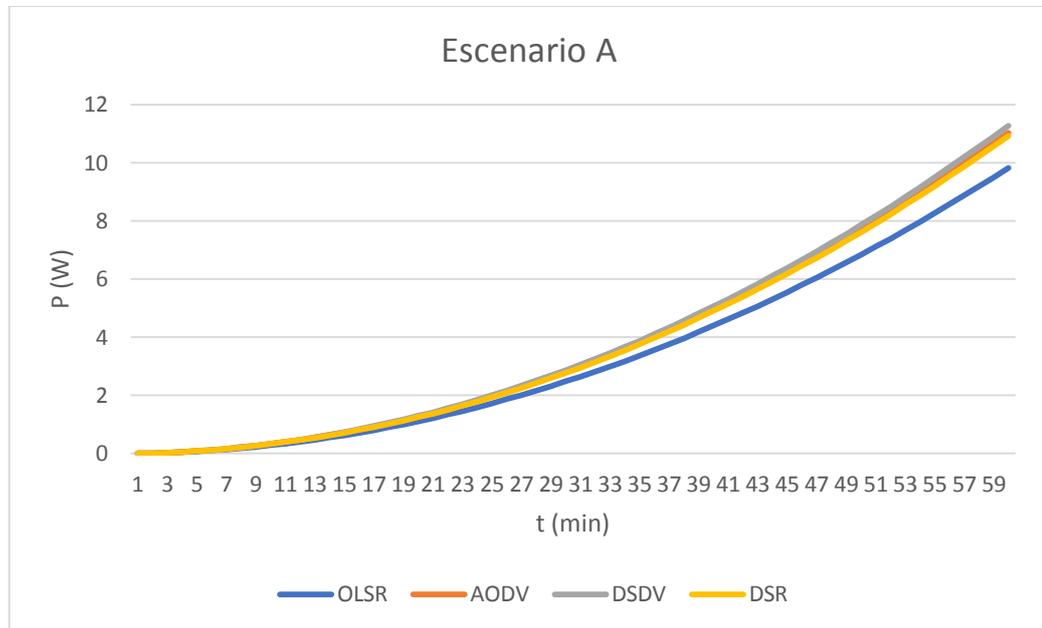


Figura 18. Consumo de energía de Wi-Fi con algunos protocolos, en un intervalo de 1h.

Fuente: Autor

Elaboración: Autor

- **Segundo escenario:**

El segundo escenario está conformado por 65 nodos en total distribuidos en el entorno presentado en la Figura 19, es fijo, es simulado con los parámetros detallados anteriormente:

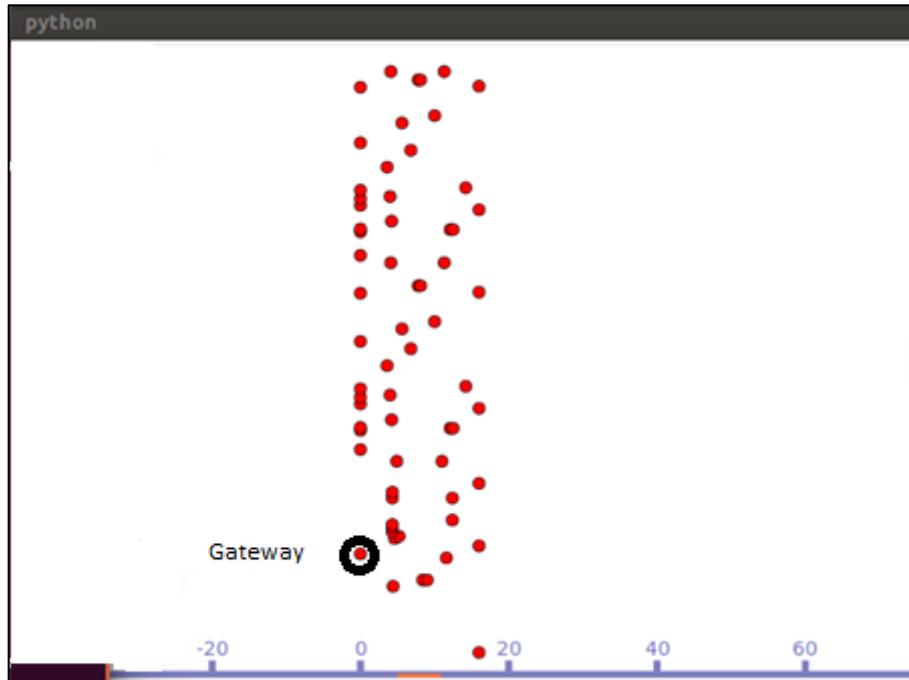


Figura 19. Simulación del escenario B
Fuente: Autor
Elaboración: Autor

En la Figura 20 se observa la transmisión y recepción de datos representando la topología estrella, el Gateway se encuentra a un extremo del plano del edificio del edificio.

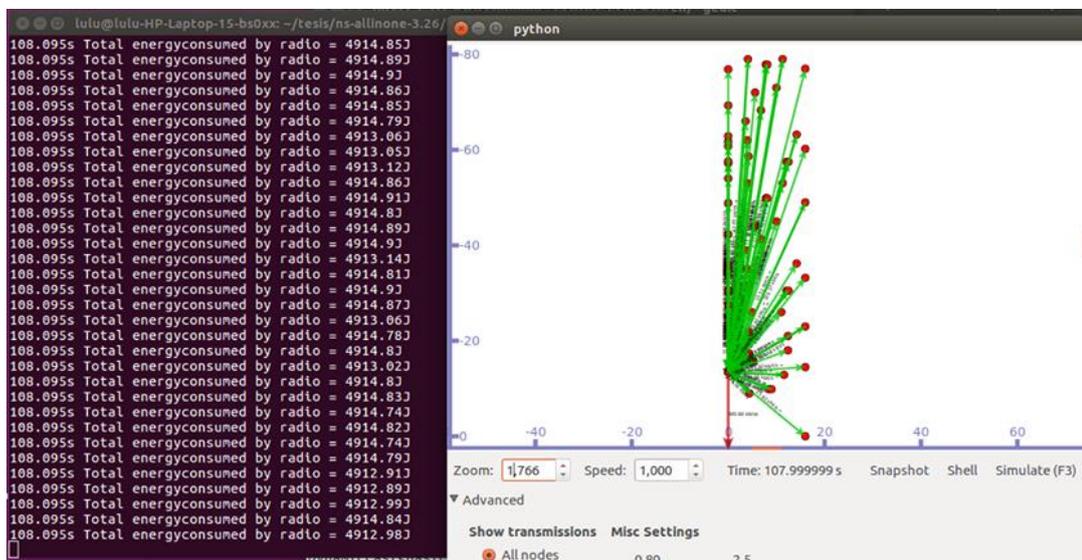


Figura 20. Visualización de la transmisión y recepción de datos en un escenario B
Fuente: Autor
Elaboración: Autor

Con el protocolo TCP:

