

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**REDES DE SENSORES INALÁMBRICOS PARA IOT
AUTOMATIZACIÓN DE REDES INALÁMBRICAS DE SENSORES**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

KEVIN ALEXANDER CAJAS TAPIA

kevin.cajas@epn.edu.ec

DIRECTOR: CARLOS ROBERTO EGAS ACOSTA

carlos.egas@epn.edu.ec

DMQ, febrero 2022

CERTIFICACIONES

Yo, Kevin Alexander Cajas Tapia declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

KEVIN ALEXANDER CAJAS TAPIA

Certifico que el presente trabajo de integración curricular fue desarrollado por Kevin Alexander Cajas Tapia, bajo mi supervisión.

CARLOS ROBERTO EGAS ACOSTA
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

NOMBRE_ESTUDIANTE

NOMBRE_DIRECTOR

NOMBRE_COLABORADOR(ES)

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
ÍNDICE DE CONTENIDO.....	III
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABLAS	VII
RESUMEN	VIII
ABSTRACT	IX
1. INTRODUCCIÓN	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	3
2. MARCO TEÓRICO	4
2.1 Redes inalámbricas de sensores	4
2.1.1 Origen e historia.....	5
2.1.2 Puntos de inflexión en la historia de la WSN	6
2.1.3 Áreas de aplicación.....	7
2.1.4 Topologías de red.....	7
2.1.5 Estándares inalámbricos utilizados en redes inalámbricas de sensores.....	9
2.1.5.1 IEEE 802.15.4.....	10
2.1.5.1.1 Tipos de dispositivos en WSN	10
2.1.5.1.2 Tipos de nodos	11
2.1.5.1.3 Capas del estándar IEEE 802.15.4	12
2.1.5.1.4 Seguridad	13
2.1.5.2 6LoWPAN.....	14
2.1.5.3 Bluetooth.....	15
2.1.5.4 ISA 100.11a.....	15
2.1.5.5 LoRaWAN.....	15
2.1.5.6 NB-IoT	16
2.1.5.7 OCARI.....	16
2.1.5.8 Wireless Hart.....	17
2.1.5.9 Z-Wave	17
2.1.5.10 ZigBee	17
2.1.5.11 XBee.....	18

2.1.5.11.1	Digimesh.....	19
2.1.5.11.2	Firmware XBee	19
2.1.5.11.3	Ecosistema XBee	20
2.2	Python.....	20
2.2.1	Utilización de Python para automatización de redes de telecomunicaciones	21
2.2.2	MicroPython.....	21
2.2.3	Modos de operación de módulos XBee para programación con Python	22
2.2.3.1	Modo transparente	22
2.2.3.2	Modo API (Application Programming Interface).....	23
2.2.3.2.1	Estructura de la trama API	25
2.3	Automatización de redes.....	27
2.3.1	Conceptos base para la automatización en redes	28
2.3.1.1	Inteligencia artificial	29
2.3.1.1.1	Aprendizaje automático.....	29
2.3.1.1.2	Aprendizaje profundo	29
2.3.1.2	Redes Definidas por Software	30
2.3.1.2.1	Programabilidad de la red.....	31
2.3.1.3	Virtualización de las funciones de la red.....	32
2.3.2	Automatización de las redes de telecomunicaciones	33
2.3.3	Ventajas y desventajas de la automatización de la red	33
2.3.3.1	Ventajas.....	33
2.3.3.2	Desventajas	34
2.3.4	Automatización de operaciones de la red.....	34
2.3.5	Definición de Herramientas para automatización de redes de telecomunicaciones	35
2.3.6	Automatización de redes inalámbricas de sensores.....	35
3.	METODOLOGÍA	38
3.1	Investigación documental.....	38
3.1.1	Tipo de enfoque	39
3.1.2	Tipo de trabajo	39
3.1.3	Técnica de recolección de información.....	39
3.1.4	Instrumentos de recolección de información	40
4.	AUTOMATIZACIÓN DE LA WSN	41
4.1	Situación actual de la automatización de la WSN	41
4.2	Estudios relacionados con la automatización de WSN	43

4.3	Herramientas para automatización y gestión de WSN	46
4.3.1	StreetLight Control	46
4.3.2	Ginjer	47
4.3.3	XCTU	47
4.3.4	Digi Remote Manager.....	48
4.4	Automatización de redes inalámbricas de sensores con Python	50
4.4.1	Ventajas y desventajas de la WSN automatizadas con Python.....	50
4.4.1.1	Ventajas.....	50
4.4.1.2	Desventajas	52
4.5	Casos de automatización procesos en WSN	52
4.5.1	Automatización de WSN con Digi Remote Manager.....	52
4.5.2	Notificación de eventos de la WSN vía SMS.....	53
4.6	Desafíos de la automatización de redes inalámbricas de sensores	55
6.	CONCLUSIONES	56
7.	REFERENCIAS BIBLIOGRÁFICAS	57

ÍNDICE DE FIGURAS

Figura 2.1. Diagrama de bloques del nodo sensor. (Fuente: [2]).....	4
Figura 2.1. Evolución del mercado, costos, consumo energético y sector del mercado de la WSN. (Fuente [3]).....	6
Figura 2.2. Puntos de inflexión que marcaron la historia de las redes inalámbricas de sensores.....	7
Figura 2.3. Topología de redes WSN. (Fuente: [8]).....	9
Figura 2.4. Clasificación de redes inalámbricas. (Fuente: [9]).....	10
Figura 2.5. Asociación de un dispositivo a la red. (Fuente: [15]).....	13
Figura 2.6. Des-asociación de un dispositivo de la red. (Fuente: [15]).....	13
Figura 2.7. Red IPv6 con la red malla 6LoWPAN. (Fuente: [16]).....	14
Figura 2.8. Comparación entre LoRaWAN y NB-IoT. (Fuente: [19]).....	16
Figura 2.9. Nodos ZigBee vs nodos Digimesh. (Fuente [29]).....	19
Figura 2.10. Protocolos compatibles con ecosistema XBee. (Fuente: [34]).....	20
Figura 2.11. Ecosistema XBee. (Fuente: [34]).....	20
Figura 2.12. Secuencia de la comunicación en modo transparente. (Fuente: [44])	23
Figura 2.13. Comunicación en modo API. (Fuente: [44]).....	24
Figura 2.14. Trama API. (Fuente: [44]).....	26
Figura 2.15. Esquema de funcionamiento de las APIs. (Fuente: [45]).....	27
Figura 2.16. Funcionamiento de la IA con ML y con DL. (Fuente: [49]).....	30
Figura 2.17. Programabilidad de la red mediante API. (Fuente: [54]).....	32
Figura 3.1. Fases de la investigación documental. (Fuente: [61]).....	38
Figura 4.1. Factores que impulsan actualmente al desarrollo de la automatización de la red. (Fuente: [63]).....	42
Figura 4.2. Características de StreetLight Control. (Fuente: [69]).....	46
Figura 4.3. Diagrama de bloques de la plataforma Ginjer. (Fuente: [70]).....	47
Figura 4.4. Interfaz gráfica de XCTU. (Fuente: [71]).....	48
Figura 4.5. Aplicaciones de utilización para Digi Remote Manager. (Fuente: [72])	49
Figura 4.6. Notificación de la temperatura de un nodo sensor vía SMS. (Fuente: [79]).....	54

ÍNDICE DE TABLAS

Tabla 2.1. Bandas libres de radio. (Fuente: [15])	100
Tabla 2.2. Canales definidos en IEEE 802.15.4. (Fuente: [15])	122
Tabla 2.3. Ventajas y desventajas de los modos de operación. (Fuente: [44]) .	2424

RESUMEN

El crecimiento constante de la tecnología y de las telecomunicaciones permite que en la actualidad sea posible la utilización de dispositivos autónomos que se comuniquen entre si con protocolos de redes inalámbricas de área personal o WPAN (Wireless Personal Area Networks) por sus siglas en inglés, esto con el propósito de monitorear parámetros físico – ambientales, el conjunto de estos dispositivos se conoce como redes inalámbricas de sensores WSN (Wireless Sensor Network).

Conforme crecen la WSN se dificulta la gestión, aumenta los errores humanos y las incoherencias de esta, por lo que, para solventar este problema se propone la automatización de esta. En el presente trabajo se abordarán los conceptos básicos que componen la WSN y la forma en cómo es posible la automatización de esta red, además, revisar su situación actual.

PALABRAS CLAVE: automatización, sensores, Python, MicroPython, XBee, redes, inalámbrico, ZigBee, programación, monitoreo, WSN, WSNA, gestión, nodos.

ABSTRACT

The constant growth of technology and telecommunications makes it possible today to use autonomous devices that communicate with each other with wireless personal area network protocols or WPAN (Wireless Personal Area Networks) for its acronym in English, this for the purpose of monitoring physical-environmental parameters, the set of these devices is known as wireless sensor networks WSN (Wireless Sensor Network).

As the WSN grows, management becomes more difficult, human errors and inconsistencies increase, so, to solve this problem, its automation is proposed. In the present work, the basic concepts that make up the WSN and the way in which the automation of this network is possible will be addressed, in addition, its current situation will be reviewed.

KEY WORDS: automation, sensors, Python, MicroPython, XBee, networks, wireless, ZigBee, programming, monitoring, WSN, WSNA, management, nodes.

1. INTRODUCCIÓN

Junto con el avance constante de la tecnología, se han desarrollado varios campos de la industria, siendo uno de estos el internet de las cosas o IoT (Internet of Things), encontrando así, la forma de aplicar la innovación tecnológica con las cuales pueden sustituirse procesos, puesto que son repetitivos y de algún modo pueden llegar a automatizarse, es decir, la capacidad de una maquina a realizar un proceso sin necesidad de intervención humana.

Una de las innovaciones tecnológicas son las redes inalámbricas de sensores o WSN por sus siglas en inglés (Wireless Sensor Network), este tipo de redes facilita la recolección de información de un ambiente o de un entorno el cual se necesite conocer parámetros como temperatura, humedad, presión, vibraciones o algún otro parámetro físico – ambiental que sea de importancia monitorear. Por lo general, una red de sensores tiene una comunicación bidireccional por lo que permite configurar, enviar comandos y actuar, este último se conoce como WSAN (Wireless Sensors and Actuator Networks), es decir que el dispositivo puede realizar alguna acción programada de acuerdo con la información recibida del entorno reduciendo así tiempos de respuesta.

Las redes de sensores pueden llegar a ser útiles en varios campos, desde domótica hasta monitoreo y/o automatización de procesos industriales, los puntos donde se encuentran los dispositivos que conforman la WSN son conocidos como nodos y están distribuidos de acuerdo a lo que se necesite monitorear o controlar, como ventaja de una WSN se tienen la flexibilidad, escalabilidad y despliegue de la red, puesto que, estos dispositivos se conectan de forma inalámbrica, permitiendo así la fácil instalación y ubicación, sin estar limitado a la infraestructura existente o a la distancia.

Con el objetivo de disminuir costos operativos, eliminar errores humanos, modernizar y lograr la máxima eficiencia de la red, varias compañías líderes en telecomunicaciones sugieren automatizar las redes, ofreciendo servicios y software para empresas, sin embargo, no se tiene hasta el momento una automatización sólida a nivel de redes WSN, por esta razón en el presente documento se pretende estudiar la automatización de una red inalámbrica de sensores para reducir errores humanos, maximizar la eficiencia, automatizar procesos y gestionar la red.