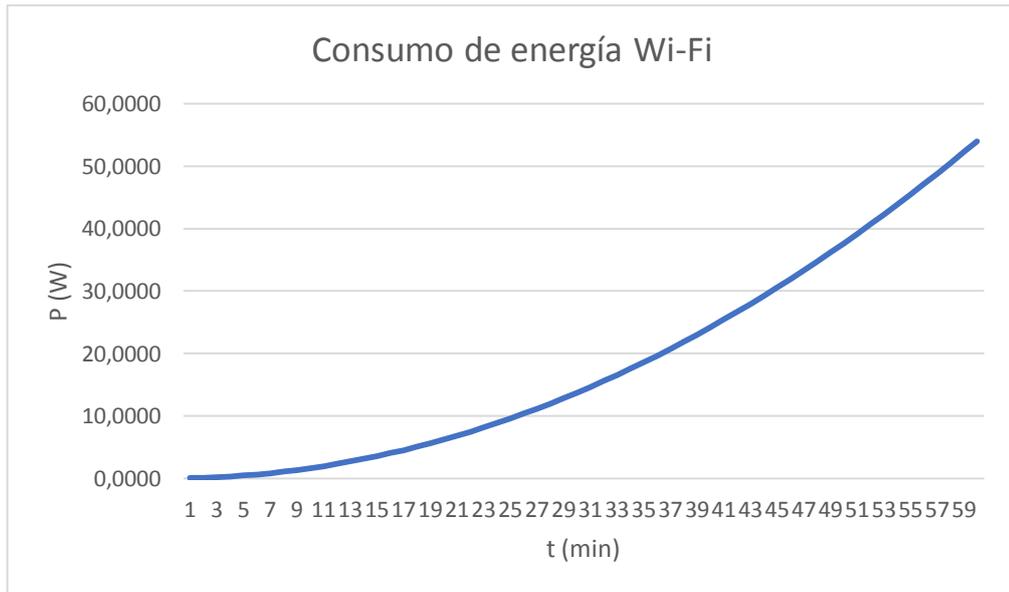


Figura 21. Consumo de energía del escenario B en un período de 1 hora
Fuente: Autor
Elaboración: Autor

En la Figura 21 se puede observar los resultados obtenidos del escenario B, se obtuvo un consumo máximo aproximado 52W-h, es importante recalcar que este resultado se obtuvo con la tecnología Wi-Fi.

Comparación de protocolos

La Figura 22 presenta los resultados obtenidos de varias simulaciones realizados con los diferentes protocolos que ahí se mencionan y la tecnología Wi-Fi.

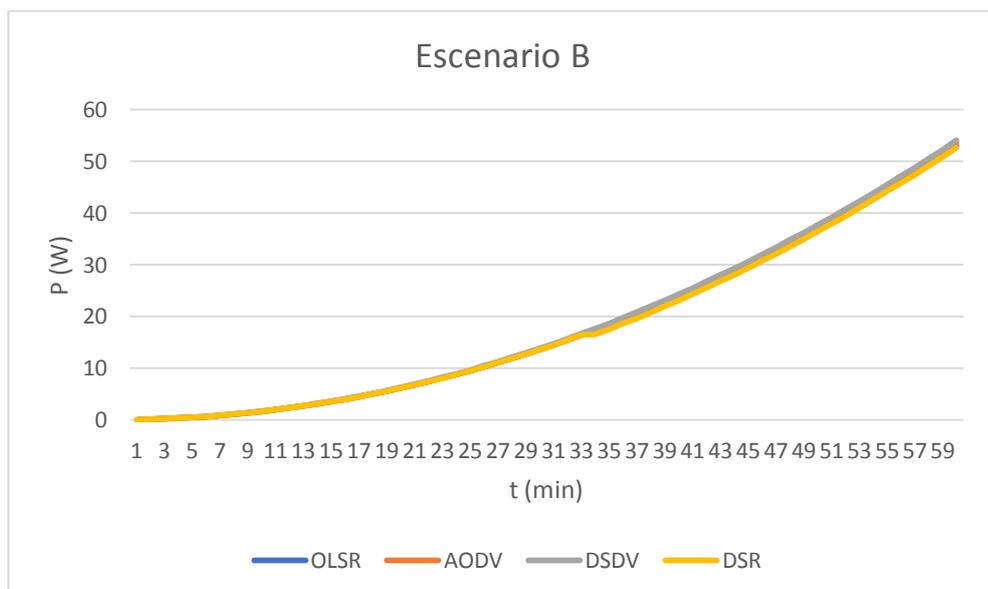


Figura 22. Consumo de energía durante la transmisión y recepción por un tiempo de 60s.
Fuente: Autor
Elaboración: Autor

Estandar ZigBee

Para el desarrollo de esta simulación se empleó un módulo ya diseñado en (Rege, 2015), LR-WPAN que son basadas en el estándar IEEE 802.15.4 que son para transmisiones de datos de baja velocidad y consumo de energía. Se empleó este módulo con el fin de analizar el consumo de energía en los dos escenarios propuestos. Y como podemos observar su consumo de energía en la Figura 23 de los escenarios A y B. Es importante mencionar que ns-3 aún no tiene desarrollado por completo el módulo de ZigBee por lo que no es compatible con los protocolos que contiene.

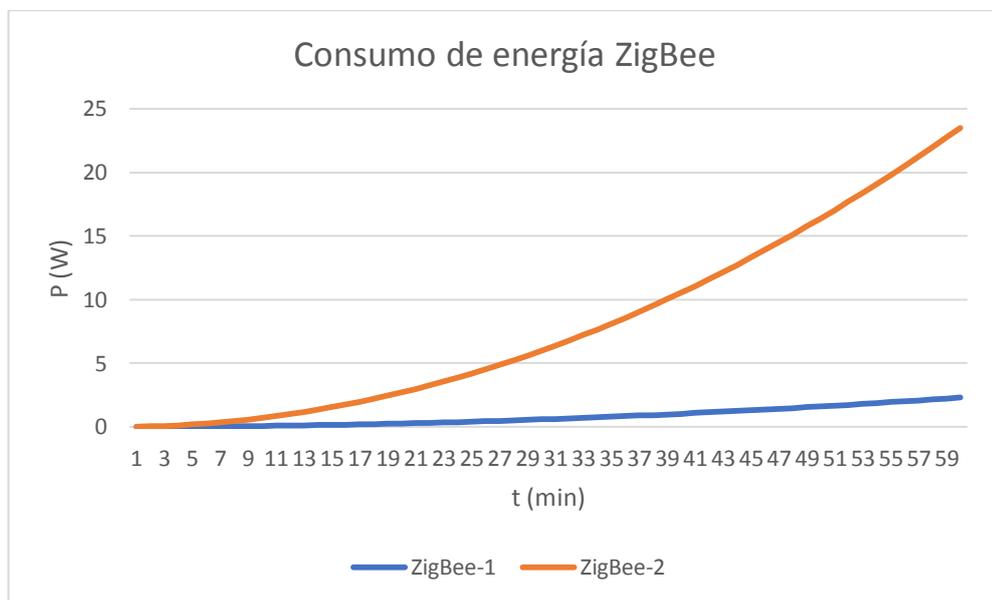


Figura 23. Resultados de consumo energético del escenario A y B con ZigBee
Fuente: Autor
Elaboración: Autor

Comparación de Wi-Fi y ZigBee

En la Figura 24 se puede observar el consumo de energía en un periodo de 60 min de Wi-Fi y ZigBee, es importante recalcar que la diferencia de consumo de energía es mayor en Wi-Fi por consiguiente con ZigBee la red minimiza su consumo.

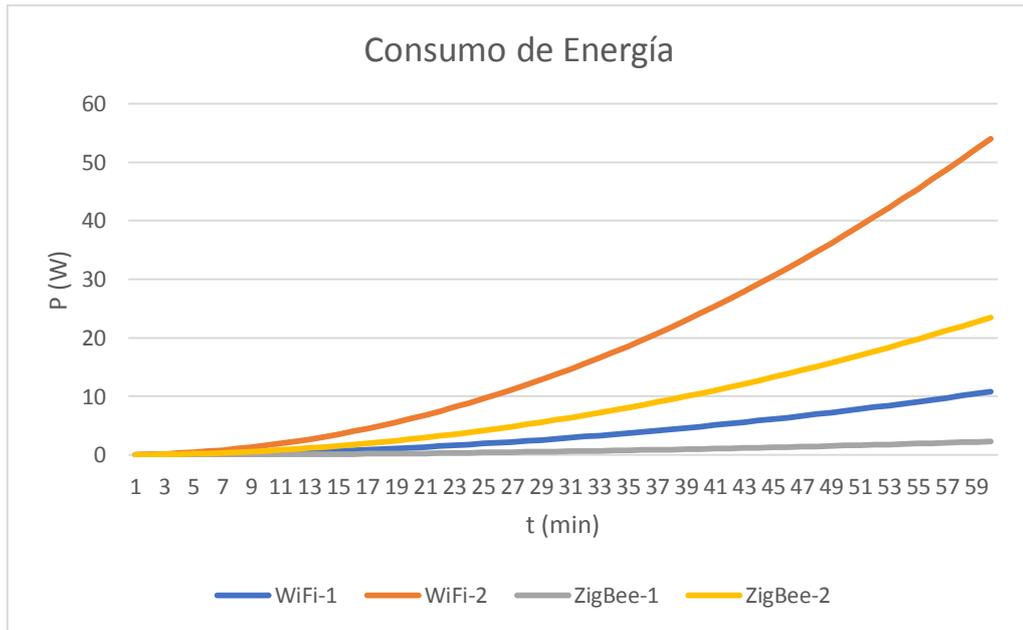


Figura 24. Consumo energético de las tecnologías Wi-Fi y Zigbee en cada escenario.
Fuente: Autor
Elaboración: Autor