Para automatizar la WSN es necesario definir herramientas de software y/o el lenguaje de programación con el cual se llevará a cabo la gestión, configuración y monitorización del estado de la red, existen lenguajes como Perl y Tool Command Language (TCL), sin embargo son complejos de programar y en algunos casos difíciles de entender debido a la

complejidad de la red, Python es otro lenguaje de programación usado para automatizar redes de telecomunicaciones, puesto que, es fácil, flexible y simple de comprender, el cual se analizara en este documento para comprender la automatización de la WSN mediante scripts.

La automatización de la WSN permitirá gestionar la red de manera simple, controlada y fácil de manejar de acuerdo con su tamaño, la programación de la red mediante scripts permite que ejecute tareas eliminando así la ejecución de acciones por comandos introducidos manualmente y por ende los errores humanos, Python es un lenguaje de programación sencillo el cual es fácil de entender tanto para la maquina como para el programador.

En conclusión, por lo explicado anteriormente es importante mantener una correcta gestión de la red sin ningún tipo de errores e incoherencias que disminuya la eficiencia de esta, puesto que los campos de aplicación de la WSN dependen de la precisión y eficacia de cada nodo, esto es posible con la automatización de la red implementado con Python en la programación de la tareas y procesos de la red o el uso de herramientas de automatización de la red.

1.1 Objetivo general

Analizar y entender la automatización de una red inalámbrica usando el lenguaje de programación Python, el cual, por su simplicidad, flexibilidad, fácil utilización y ser de código abierto es propicio para automatizar este tipo de red de telecomunicaciones.

1.2 Objetivos específicos

A continuación, se describirán los objetivos específicos que se pretenden cumplir con el presente trabajo.

- Analizar la programabilidad de redes inalámbricas de sensores con el lenguaje de programación Python.
- Analizar el estado actual de la automatización de redes inalámbricas de sensores.
- Investigar herramientas usadas para la automatización de redes.

1.3 Alcance

Al concluir el presente documento se pretende entender la arquitectura de la WSN y la automatización de redes de telecomunicaciones. También, se planea comprender la programabilidad de una red inalámbrica de sensores con un lenguaje de programación simple y fácil de comprender como lo es Python con el propósito de analizar la automatización de tareas y procesos como supervisión del estado de la WSN, configuración masiva de los nodos y despliegue de nuevos nodos, además, realizar un diagnóstico de la situación actual de las redes de sensores con el objetivo de analizar la viabilidad de implementación de la automatización en este tipo de redes para aplicaciones de IoT.

2. MARCO TEÓRICO

En este capítulo revisaremos brevemente conceptos claves para entender las redes inalámbricas de sensores y el lenguaje de programación Python.

2.1 Redes inalámbricas de sensores

En la actualidad las WSN son ampliamente utilizadas en el área industrial por su flexibilidad, bajo consumo energético, dispositivos compactos, fácil instalación, medición de parámetros físicos en tiempo real y la capacidad de comunicarse a una distancia relativamente corta. Estas características hacen que las redes inalámbricas de sensores sean la mejor opción al momento de recolectar información de un entorno establecido, siendo posible ubicarlas en cualquier lugar sin estar limitadas por espacio, instalación cableada u obstáculos gracias a su comunicación inalámbrica.

La WSN se conforma por nodos sensores o también conocidos como motas, los cuales están compuestos principalmente con una batería (fuente de alimentación energética), microprocesador, memoria para almacenamiento y procesamiento, uno o dos sensores y un módulo de transmisión/recepción o transceptor que permite la comunicación entre los nodos sensores.

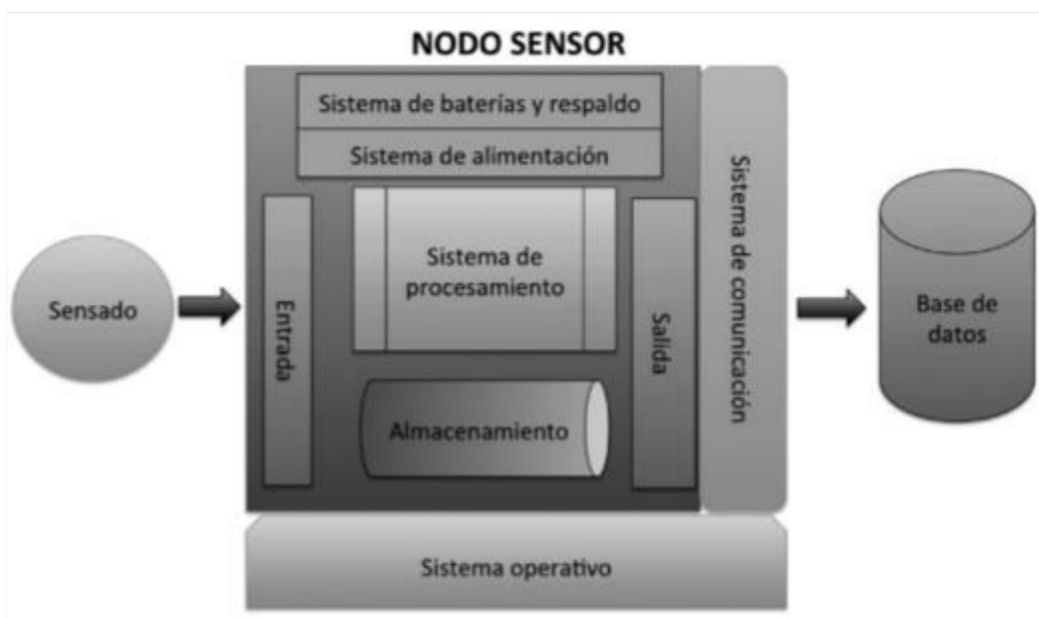


Figura 2.1. Diagrama de bloques del nodo sensor. (Fuente: [2])