El consumo de energía obtenido por las simulaciones realizadas en el entorno de ns-3 gracias a la introducción del modelo de energía, se pudo constatar que minimiza el consumo al permanecer los nodos fijos y si se emplea ZigBee con el escenario B se obtiene un consumo máximo de 11W-h, en cambio con Wi-Fi 53W-h aproximadamente. La energía consumida por cada nodo es proporcional a los paquetes transmitidos, también influye la potencia de transmisión y recepción e incluso la distancia entre cada nodo.

4.3.6 PROPUESTA

El consumo de energía es un problema muy común en WSN, por ende, las investigaciones se centran en este tema con el fin de obtener soluciones tecnológicas e innovadoras que minimicen su consumo y maximicen el tiempo de vida útil de la red inalámbrica. El objetivo principal de este documento es recomendar tecnologías y algunos parámetros que reduzcan el consumo de energía de la red WSN.

Según (Han & Lim, 2010) existen en la actualidad organizaciones que emplean IEEE 802.15.4 y ZigBee con el fin de brindar soluciones efectivas y viables en varias áreas, incluyendo el control de dispositivos electrónicos de consumo, la gestión de energía y la administración de plantas industriales. En este documento sugieren un sistema de administración de energía para el hogar basado en estos protocolos, este sistema propuesto divide y asigna diferentes tareas a la red doméstica.

Los resultados obtenidos del diseño de la red son los que permiten elegir la tecnología junto con las diferentes características que influyen en su consumo en cualquier escenario, además de algunas características de los sensores de corriente y voltaje empleados para el diseño e implementación del enchufe inteligente (Peréz, 2019). En cuanto a la tecnología se recomienda la que minimice el consumo de energía en la red, los resultados que se obtuvieron con la tecnología ZigBee permitieron concluir que es de menor consumo y además es considerada una tecnología de bajo costo, a pesar de que no es muy habitual en los hogares. En cambio, WiFi es parte de la mayoría de los hogares y dispositivos inalámbricos inteligentes pero su consumo es mayor comparado con la tecnología anterior, a pesar de ello se la puede utilizar y buscar soluciones para reducir su consumo.

Gracias a los resultados que se obtuvieron se puede predecir el consumo de la red durante un día e incluso al mes, lo que se observa en la Tabla 6 son los valores de consumo diario y mensual para Wi-Fi y ZigBee, tomando en cuenta algunas características que permitieron analizar a una red inalámbrica desde el consumo de energía que influye en su tiempo de vida. En (Peréz, 2019) se diseñó e implemento un prototipo de medidor inteligente doméstico la misma que puede ser administrada por el usuario a través de una plataforma web, este dispositivo mide diferentes potencias y el consumo eléctrico de un electrodoméstico o varios a la vez y las sube a la plataforma web a establecer conexión a una red inalámbrica, para ello emplearon dispositivos de bajo costo. De este documento se obtuvieron unos parámetros para la simulación.

Tabla 6. Consumo de energía en los estándares Wifi y ZigBee

Wi-Fi		
Nodos	Consumo diario (W-h)	Consumo Mensual (kW-h)
14	259.5702	7.268
65	1295.9036	36.285
ZigBee		
14	55.2703	1.548
65	564.361	15.802

Fuente: Autor
Elaboración: Autor

En la Tabla 6 se puede observar un consumo mensual máximo de 36.285KWh obtenido en un entorno con 65 nodos desplegados en un edificio de 3 o 4 pisos con una frecuencia de envío de 3 segundos. Además, se vuelve a verificar que ZigBee es de bajo consumo energético, específicamente LR-WPAN por lo tanto es una opción viable y eficiente para la

implementación de futuras redes inalámbricas. Además, es una tecnología de transmisión de datos a baja velocidad, de bajo consumo de energía y bajo costo.

La finalidad de este proyecto es informar y permitir conocer cuanta energía se consume en una red inalámbrica, buscar la eficiencia energética de la red en general; para que en futuras implementaciones de redes se puedan adoptar medidas que mejoren su eficiencia energética. Según (González, 2012) la eficiencia adopta una serie de hábitos responsables que se pueden aprender para tener una calidad de vida sostenible cuyo objetivo es ahorrar energía y dinero.

Existen varios sistemas inteligentes de energía que buscan la eficiencia energética en los hogares uno de ellos es HEMS, que cuenta con varios prototipos que gestionan la energía con la ayuda de fuentes de energía renovables, esta es una opción que se puede adaptar a la red inalámbrica. Según (Garcés Mendieta, 2017; Pérez, 2019; Schwartz et al., 2015) si el consumidor conoce el consumo generado por cada electrodoméstico se autorregula con solo conocer.

Hay varias plataformas de Internet de las Cosas de bajo costo y son muy utilizadas, en la Tabla 7 se describen algunas de ellas con sus principales características:

Tabla 7. Plataformas hardware IoT

Plataformas de IoT	Microcontrolador	Tecnologías	Voltaje
NodeMCU ESP866	ESP8266	Wifi	3.3 y 5V
Raspberry pi	Chipset Broadcom BCM2387/ 1,2 GHz de cuatro núcleos ARM Cortex-A53	Wifi/ Bluetooth/ Zigbee	5V
Arduino	Atmel 328	Wifi/ Ethernet	5V
Waspote	ATMega 1281	Zigbee/ Bluetooth/ GPRS	3.3-4.2V

Fuente: (Garrido, 2015)
Elaboración: Autor

CONCLUSIONES

- Con la ayuda del Estado del Arte sobre HEMS, tecnologías, protocolos y más características se tuvo una idea del funcionamiento de este sistema, la manera como gestionan la energía en el hogar y conocer si está relacionado con el diseño e implementación de redes inalámbricas. A su vez adoptar y tomar en cuenta medidas para regular el consumo en el hogar.
- Se diseñó la arquitectura de una red WSN principalmente por la escalabilidad, con la finalidad de a futuro añadir uno o más nodos al sistema de manera fácil y sencilla, este tipo de red puede gestionar un gran número de dispositivos.
- Mediante la simulación de diversos escenarios del proyecto con el fin de recrear situaciones reales, se utilizaron dos tecnologías Wifi y ZigBee. En ns-3 se obtuvo el consumo de energía, que dependen de la cantidad de paquetes de datos, a las características de las plataformas y sensores orientados a WSN. Al compararlas en los dos escenarios propuestos se constató que Zigbee es la tecnología de menor consumo energético. Aunque es importante recalcar que la plataforma ns-3 ún está desarrollando esta tecnología.
- Los protocolos de enrutamiento que son parte de la plataforma de simulación ns-3 como OLSR, AODV, DSDV y DSR son orientados a redes adhoc, se los empleó para analizar el consumo de energía junto con la tecnología Wi-Fi, se concluye que el protocolo de enrutamiento de menor consumo es DSR.
- El aumento del consumo de energía en redes inalámbricas WSN influyen en su rendimiento, vida útil, costo de implementación y el desgaste prematuro de las baterías. Las tecnologías también cumplen una función importante en el desarrollo de esta red para ello es importante concluir que Wi-Fi es una estándar de mayor consumo, pero si se desea gestionar y monitorear esta red desde cualquier lugar, esta tecnología es la más adecuada. ZigBee puede ofrecer un mejor servicio en cuanto a costo y consumo. Bluetooth se la puede emplear por su economía, pero su desventaja es que solo se la puede monitorear por medio del Gateway dentro del hogar. El consumo de energía obtenido en los dos escenarios propuestos no aumenta el consumo en su hogar al contrario lo mantiene.