De acuerdo con el propósito de la medición y la extensión del entorno se define la cantidad y ubicación de nodos para cumplir con la recolección de información necesaria, que una vez medido el parámetro físico por el sensor se convierte esta información análoga a digital gracias a un conversor ADC (Analog-to-Digital Converter) para su posterior procesamiento y transmisión a los nodos más cercanos dependiendo de la topología usada, de las tres topologías posibles a usarse, estas coinciden en que necesitan un Gateway, es decir una conexión fuera de la WSN a un servidor o a un computador directamente para un procesamiento de los datos a mayor capacidad, puesto que los nodos disponen de un procesamiento limitado, que soporta la recolección de datos, transmisión/recepción y almacenamiento de información momentáneo.

2.1.1 Origen e historia

Las redes inalámbricas de sensores han pasado de ser una tecnología emergente a ser una tecnología totalmente desarrollada en la actualidad gracias a la innovación y evolución de esta, sin embargo, como todo avance tecnológico de la historia de la humanidad, las redes inalámbricas involucrando sensores se crearon para aplicaciones militares alrededor de los años 50's por el departamento de defensa de Estados Unidos conocido como DARPA (Defense Advanced Research Projects Agency) por sus siglas en inglés.

La primera red inalámbrica que se tiene registrada y que guarda un parecido con la WSN actuales se conoció como SOSUS (Sound Surveillance System), la red fue creada y desarrollada por DARPA en 1950 con el propósito de detectar y atacar con rapidez a submarinos de la ex Unión Soviética en la guerra fría [3].

A inicios de los años 80's DARPA empezó a desarrollar aún más estas redes de sensores, iniciando así el programa DSN (Distributed Sensor Network) de esta forma comenzó el estudio de las ventajas, desventajas, desafíos, problemas y posibles soluciones a implementarse a las DSNs para lograr el óptimo funcionamiento.

La creación del programa DSN permitió en lo posterior la participación en la investigación y desarrollo a universidades asociadas a DARPA como Carnegie Mellon University y The Massachusetts Institute of Technology Lincoln Labs, con el paso del tiempo eventualmente se desarrolló la tecnología DSN a lo que conocemos actualmente como WSN, este cambio se dio junto al desarrollo de ARPAnet (Advanced Research Projects Agency Network) lo que conocemos hoy como internet también desarrollado por DARPA.

Con la evolución de la industria se abrió campo para la WSN, visto de esta forma por corporaciones líderes en su momento como IBM (International Business Machines) y Bell Labs, por lo que hallaron aplicaciones de estas redes para la industria, como automatización de procesos industriales en fábricas, distribución de energía, tratamiento de aguas residuales, medición de temperatura y humedad en ambientes hostiles, entre otras aplicaciones.

El principal problema que tenían la WSN de esta época eran sus dispositivos voluminosos, costos altos de implementación y operación, además, al no estar estandarizados los protocolos de comunicación utilizados eran patentados por diferentes corporaciones, obligando al usuario/industria de la WSN a permanecer con el fabricante, puesto que estos dispositivos solo eran compatibles con dispositivos del mismo fabricante.

2.1.2 Puntos de inflexión en la historia de la WSN

Desde el momento en que las WSN se comercializaron se conocía el potencial que podrían llegar a tener en la industria, por este motivo universidades y sectores de la industria apoyaron la iniciativa de estandarizar este tipo de redes y resolver problemas de ingeniería. El propósito que tenía cada una de las iniciativas creadas por fabricantes era el de desarrollar y estandarizar la WSN, pero también tenía como objetivo ampliar las aplicaciones de este tipo de redes.

Como consecuencia de la disminución de los costos de la producción e implementación de sensores para redes inalámbricas de sensores aumento la demanda en el mercado llegando a consumidores comunes, gracias a esto surgió el internet de las cosas o IoT (Internet of Things).

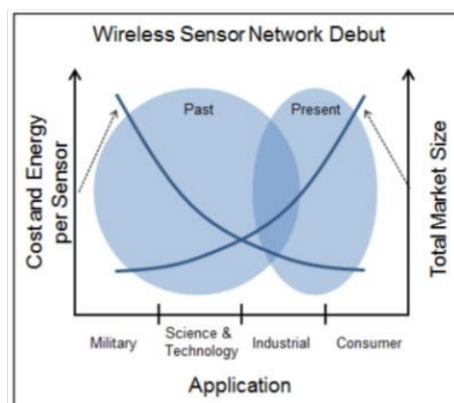


Figura 2.1. Evolución del mercado, costos, consumo energético y sector del mercado de la WSN. (Fuente [3])



Figura 2.2. Puntos de inflexión que marcaron la historia de las redes inalámbricas de sensores.

2.1.3 Áreas de aplicación

Existen varias aplicaciones para la WSN y prácticamente se encuentran en todos los sectores como la industria, salud, ciencia, hogar, ciudades o cualquier área en la cual permita una conexión a internet y necesite ser monitoreada o mejorar tiempos de respuesta frente a una acción que deba solucionarse de inmediato [5]. Las principales aplicaciones son seguridad industrial, automatización de procesos industriales, monitorización en la industria energética, agrícola y ganadera, medición de factores medioambientales, monitoreo de la vida silvestre, tráfico vehicular, hogar, ciudad, y edificación inteligente.

2.1.4 Topologías de red

La WSN puede utilizar tres topologías y combinaciones de las detalladas a continuación:

- **Topología de árbol**

En este tipo de topología se involucran tres elementos que son nodo final, un nodo que actúa como router y un nodo coordinador que actúa como Gateway el cual también se lo conoce como coordinador y es quien recolecta los datos para posteriormente transmitirlo a un servidor o directamente a un computador, en esta topología se tienen niveles interconectados por una conexión bus donde se encuentran los nodos routers y los nodos finales conectados a estos, la principal ventaja es la fácil expansión de la red, sin embargo, si la conexión bus se rompe toda la red colapsa.

- **Topología de malla**

Esta topología permite la fácil instalación y crecimiento de la red, puesto que, la comunicación de la información se realizará de un nodo a otro que este dentro del rango de transmisión de radio. En el caso que un nodo A necesite comunicarse con otro nodo B que esta fuera del rango de transmisión de radio es necesario tener un nodo C intermedio que recepte el mensaje y lo replique para enviarlo al nodo B, al igual que la topología de árbol esta también utiliza tres elementos, nodo final, nodo router y Gateway.

- **Topología de estrella**

Esta topología es la más simple de las tres, puesto que, solamente necesita de dos elementos que son el nodo final y el Gateway, los nodos finales se conectan directamente al Gateway por lo que, se reduce la latencia entre los dos puntos siendo una de las principales ventajas de esta topología, la otra ventaja se relaciona con el consumo energético ya que al estar relativamente más cerca del Gateway los nodos finales mantienen un nivel de energía bajo, sin embargo, la comunicación se restringe al área de transmisión de radio del Gateway, es decir que si un nodo esta fuera del rango de transmisión de radio de la puerta de enlace es imposible comunicarse.

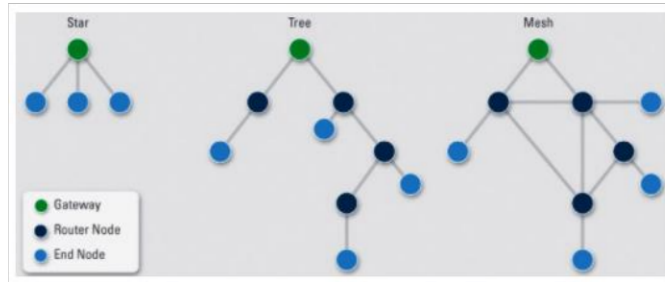


Figura 2.3. Topología de redes WSN. (Fuente: [8])

Para aplicaciones en campo de redes inalámbricas de sensores la topología que permite escalabilidad y flexibilidad de la red es la topología tipo malla o también conocida como mesh, por esta razón es ampliamente utilizada y se ha desarrollado continuamente, ejemplo de esto es la tecnología denominada Digimesh, la cual es una topología tipo malla inalámbrica con el objetivo de mejorar la eficiencia de la red con ayuda de módulos de comunicación llamados XBee, la cual se revisara más adelante.

2.1.5 Estándares inalámbricos utilizados en redes inalámbricas de sensores

En la actualidad existen gran variedad de estándares utilizados para trabajar en cada nodo de la red, en casos particulares y especiales la WSN puede trabajar con dos estándares. Dependiendo del fabricante, los dispositivos son creados para trabajar solo con un estándar o trabajar con alguno de los demás estándares conociéndose a estos dispositivos como multiprotocolo.

La utilización de un estándar en específico para trabajar en la WSN está en manos del diseñador de la red, puesto que, de acuerdo con los dispositivos adquiridos, características, uso y propósito de la red se escoge el estándar inalámbrico que mejor se ajuste a la necesidad del diseño.

Los estándares difieren entre sí en características de eficiencia energética, alcance y en algunos casos por ser propietarios de ciertos fabricantes. Se listan a continuación con sus características principales.

En este apartado se mencionarán los protocolos que pueden utilizarse para redes inalámbricas de sensores, siendo 6LoPAN, ZigBee y XBee los más utilizados basados en el estándar IEEE 802.15.4, por lo que, es necesario explicar sus características con el fin de comprender de mejor manera los protocolos mencionados.

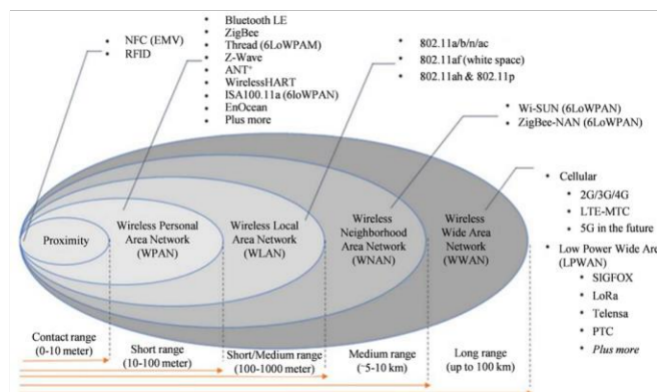


Figura 2.4. Clasificación de redes inalámbricas. (Fuente: [9])

2.1.5.1 IEEE 802.15.4

Desarrollado por IEEE (Institute of Electrical and Electronics Engineers) el estándar especifica la comunicación de las capas física (PHY) y de acceso (MAC) para redes WPAN (Wireless Personal Area Network) con tasas de transmisión bajas. Este estándar es el más básico de todos y por el cual se desarrollaron otros estándares basándose en este como ZigBee [15].

Tabla 2.1. Bandas libres de radio. (Fuente: [15])

Banda	Licenciada	Región	Velocidad de transmisión	Canales
2.405 - 2.480 GHz	No	Mundial	250 kbps	11 - 26
868.3 MHz	No	Europa	20 kbps	0
902 - 928 MHz	No	Estados Unidos	40 kbps	1 - 10

2.1.5.1.1 Tipos de dispositivos en WSN

El protocolo 802.15.4 considera dos tipos de dispositivos que conformar la red, con el objetivo de optimizar los recursos:

- **Dispositivos de función completa**

Llamados FDD (Full Function Device) por sus siglas en inglés, como indica su nombre estos dispositivos tienen la capacidad de tener varias funciones por lo que son usados en las redes ZigBee como coordinadores o routers.