RECOMENDACIONES

- Para futuros trabajos investigativos relacionados con el consumo de energía se recomienda aprender a utilizar la plataforma de red ns-3, añadir módulos de la manera correcta con el fin de emplear protocolos orientados a redes de sensores inalámbricos como MQTT, CoAp, entre otros estándares abiertos. Para tener una mejor perspectiva del consumo de energía de una red inalámbrica de sensores.
- El incremento de electrodomésticos en el hogar es común en la mayoría de hogares, debido a esto, se recomienda tener en cuenta ciertas medidas de eficiencia energética, nuevas técnicas, estrategias, tecnologías, protocolos que mejoren la infraestructura de la red WSN para el hogar.
- Se recomienda realizar la misma simulación en otros simuladores de red con el fin de comprobar el consumo obtenido en los escenarios propuestos o en cualquiera que se requiera.

REFERENCIAS

- Ahmed, M. S., Al Hassan, E., Mohamed, A., & Shareef, H. (2018). Review on Home Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers. *IEEE Access*, 6, 24498–24509. <https://doi.org/10.1109/ACCESS.2018.2831917>
- Akbar, S. R., Amron, K., Mulya, H., & Hanifah, S. (2017). Message queue telemetry transport protocols implementation for wireless sensor networks communication — A performance review. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 107–112). Malang, Indonesia: IEEE. <https://doi.org/10.1109/SIET.2017.8304118>
- Alipio, M., Tiglao, N. M., Grilo, A., Bokhari, F., Chaudhry, U., & Qureshi, S. (2017). Cache-based transport protocols in wireless sensor networks: A survey and future directions. *Journal of Network and Computer Applications*, 88(October 2016), 29–49. <https://doi.org/10.1016/j.jnca.2017.04.001>
- Anonymous. (2010). Energy Framework. Retrieved December 31, 2018, from <https://www.nsnam.org/docs/release/3.10/manual/html/energy.html>
- Arias, A., Alfredo, M., Vilchis, M., Aurora, M., Ortigoza, S., Melchor, M., ... Wireless, L. A. S. (2008). *Revisión del proceso de identificación de nodos en las Wireless Sensor Networks*. Universidad Rafael Bellosó Chacín. Retrieved from <http://www.redalyc.org/pdf/784/78411785001.pdf>
- Batista, N. C., Melício, R., Matias, J. C. O., & Catalão, J. P. S. (2013). Photovoltaic and wind energy systems monitoring and building/home energy management using ZigBee devices within a smart grid. *Energy*, 49(1), 306–315. <https://doi.org/10.1016/j.energy.2012.11.002>
- Bosmediano, S., & Ochoa, P. (2017). *Diseño e implementación de una plataforma hardware para aplicaciones de redes inalámbricas de sensores. (Trabajo de Titulación de Ingeniero en Electrónica y Telecomunicaciones)*. UTPL, Loja. Retrieved from <http://dspace.utpl.edu.ec/handle/123456789/17016>
- Callaway, E., Gorday, P., Hester, L., Gutierrez, J. A., Naeve, M., Heile, B., & Bahl, V. (2002). Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine*, 40(8), 70–77.

<https://doi.org/10.1109/MCOM.2002.1024418>

Castro, J., & Losilla, F. (2014). *Uso del protocolo CoAP para la implementación de una aplicación domótica con redes de sensores inalámbricas*. Universidad Politécnica de Cartagena. Retrieved from <http://repositorio.upct.es/bitstream/handle/10317/4163/pfc5908.pdf;sequence=1>

Chee-Yee Chong, & Kumar, S. P. (2003). Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8), 1247–1256. <https://doi.org/10.1109/JPROC.2003.814918>

Collins. (2012). Energy consumption definition and meaning | Collins English Dictionary. Retrieved December 31, 2018, from <https://www.collinsdictionary.com/dictionary/english/energy-consumption>

Collotta, M., & Pau, G. (2015). A Novel Energy Management Approach for Smart Homes Using Bluetooth Low Energy. *IEEE Journal on Selected Areas in Communications*, 33(12), 2988–2996. <https://doi.org/10.1109/JSAC.2015.2481203>

Edwards, C. (2012). Smart dust. *Engineering & Technology*, 7(6), 74. <https://doi.org/10.1049/et.2012.0612>

Fateh, B., & Govindarasu, M. (2015). Joint Scheduling of Tasks and Messages for Energy Minimization in Interference-Aware Real-Time Sensor Networks. *IEEE Transactions on Mobile Computing*, 14(1), 86–98. <https://doi.org/10.1109/TMC.2013.81>

Feeney, L. M., & Nilsson, M. (2001). Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)* (Vol. 3, pp. 1548–1557). Anchorage, AK, USA: IEEE. <https://doi.org/10.1109/INFCOM.2001.916651>

Garcés Mendieta, L. T. (2017). *Diseño e implementación de una calculadora energética doméstica*. UTPL, Loja. Retrieved from <http://dspace.utpl.edu.ec/handle/123456789/19937>

García, L. (2015). *Energy Consumption in Wireless Mesh and Wireless Ad-hoc Networks*. Universidad Politecnica de Catalunya, Barcelona. Retrieved from https://upcommons.upc.edu/bitstream/handle/2117/79562/Energy_Consumption_in_Wireless_Mesh_Networks_Degree

Thesis.pdf?sequence=1&isAllowed=y&fbclid=IwAR35Jok4_a0td_yrf0xyG5C35pnEPPdRN4op2NOnfijRfhCzG1rgeXXFNAU

- Garrido, R. (2015). *Estudio Plataformas IoT. PFC* (Vol. 28). Retrieved from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/42812/6/rloureiroTFC0615memoria.pdf>
- González, R. (2012). Qué es la eficiencia energética. Retrieved February 13, 2019, from <https://twenergy.com/a/que-es-la-eficiencia-energetica-39>
- Guang, N. L. L., Logenthiran, T., & Abidi, K. (2017). Application of Internet of Things (IoT) for home energy management. In *2017 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)* (pp. 1–6). IEEE. <https://doi.org/10.1109/APPEEC.2017.8308962>
- Hacks, C. (2016). Waspote - Wireless Sensor Networks Open Source Platform. Retrieved March 9, 2019, from <https://www.cooking-hacks.com/documentation/tutorials/waspote/>
- Hammi, B., Khatoun, R., Zeadally, S., Fayad, A., & Khoukhi, L. (2018). IoT technologies for smart cities. *IET Networks*, *3203*(1), 1–13. <https://doi.org/10.1049/iet-net.2017.0163>
- Han, D.-M., & Lim, J.-H. (2010). Design and implementation of smart home energy management systems based on zigbee. *Consumer Electronics, IEEE Transactions On*, *56*(3), 1417–1425. <https://doi.org/10.1109/TCE.2010.5606278>
- Ingalaturre, G. D., Lopéz, J., & Villajosana, X. (2018). *Integración de un modelo de energía en la simulación de redes LoRaWAN en NS-3*. Universitat Oberta de Catalunya. Retrieved from http://openaccess.uoc.edu/webapps/o2/bitstream/10609/80727/4/gdobatoTFG0618memoria.pdf?fbclid=IwAR3_r33fS-M4cDqtUBminb5ZoPMO8tuYWtKRAkAd8qk6fFvPu3hDYZUrKg0#%5B%7B%22num%22%3A312%2C%22gen%22%3A0%7D%2C%7B%22name%22%3A%22XYZ%22%7D%2C82%2C715%2C0%5D
- Inoue, M., Higuma, T., Ito, Y., Kushiro, N., & Kubota, H. (2003). Network architecture for home energy management system. *IEEE Transactions on Consumer Electronics*, *49*(3), 606–613. <https://doi.org/10.1109/TCE.2003.1233782>
- Khajuria, R., & Gupta, S. (2015). Energy optimization and lifetime enhancement techniques in wireless sensor networks: A survey. In *International Conference on Computing*,