- **Dispositivos de función reducida**

Los RFD (Reduced Function Device) son dispositivos que están limitados solo a comunicarse con los demás dispositivos de la red como coordinadores en topología estrella o con los routers en topología clúster y malla, adicional, estos dispositivos pueden hacer uso de la función llamada "sleep" la cual permite que se ahorre energía de la batería que alimenta al nodo, siendo posible un consumo muy bajo.

2.1.5.1.2 Tipos de nodos

Dentro del estándar se determina tres tipos de nodos que permiten la comunicación, se explican a continuación:

- **Nodo coordinador**

En cada red basada en este estándar es necesario uno y solo un nodo coordinador por red, no pueden existir dos coordinadores o más, puesto que este es el que crea la red y coordina la red PAN (Personal Area Network), la capacidad de procesamiento de este es alto por ser un dispositivo con funciones completas (FFD). El nodo coordinador también se lo conoce como Gateway, ya que permite que los datos recolectados por cada nodo salgan de la red hacia un servidor o un computador para su posterior procesamiento.

- **Nodo Router**

Al igual que el nodo coordinador es un dispositivo del tipo FFD, sin embargo, no es un coordinador. Las funciones que tiene este tipo de nodo es ampliar la cobertura de la red utilizando el enrutamiento multi-hop, además de aumentar la disponibilidad de la red al 100%.

- **Nodo final**

Estos nodos son de tipo RFD, puesto que solo se comunican con un router o un coordinador, al tener funciones reducidas no es necesario que el nodo disponga de

gran cantidad de procesamiento y potencia, estos dispositivos pueden activar el modo sleep, esto significa que se reduce al mínimo el consumo de energía, sin embargo, esta alerta para recibir la señal de transmisión o recepción de un nodo coordinador o un nodo router.

2.1.5.1.3 Capas del estándar IEEE 802.15.4

- **Capa física (PHY)**

El estándar IEE 802.15.4, tiene relación con la capa de datos o también conocida como capa MAC, esta capa define los parámetros físicos importantes para la comunicación, tales como potencia de transmisión y la sensibilidad que debe tener el receptor para recibir los datos sin errores o pérdidas de paquetes, además, esta capa también modulación y canales que trabajara la comunicación.

Tabla 2.2. Canales definidos en IEEE 802.15.4. (Fuente: [15])

Número de página	Canal	Banda [MHz]	Modulación
0	0	868	BPSK
	1 - 10	915	BPSK
	11 - 26	2400	O-QPSK
1	0	868	ASK
	1 - 10	915	ASK
	11 - 26	Reservado	-
2	0	868	O-QPSK
	1 - 10	915	O-QPSK
	11 - 26	Reservado	-
3 - 31	Reservado	Reservado	-

- **Capa de acceso al medio (MAC)**

Esta capa se la conoce como capa MAC en el estándar ZigBee por sus siglas en inglés (Media Access Control), la capa MAC actúa como una interfaz que permite la comunicación entre la capa PHY y la capa NWK, ya que en esta capa se ocupa de crear bloques de información conocidas como tramas, permite un enlace seguro entre los dispositivos que están comunicándose por medio de la MAC de cada dispositivo, provee de servicios de asociación o des asociación de un dispositivo en la red.

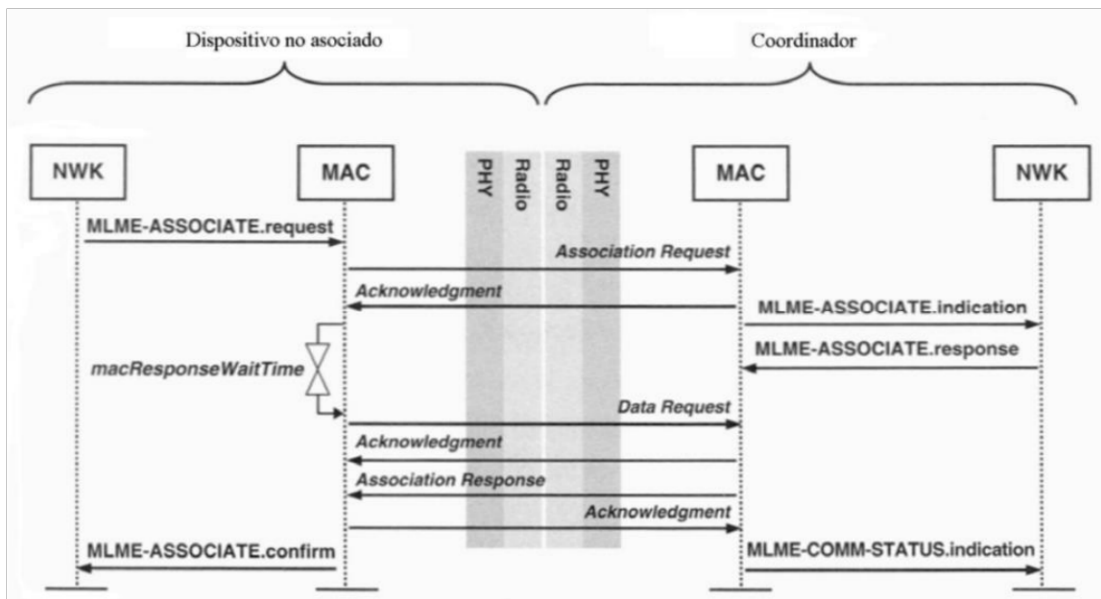


Figura 2.5. Asociación de un dispositivo a la red. (Fuente: [15])