Communication & Automation (pp. 396–402). Jammu, India: IEEE.
<https://doi.org/10.1109/CCAA.2015.7148408>

Khalid, A., Javaid, N., Guizani, M., Alhussein, M., Aurangzeb, K., & Ilahi, M. (2018). Towards Dynamic Coordination among Home Appliances Using Multi-Objective Energy Optimization for Demand Side Management in Smart Buildings. *IEEE Access*, 6, 19509–19529. <https://doi.org/10.1109/ACCESS.2018.2791546>

Kodali, R. K., & Aravapalli, kumar N. (2014). Multi level secure LEACH protocol model using NS-3. In *2014 First International Conference on Networks & Soft Computing (ICNSC2014)* (pp. 198–202). IEEE. <https://doi.org/10.1109/CNSC.2014.6906715>

Lee, J. S., & Kao, T. Y. (2016). An Improved Three-Layer Low-Energy Adaptive Clustering Hierarchy for Wireless Sensor Networks. *IEEE Internet of Things Journal*, 3(6), 951–958. <https://doi.org/10.1109/JIOT.2016.2530682>

Ling, Z., Luo, J., Xu, Y., Gao, C., Wu, K., & Fu, X. (2017). Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System. *IEEE Internet of Things Journal*, 4(6), 1899–1909. <https://doi.org/10.1109/JIOT.2017.2707465>

Mao, G., Fidan, B., & Anderson, B. D. O. (2007). Wireless sensor network localization techniques. *Computer Networks*, 51(10), 2529–2553. <https://doi.org/10.1016/j.comnet.2006.11.018>

Mattei, U. (2014). *Extended physical layer modeling for smart metering utility network simulators*. KTH VETENSKAP OCH KONST. Retrieved from <http://www.diva-portal.se/smash/get/diva2:750092/FULLTEXT01.pdf?fbclid=IwAR3iVEu-MBGUKNw3clgUnNiKzIjIxuoUCKoUcnCP4XappFkDO5I2OSe2XUE>

McDermott-Wells, P. (2005). What is Bluetooth? *IEEE Potentials*, 23(5), 33–35. <https://doi.org/10.1109/MP.2005.1368913>

Meena, S. S., & Manikandan, J. (2017). Study and evaluation of different topologies in wireless sensor network. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 107–111). Chennai, India: IEEE. <https://doi.org/10.1109/WiSPNET.2017.8299729>

Moridi, M. A., Kawamura, Y., Sharifzadeh, M., Chanda, E. K., Wagner, M., & Okawa, H. (2018). Performance analysis of ZigBee network topologies for underground space monitoring and communication systems. *Tunnelling and Underground Space*

Technology, 71(August 2017), 201–209. <https://doi.org/10.1016/j.tust.2017.08.018>

Mundaca, S. (2014). ▷ Plano eléctrico casa dos pisos en AutoCAD | CAD (82.6 KB) | Bibliocad. Retrieved January 31, 2019, from https://www.bibliocad.com/es/biblioteca/plano-electrico-casa-dos-pisos_67457/

Nayak, P., & Devulapalli, A. (2016). A Fuzzy Logic-Based Clustering Algorithm for WSN to Extend the Network Lifetime. *IEEE Sensors Journal*, 16(1), 137–144. <https://doi.org/10.1109/JSEN.2015.2472970>

ns-3. (2010). Tutorial ns-3: 4.2 Un primer script ns-3. Retrieved December 28, 2018, from https://www.nsnam.org/docs/release/3.9/tutorial/tutorial_18.html?fbclid=IwAR3tUyVwyKuFIXRDNU95JUBYbQTuDjONFNSKVLy4eATQSR8tcADrqPQGpdA#A-First-ns_002d3-Script

Patel, R., & Kamboj, P. (2015). Investigation of Network Simulation Tools and Comparison Study : NS3 vs NS2. *Journal of Network Communications and Emerging Technologies*, 5(2), 137–142. Retrieved from www.jncet.org

Peréz, V. (2019). *Diseño y construcción de un hub de medición de consumo de energía eléctrica para uso doméstico. (Trabajo de Titulación de Ingeniero en Electrónica y Telecomunicaciones)*. UTPL, Loja.

Rege, V. (2015). vrege/ns-3-gsoc: files. Retrieved February 11, 2019, from <http://code.nsnam.org/vrege/ns-3-gsoc/file/c43c335bc921/?fbclid=IwAR3gNdegKTi8VyG6ECOMDH914JI6fvj-LiiJsKCsWkKwdUakcuotwrcXYUA>

Rivera, D., & Gil, M. E. (2013). Instalaciones eléctricas edificio 3 plantas - planos y bloques en AutoCAD sobre arquitectura y construcción. Retrieved January 31, 2019, from <http://documentos.arq.com.mx/Detalles/109793.html>

Rodriguez, G., & Padilla, A. (2015). *Análisis y Simulación de protocolos de enrutamiento adecuados en diferentes escenarios para redes AdHoc, mediante la herramienta Ns-3. Tesis*. Universidad Nacional de Loja. Retrieved from http://dspace.unl.edu.ec/jspui/bitstream/123456789/11585/1/Rodríguez_Pineda%2C_Gabriela_Maribel.pdf

Salazar, J., & Silvestre, S. (2000). *Internet de las cosas. České vysoké učení technické v Praze Fakulta elektrotechnická*.

- Schwartz, T., Stevens, G., Jakobi, T., Deneff, S., Ramirez, L., Wulf, V., & Randall, D. (2015). What People Do with Consumption Feedback: A Long-Term Living Lab Study of a Home Energy Management System. *Interacting with Computers*, 27(6), 551–576. <https://doi.org/10.1093/iwc/iwu009>
- Singh, P., Barkhodia, E., & Walia, G. K. (2012). Performance Study of Different Routing Protocols (OLSR , DSR , AODV) Under Different Traffic Loads and with Same Number of Nodes in MANET using OPNET. *Ijunct*, 7109, 155–157.
- Sohraby, K., Minoli, D., & Znati, T. (2007). *Wireless Sensor Networks*. *Booksgooglecom*. <https://doi.org/10.1002/047011276X>
- Soua, R., & Minet, P. (2011). A survey on energy efficient techniques in wireless sensor networks. In *2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)* (Vol. 14, pp. 1–9). Toulouse, France: IEEE. <https://doi.org/10.1109/WMNC.2011.6097244>
- Steven, B. (2012). *Arduino Microcontroller: Processing for Everyone! Second Edition* (1st ed.). Morgan & Claypool. <https://doi.org/10.2200 / S00421ED1V01Y201205DCS038>
- Tazaki, H., Requena, M., Pecoralla, T., Miozzo, M., Krotov, A., Katsaros, K., ... Ali, Z. (2011). Acerca de | ns-3. Retrieved October 22, 2018, from <https://www.nsnam.org/about/>
- Toriumi, K., Kakizaki, Y., & Iwamura, K. (2011). Fast Implementation of the Advanced Encryption Standard Using Atmega1281. In *2011 Sixth International Conference on Availability, Reliability and Security* (pp. 342–346). Vienna, Austria: IEEE. <https://doi.org/10.1109/ARES.2011.57>
- Vasanthi, M. S., Kumar, D., & Rama Rao, T. (2013). Analysis of AQ-DBPSK modulation for WSN transceivers in indoor environments. In *2013 International Conference on Information Communication and Embedded Systems (ICICES)* (pp. 904–907). IEEE. <https://doi.org/10.1109/ICICES.2013.6508198>
- Vinodhini, S., & Priya, B. (2017). Optimum modulation technique for modulating RSSI in WSN. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 1670–1673). Chennai, India, India: IEEE. <https://doi.org/10.1109/ICECDS.2017.8389732>
- Wei Ye, Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE*

Computer and Communications Societies (Vol. 3, pp. 1567–1576). New York, NY, USA, USA: IEEE. <https://doi.org/10.1109/INFCOM.2002.1019408>

Wu, H., Nabar, S., & Poovendran, R. (2011). An Energy Framework for the Network Simulator 3 (ns-3). In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques* (Vol. 3, p. 9). ACM. <https://doi.org/10.4108/icst.simutools.2011.245534>

Xiaodong Xian, Weiren Shi, & He Huang. (2008). Comparison of OMNET++ and other simulator for WSN simulation. In *2008 3rd IEEE Conference on Industrial Electronics and Applications* (pp. 1439–1443). Singapore: IEEE. <https://doi.org/10.1109/ICIEA.2008.4582757>

Yang, X., Wang, L., Su, J., & Gong, Y. (2018). Hybrid MAC Protocol Design for Mobile Wireless Sensors Networks. *IEEE Sensors Letters*, 2(2), 1–4. <https://doi.org/10.1109/LSENS.2018.2828339>

Zheng, J., & Jamalipour, A. (2009). *Wireless Sensor Networks*. (J. Zheng & A. Jamalipour, Eds.), *Wireless Sensor Networks: A Networking Perspective*. Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470443521>

Zheng, J., & Lee, M. (2002). A Comprehensive Performance Study of IEEE 802.15.4. In *IEEE* (pp. 1202–1206). IEEE. Retrieved from <https://pdfs.semanticscholar.org/622f/702aec96d4fe2ffe426f901f542aa92657d0.pdf>

Zhou, B., Li, W., Chan, K. W., Cao, Y., Kuang, Y., Liu, X., & Wang, X. (2016). Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61, 30–40. <https://doi.org/10.1016/j.rser.2016.03.047>

ANEXOS

ANEXO A

Escenario A Simulación Wi-Fi

```
//Modulos de Inclusión
#include "ns3/core-module.h"
#include "ns3/command-line.h"
#include "ns3/network-module.h"
#include "ns3/wifi-module.h"
#include "ns3/applications-module.h"
#include "ns3/config-store-module.h"
#include "ns3/energy-module.h"
#include "ns3/internet-module.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/mobility-helper.h"
#include "ns3/ipv4-address-helper.h"
#include "ns3/ipv4-static-routing-helper.h"
#include "ns3/ipv4-nix-vector-helper.h"

#include "ns3/string.h"
#include <algorithm>
#include <ctime>
#include <iostream>
#include <sstream>
#include <fstream>
#include <vector>
#define TOTAL_ENERGY
#define NODE_ENERGY

#ifdef TOTAL_ENERGY
    double total_energy = 0.0;
#endif

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("Tesis");
void ReceivePacket (Ptr<Socket> socket)
{
    while (socket->Recv ())
    {
        NS_LOG_UNCOND ("Paquete recibido!");
    }
}

#ifdef NODE_ENERGY
template <int node>
void RemainingEnergy (double oldValue, double newValue)
```

```

{
std::stringstream ss;
ss << "uno_" << node << ".log";

static std::fstream f (ss.str().c_str(), std::ios::out);

f << Simulator::Now().GetSeconds() << " remaining energy=" << newValue <<
std::endl;
}

#endif

#ifdef TOTAL_ENERGY
void TotalEnergy (double oldValue, double totalEnergy)
{

NS_LOG_UNCOND ( Simulator::Now().GetSeconds() <<"s Total energyconsumed by
radio = " << totalEnergy << "J");

}
#endif

//Definición de variables
//dB
#define rss -80
#define packetSize 3600
#define numPackets 200
#define n 14 //numero de nodos
#define duration 3600 //tiempo de simulacion
#define initialEnergy 5000 //Joule
#define voltage 10.0 //volts
#define txPowerStart 0.0 //dBm
#define txPowerEnd 18 //dBm
#define ntxPowerLevels 16
#define idleCurrent 0.001 // Ampere
#define txCurrent 0.380 // Ampere

int main (int argc, char *argv[])
{
std::string dataRate = "2Mbps";
std::string phyMode ("DsssRate2Mbps");//Tasa de datos de 2Mbps
bool verbose = true;

CommandLine cmd;
cmd.AddValue ("dataRate", "Data rate", dataRate);
cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);
cmd.AddValue ("verbose", "turn on all WifiNetDevice log components", verbose);
cmd.Parse (argc, argv);

Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode",

```

```

        StringValue (phyMode));
if (verbose)
{
    LogComponentEnable("UdpEchoClientApplication",LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
}

NodeContainer nodos;
nodos.Create (n);

// Configuración del estandar IEEE 802.11b
SeedManager::SetRun(23);
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel;
WifiHelper wifi;
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
//wifiPhy.Set("EnergyDetectionThreshold", DoubleValue (-86.0));
//wifiPhy.Set("CcaModelThreshold", DoubleValue (-99.0));
wifiPhy.Set ("TxPowerStart", DoubleValue (txPowerStart));
wifiPhy.Set ("TxPowerEnd", DoubleValue (txPowerEnd));
wifiPhy.Set("TxPowerLevels", UIntegerValue(ntxPowerLevels));
wifiPhy.Set("TxGain", DoubleValue(1.0));
wifiPhy.Set("RxGain", DoubleValue(1.0));
wifiPhy.Set("RxNoiseFigure", DoubleValue (7.0) ); // defulat val is 7dB
wifiPhy.Set("ChannelNumber", UIntegerValue(3));
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::FixedRssLossModel","Rss",DoubleValue (rss));
wifiPhy.SetChannel (wifiChannel.Create ());

WifiMacHelper wifiMac;
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
    "DataMode",StringValue (phyMode),
    "ControlMode",StringValue (phyMode));

// Modo Adhoc
wifiMac.SetType ("ns3::AdhocWifiMac");
NetDeviceContainer devices = wifi.Install (wifiPhy, wifiMac, nodos);

//Posicionamiento de los nodos
MobilityHelper mobility;
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
//Posiciones del primer piso
//Gateway
positionAlloc->Add (Vector (1.86, 6.64 ,0.0 ));
//Nodos
positionAlloc->Add (Vector (-9.86, 15.0, 0.0));
positionAlloc->Add (Vector (1.86, 15.0, 0.0));
positionAlloc->Add (Vector (1.86, 0.0, 0.0));
positionAlloc->Add (Vector (-3.86, 2.0, 0.0));
positionAlloc->Add (Vector (-4.86, 3.0, 0.0));
positionAlloc->Add (Vector (-7.86, 1.02, 0.0));

```

```

positionAlloc->Add (Vector (-7.86, 1.86, 0.0));
//Posiciones del segundo piso
positionAlloc->Add (Vector (-7.86, -4.02, 0.0));
positionAlloc->Add (Vector (-2.06, -4.02, 0.0));
positionAlloc->Add (Vector (-9.86, -10.64, 0.0));
positionAlloc->Add (Vector (1.86, -10.64, 0.0));
positionAlloc->Add (Vector (-4.86, -14.64, 0.0));
positionAlloc->Add (Vector (-2.06, -19.64, 0.0));
mobility.SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodos);

/** Internet stack */
InternetStackHelper internet;
internet.SetIpv6StackInstall(false);
internet.Install (nodos);

Ipv4AddressHelper addr;
NS_LOG_INFO ("Assign IP Addresses.");
addr.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interfaces = addr.Assign (devices);

ApplicationContainer apps;
for (uint32_t i = 0; i < n; i++) {
std::string transportProto = std::string("ns3::UdpSocketFactory");
OnOffHelper onOff(transportProto, InetSocketAddress (Ipv4Address ("10.1.1.1"), 9000));

onOff.SetAttribute ("DataRate", DataRateValue (DataRate (dataRate)));
onOff.SetAttribute ("PacketSize", UintegerValue (packetSize));
onOff.SetAttribute ("OffTime", StringValue
("ns3::ConstantRandomVariable[Constant=0.001]"));

apps = onOff.Install (nodos.Get (i));

apps.Start (Seconds (0.01));
apps.Stop (Seconds (duration));

PacketSinkHelper sink (transportProto, InetSocketAddress (Ipv4Address::GetAny (),
9001));
apps = sink.Install (nodos.Get (i));
apps.Start (Seconds (0.01));
}
apps.Stop (Seconds (duration));

//Modulo de energía
EnergySourceContainer eSources;
BasicEnergySourceHelper basicSourceHelper;
WifiRadioEnergyModelHelper radioEnergyHelper;

basicSourceHelper.Set ("BasicEnergySourceInitialEnergyJ", DoubleValue