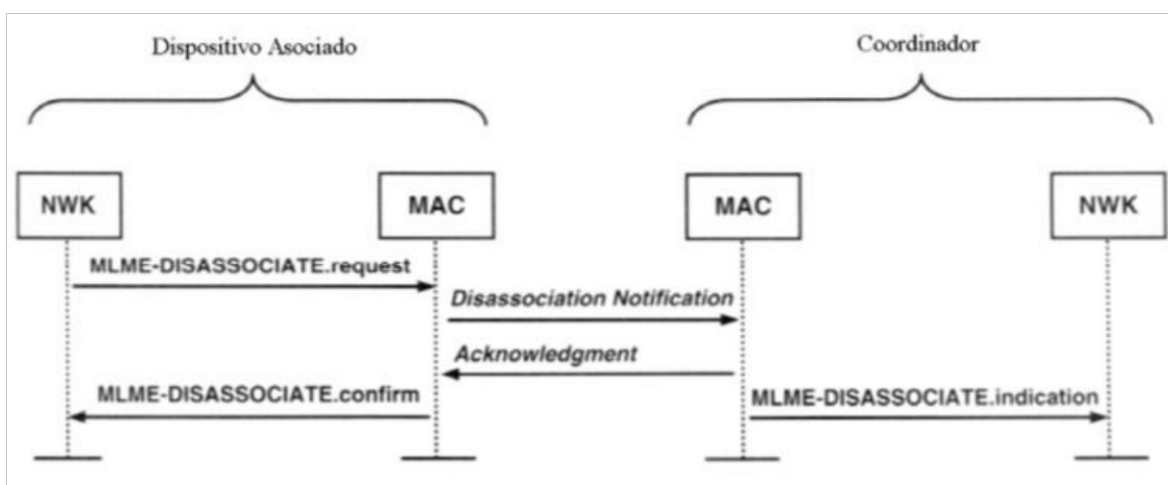


Figura 2.6. Des-asociación de un dispositivo de la red. (Fuente: [15])

2.1.5.1.4 Seguridad

Al transmitirse la información de manera inalámbrica y que cada mensaje enviado por el nodo puede ser receptado por cualquier dispositivo que se encuentre dentro del rango de transmisión/recepción, existe la posibilidad que dicho mensaje sea interceptado por un tercero con intenciones maliciosas o inhabilitar el sistema de comunicación, por tal motivo se usa protocolos de encriptación y autenticación, siendo de esta forma como se garantiza que la información no este vulnerable, cabe mencionar que entre más seguro sea el

estándar más consumo de recursos ocupara como memoria, procesamiento y por consecuencia consumo energético.

El estándar IEEE 802.15.4 ofrece una comunicación en las cuales cada capa permite un cierto nivel de seguridad, las capas relacionadas por el estándar es la capa física (PHY) y la capa de acceso al medio (MAC). La capa MAC provee servicios de comunicación de forma directa entre los dispositivos, además asegura la fiabilidad de las conexiones.

2.1.5.2 6LoWPAN

El nombre de este estándar es un acrónimo, pues se creó a partir de tres tecnologías por el IETF (International Engineering Task Force). El estándar 6LoWPAN es una red inalámbrica de área personal de baja potencia la cual está basada en IPv6 y en el estándar IEEE 802.15.4. Se desarrollo con el propósito de cubrir un área geográfica grande a bajo costo, cómputo y potencia.

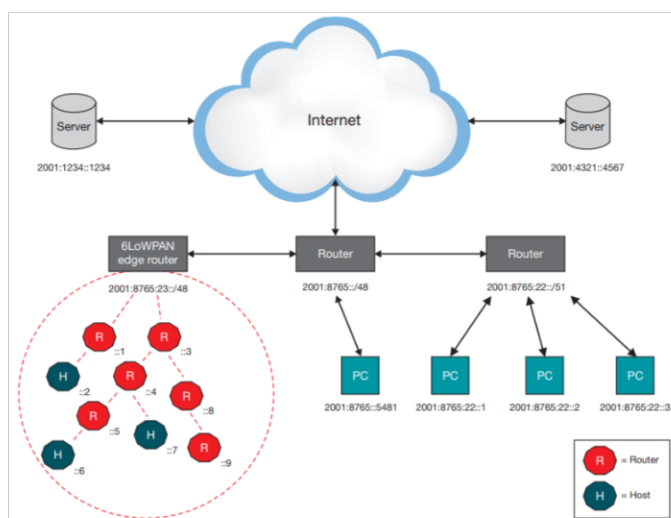


Figura 2.7. Red IPv6 con la red malla 6LoWPAN. (Fuente: [16])

Fue desarrollado para permitir la conexión a redes IP existentes sin tener la necesidad de realizar alguna traducción en el gateway a IPv4 o a IPV6. La integración de IPv6 permite la conexión a internet con tasas de velocidad bajas desde 20 kbps hasta 250 kbps a una distancia de 10 metros y una latencia reducida. Las limitaciones de 6LoWPAN son las fallas en los enlaces y la seguridad de extremo a extremo.

En la red 6LoWPAN el nodo router direcciona la información al nodo coordinador que es la puerta de enlace a la red IP y el nodo final que recolecta los datos para posteriormente enviarlos al nodo router.

2.1.5.3 Bluetooth

Bluetooth es el estándar más popular a nivel mundial pues se encuentra integrado en cualquier smartphone, laptop o tabletas, además, ofrece una conectividad eficiente, por este motivo se expandió su utilidad a la WSN, las características de transmisión permiten trabajar con potencias entre 1mW a 100mW para alcanzar distancias cortas que van de 1 metro hasta 30 metros en la banda de frecuencia no licenciada de 2.4 GHz.