```

```

(initialEnergy));
basicSourceHelper.Set ("BasicEnergySupplyVoltageV", DoubleValue (voltage));

radioEnergyHelper.Set ("TxCurrentA", DoubleValue (txCurrent));
radioEnergyHelper.Set ("RxCurrentA", DoubleValue (0.170));
radioEnergyHelper.Set ("IdleCurrentA", DoubleValue (idleCurrent));
radioEnergyHelper.Set ("CcaBusyCurrentA", DoubleValue (0));
radioEnergyHelper.Set ("SwitchingCurrentA", DoubleValue (0.001));
radioEnergyHelper.Set ("SleepCurrentA", DoubleValue (0.001));

double eta = WifiTxCurrentModel::DbmToW (txPowerStart) / ((txCurrent - idleCurrent) *
voltage);

radioEnergyHelper.SetTxCurrentModel ("ns3::LinearWifiTxCurrentModel",
                                     "Voltage", DoubleValue (voltage),
                                     "IdleCurrent", DoubleValue (idleCurrent),
                                     "Eta", DoubleValue (eta));

// Instala la fuente de energía en cada nodo
for (NodeContainer::Iterator c = nodos.Begin(); c != nodos.End(); c++)
{
    eSources.Add (basicSourceHelper.Install (*c));

    Ptr<WifiNetDevice> wnd;

    for (uint32_t i = 0; i < (*c)->GetNDevices (); ++i)
    {
        wnd = (*c)->GetDevice (i)->GetObject<WifiNetDevice> ();
        // Este es el dispositivo Wifi
        if (wnd != 0)
        {
            radioEnergyHelper.Install (wnd, eSources.Get (eSources.GetN()-1));
        }
    }
}

// Se guarda los datos obtenidos en un documento .txt
eSources.Get (0)->TraceConnectWithoutContext ("RemainingEnergy",
MakeCallback(&RemainingEnergy<0>));
eSources.Get (1)->TraceConnectWithoutContext ("RemainingEnergy",
MakeCallback(&RemainingEnergy<1>));
eSources.Get (13)->TraceConnectWithoutContext ("RemainingEnergy",
MakeCallback(&RemainingEnergy<13>));

// Indica el numero de paquetes al inicio de la simulación
NS_LOG_UNCOND ("Testing " << numPackets << " packets sent with receiver rss "
<< rss );
/*

```

```

#ifdef NODE_ENERGY
Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>
(eSources.Get (1));
*basicSourcePtr->TraceConnectWithoutContext ("RemainingEnergy", MakeCallback
(&RemainingEnergy));
#endif
*/

#ifdef TOTAL_ENERGY
uint32_t i;
for( i=0 ; i<n ; i++){
Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>
(eSources.Get (i));
basicSourcePtr->TraceConnectWithoutContext ("RemainingEnergy",
MakeCallback(&TotalEnergy));
}
#endif

NS_LOG_INFO ("Run Simulation.");
    Simulator::Stop(Hours (1));
    Simulator::Run ();
    Simulator::Destroy ();
#ifdef TOTAL_ENERGY
std::cout << "Total energy consumed in network = " << total_energy << " J" << std::endl;
#endif
    return 0;
}

```

ANEXO B

Escenario B Simulación para el estándar Wi-Fi

Las mismas características de simulación se utilizaron para el Escenario B lo único que cambia es la ubicación de los 65 nodos:

```
//Posicionamiento de los nodos
MobilityHelper mobility;
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
//Gateway
positionAlloc->Add (Vector (0.0, -13.4, 0.0));
//Posiciones del primer piso
positionAlloc->Add (Vector (16.0, 0.0, 0.0));
positionAlloc->Add (Vector (4.4, -9.0, 0.0));
positionAlloc->Add (Vector (8.4, -9.86, 0.0));
positionAlloc->Add (Vector (9.0, -9.86, 0.0));
positionAlloc->Add (Vector (11.6, -12.86, 0.0));

positionAlloc->Add (Vector (16.0, -14.5, 0.0));
positionAlloc->Add (Vector (4.6, -15.5, 0.0));
positionAlloc->Add (Vector (5.2, -15.8, 0.0));
positionAlloc->Add (Vector (4.3, -16.4, 0.0));
positionAlloc->Add (Vector (4.3, -17.0, 0.0));
positionAlloc->Add (Vector (4.3, -17.4, 0.0));
positionAlloc->Add (Vector (12.4, -18.0, 0.0));
positionAlloc->Add (Vector (12.4, -21.0, 0.0));
positionAlloc->Add (Vector (16.0, -23.0, 0.0));
positionAlloc->Add (Vector (4.3, -21.0, 0.0));
positionAlloc->Add (Vector (4.3, -21.8, 0.0));
positionAlloc->Add (Vector (11.0, -26.0, 0.0));
positionAlloc->Add (Vector (4.9, -26.0, 0.0));

//Posiciones del segundo piso
positionAlloc->Add (Vector (0.0, -27.6, 0.0));
positionAlloc->Add (Vector (0.0, -30.2, 0.0));
positionAlloc->Add (Vector (0.0, -30.6, 0.0));
positionAlloc->Add (Vector (4.2, -31.65, 0.0));
positionAlloc->Add (Vector (12.1, -30.5, 0.0));
positionAlloc->Add (Vector (12.5, -30.5, 0.0));
positionAlloc->Add (Vector (16.0, -33.2, 0.0));
positionAlloc->Add (Vector (14.2, -36.2, 0.0));
positionAlloc->Add (Vector (0.0, -33.8, 0.0));
positionAlloc->Add (Vector (0.0, -34.7, 0.0));
positionAlloc->Add (Vector (4.0, -35.0, 0.0));
positionAlloc->Add (Vector (0.0, -35.86, 0.0));
positionAlloc->Add (Vector (3.6, -39.0, 0.0));
positionAlloc->Add (Vector (6.8, -41.3, 0.0));
positionAlloc->Add (Vector (0.0, -42.3, 0.0));
positionAlloc->Add (Vector (5.6, -44.0, 0.0));
```

```

positionAlloc->Add (Vector (10.0, -45.0, 0.0));
positionAlloc->Add (Vector (16.0, -49.0, 0.0));
positionAlloc->Add (Vector (0.0, -48.86, 0.0));
positionAlloc->Add (Vector (7.8, -49.86, 0.0 ));
positionAlloc->Add (Vector (8.1, -49.86, 0.0));
positionAlloc->Add (Vector (4.1, -53.0, 0.0));
positionAlloc->Add (Vector (11.3, -53.0, 0.0));

//Posiciones del tercer piso
positionAlloc->Add (Vector (0.0, -54.0, 0.0));
positionAlloc->Add (Vector (0.0, -57.2, 0.0));
positionAlloc->Add (Vector (0.0, -57.6, 0.0));
positionAlloc->Add (Vector (4.2, -58.65, 0.0));
positionAlloc->Add (Vector (12.1, -57.5, 0.0));
positionAlloc->Add (Vector (12.5, -57.5, 0.0));
positionAlloc->Add (Vector (16.0, -60.2, 0.0));
positionAlloc->Add (Vector (14.2, -63.2, 0.0));
positionAlloc->Add (Vector (0.0, -60.8, 0.0));
positionAlloc->Add (Vector (0.0, -61.7, 0.0));
positionAlloc->Add (Vector (4.0, -62.0, 0.0));
positionAlloc->Add (Vector (0.0, -62.86, 0.0));
positionAlloc->Add (Vector (3.6, -66.0, 0.0));
positionAlloc->Add (Vector (6.8, -68.3, 0.0));
positionAlloc->Add (Vector (0.0, -69.3, 0.0));
positionAlloc->Add (Vector (5.6, -72.0, 0.0));
positionAlloc->Add (Vector (10.0, -73.0, 0.0));
positionAlloc->Add (Vector (16.0, -77.0, 0.0));
positionAlloc->Add (Vector (0.0, -76.86, 0.0));
positionAlloc->Add (Vector (7.8, -77.86, 0.0 ));
positionAlloc->Add (Vector (8.1, -77.86, 0.0));
positionAlloc->Add (Vector (4.1, -79.0, 0.0));
positionAlloc->Add (Vector (11.3, -79.0, 0.0));
mobility.SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodos);

```

Escenarios A y B, visualización durante la transmisión y recepción



ANEXO C

Configuración del modelo de energía de LR-WPAN

```
Ptr<LrWpanRadioEnergyModel> em0 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em1 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em2 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em3 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em4 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em5 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em6 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em7 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em8 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em9 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em10 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em11 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em12 = CreateObject<LrWpanRadioEnergyModel> ();  
Ptr<LrWpanRadioEnergyModel> em13 = CreateObject<LrWpanRadioEnergyModel> ();
```

```
Ptr<BasicEnergySource> es0 = CreateObject<BasicEnergySource> ();  
es0->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es1 = CreateObject<BasicEnergySource> ();  
es1->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es2 = CreateObject<BasicEnergySource> ();  
es2->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es3 = CreateObject<BasicEnergySource> ();  
es3->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es4 = CreateObject<BasicEnergySource> ();  
es4->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es5 = CreateObject<BasicEnergySource> ();  
es5->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es6 = CreateObject<BasicEnergySource> ();  
es6->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es7 = CreateObject<BasicEnergySource> ();  
es7->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es8 = CreateObject<BasicEnergySource> ();  
es8->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es9 = CreateObject<BasicEnergySource> ();  
es9->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es10 = CreateObject<BasicEnergySource> ();  
es10->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es11 = CreateObject<BasicEnergySource> ();  
es11->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es12 = CreateObject<BasicEnergySource> ();  
es12->SetSupplyVoltage(5.0);  
Ptr<BasicEnergySource> es13 = CreateObject<BasicEnergySource> ();  
es13->SetSupplyVoltage(5.0);
```

ANEXO D

Resultados obtenidos de la simulación escenario A en J (Joules) y transformados a W-h.

$$Wh = \frac{E (J)}{t} \quad t \text{ en } h$$

Protocolo TCP- Escenario A			
Tiempo (min)	Energía Consumida (J)	W-h	Acumulado
1	22.32	0.0062	0.0062
2	44.53	0.0124	0.0186
3	66.56	0.0185	0.0371
4	88.71	0.0246	0.0617
5	109.01	0.0303	0.0920
6	130.23	0.0362	0.1282
7	152.08	0.0422	0.1704
8	173.84	0.0483	0.2187
9	194.46	0.0540	0.2727
10	215.93	0.0600	0.3327
11	236.96	0.0658	0.3985
12	258.42	0.0718	0.4703
13	279.29	0.0776	0.5479
14	300.22	0.0834	0.6313
15	320.69	0.0891	0.7203
16	340.67	0.0946	0.8150
17	360.85	0.1002	0.9152
18	381.93	0.1061	1.0213
19	403.36	0.1120	1.1334
20	424.7	0.1180	1.2513
21	444.72	0.1235	1.3749
22	465.6	0.1293	1.5042
23	487.14	0.1353	1.6395
24	509.4	0.1415	1.7810
25	531.53	0.1476	1.9287
26	553.06	0.1536	2.0823
27	575.13	0.1598	2.2420
28	596.45	0.1657	2.4077
29	618.27	0.1717	2.5795
30	639.28	0.1776	2.7570
31	660.34	0.1834	2.9405
32	680.68	0.1891	3.1295
33	701.89	0.1950	3.3245
34	723.8	0.2011	3.5256

35	745.45	0.2071	3.7326
36	766.33	0.2129	3.9455
37	787.6	0.2188	4.1643
38	808.69	0.2246	4.3889
39	829.82	0.2305	4.6194
40	850.98	0.2364	4.8558
41	872.11	0.2423	5.0981
42	893.22	0.2481	5.3462
43	914.19	0.2539	5.6001
44	935.07	0.2597	5.8599
45	956.27	0.2656	6.1255
46	977.53	0.2715	6.3970
47	998.49	0.2774	6.6744
48	1019.4	0.2832	6.9576
49	1040.51	0.2890	7.2466
50	1062.47	0.2951	7.5417
51	1084.01	0.3011	7.8428
52	1104.89	0.3069	8.1497
53	1125.99	0.3128	8.4625
54	1146.77	0.3185	8.7811
55	1167.25	0.3242	9.1053
56	1188.01	0.3300	9.4353
57	1209.73	0.3360	9.7713
58	1231.64	0.3421	10.1135
59	1252.88	0.3480	10.4615
60	1274.18	0.3539	10.8154

ANEXO E

Resultados obtenidos de la simulación escenario B en J (Joules) y transformados a W-h

Protocolo TCP- Escenario B			
Tiempo (min)	Energía Consumida (J)	W-h	Acumulada
1	107.38	0.0298	0.0298
2	213.48	0.0593	0.0891
3	320.1	0.0889	0.1780
4	426	0.1183	0.2964
5	531.94	0.1478	0.4441
6	631.78	0.1755	0.6196
7	744	0.2067	0.8263
8	849.56	0.2360	1.0623
9	955.76	0.2655	1.3278
10	1061.9	0.2950	1.6228
11	1168.18	0.3245	1.9472
12	1274.58	0.3541	2.3013
13	1380.84	0.3836	2.6849
14	1487.2	0.4131	3.0980
15	1593.62	0.4427	3.5406
16	1700.12	0.4723	4.0129
17	1806.4	0.5018	4.5147
18	1912.4	0.5312	5.0459
19	2018.86	0.5608	5.6067
20	2124.88	0.5902	6.1969
21	2231.38	0.6198	6.8168
22	2337.28	0.6492	7.4660
23	2443.44	0.6787	8.1447
24	2549.76	0.7083	8.8530
25	2655.88	0.7377	9.5908
26	2761.8	0.7672	10.3579
27	2867.96	0.7967	11.1546
28	2974.52	0.8263	11.9808
29	3080.72	0.8558	12.8366
30	3187.12	0.8853	13.7219
31	3294	0.9150	14.6369
32	3399.88	0.9444	15.5813
33	3506	0.9739	16.5552
34	3611.96	1.0033	17.5585

35	3717.6	1.0327	18.5912
36	3823.88	1.0622	19.6534
37	3930.26	1.0917	20.7451
38	4036.2	1.1212	21.8663
39	4142.44	1.1507	23.0170
40	4248.8	1.1802	24.1972
41	4355.08	1.2097	25.4069
42	4461.06	1.2392	26.6461
43	4567.32	1.2687	27.9148
44	4673.76	1.2983	29.2131
45	4780.14	1.3278	30.5409
46	4886.18	1.3573	31.8982
47	4992.32	1.3868	33.2849
48	5098.62	1.4163	34.7012
49	5205.14	1.4459	36.1471
50	5310.94	1.4753	37.6223
51	5417.04	1.5047	39.1271
52	5523.24	1.5342	40.6613
53	5629.32	1.5637	42.2250
54	5735.86	1.5933	43.8183
55	5841.96	1.6228	45.4411
56	5947.86	1.6522	47.0933
57	6053.44	1.6815	48.7748
58	6159.42	1.7110	50.4857
59	6265.46	1.7404	52.2261
60	6371.52	1.7699	53.9960