Se han implementan mejoras de este estándar, desarrollando Bluetooth LE (Low Energy) para comunicación punto a punto y Bluetooth Mesh para comunicaciones punto a punto y topología estrella o malla.

2.1.5.4 ISA 100.11a

Este estándar nace del grupo de trabajo ISA 100 con el objetivo de proporcionar una pila de protocolos sólidos para la industria como automatización y control de procesos, la topología con las que puede trabajar son malla, estrella y una híbrida de estas.

La pila de protocolos se basa en los estándares IEEE 802.15.4 e IPv6, permitiendo la compatibilidad a nivel de capa física y capa de acceso al medio, además, al usar IPv6 también es compatible con 6LoWPAN a nivel de capa transporte y de capa de red.

Las características que proporciona el estándar son confiabilidad, soporte para múltiples protocolos, seguridad, robusticidad en la comunicación, flexibilidad y compatibilidad para trabajar con otros estándares inalámbricos gracias a los protocolos basados de este estándar [17].

2.1.5.5 LoRaWAN

LoRaWAN parte del estándar de comunicación inalámbrica LoRa el cual utiliza una modulación de radiofrecuencia patentada y administrada por LoRa Alliance, esta modulación es ampliamente usada para comunicaciones espaciales y militares, se conoce como CSS (Chirp Spread Spectrum) [17].

Ideal para construir redes inalámbricas de sensores para cubrir grandes áreas geográficas, pues permite un alcance desde 10 km hasta 20 km con tasas de transmisión baja, por lo que el consumo energético se reduce al mínimo dando como resultado duración de la batería de 15 años aproximadamente. Esta tecnología fue desarrollada ampliamente hasta lograr alta tolerancia a la interferencia y obtener una sensibilidad en la recepción de -168 dB con conexión punto a punto y topología estrella. La arquitectura de la red LoRaWAN está clasificada por gateways y nodos. Es escalable, seguro, interoperable con dispositivos que usen el mismo estándar y permite hasta un millón de nodos conectados en la red.

2.1.5.6 NB-IoT

Al igual que LoRaWAN este estándar está orientado a satisfacer necesidades del mercado de IoT para aplicaciones que requieran de gran cobertura de área geográfica con bajos costes de implementación, mantenimiento, consumo de potencia bajo, tasas de transmisión bajas, eficiente y con capacidad de soportar varios nodos en la red.

NB-IoT ofrece un tiempo de vida útil de las baterías alrededor de 10 años con latencias menores a 10 segundos, además, utiliza la infraestructura de la telefonía móvil para desplegar la red de nodos, existiendo una evolución de esta tecnología conocida como LTE-M, la cual ofrece características mejoradas de NB-IoT con respecto al ancho de banda, tasas de transmisión y latencia.

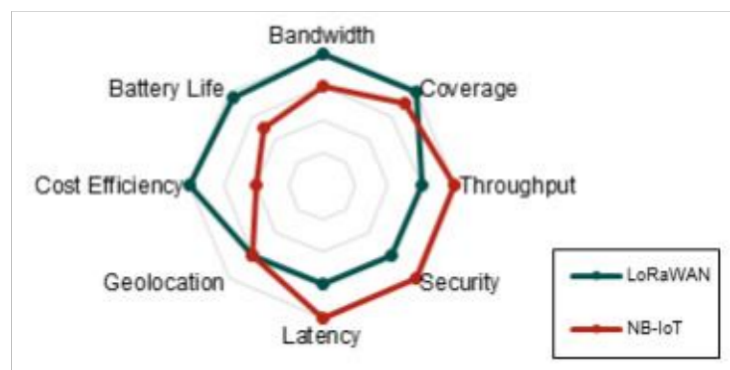


Figura 2.8. Comparación entre LoRaWAN y NB-IoT. (Fuente: [19])

2.1.5.7 OCARI

Este estándar aún se encuentra en desarrollo, sin embargo, promete extender al máximo la vida útil de la red y cumplir con las exigencias de comunicación para entornos hostiles

como la industria eléctrica de alta potencia, transportes en campos de guerra, entre otras aplicaciones. La arquitectura de la red de este estándar consiste en tres componentes que son nodo final, nodo coordinador y nodo ruteador, OCARI busca mejorar la pila de protocolos de ZigBee por lo que guarda ciertas similitudes [18].

Para prolongar la vida útil de la red, OCARI utiliza un enrutamiento que logra tener un eficiente uso de la energía, además, la capa de acceso propietaria de este estándar coordina horarios de sueño regulares para los nodos de la red, ayudando así al ahorro de energía y garantizando un bajo consumo de potencia.

2.1.5.8 Wireless Hart

Es un estándar industrial para aplicaciones de automatización de procesos desarrollado por HCF (HART Communication Foundation), la pila de protocolos de Wireless Hart está basado en comunicación HART, en cuanto para la capa física y MAC se basa en el estándar IEEE 802.15.4, sin embargo, no es compatible con otros estándares de comunicación inalámbrica que estén basados de igual manera en IEEE 802.15.4 [18].

Como principales características se encuentra la eficiencia energética, escalable, interoperable con demás dispositivos de HCF, permite enrutamiento redundante, es robusto, simple de implementar, autosustentable para curación y organización. La topología implementada en este tipo de redes son malla, estrella e híbrida de estas dos.

2.1.5.9 Z-Wave

Z-Wave es un estándar propietario conformado por diferentes fabricantes orientados a la producción de dispositivos para IoT en hogares, es similar a ZigBee en comparación con el consumo de potencia el cual es bajo con tasas de transmisión bajas desde 9.5 kbps hasta 100 kbps. No es compatible con otros estándares inalámbricos de comunicación, permite topologías de tipo malla y estrella. Al estar orientado para el hogar tiene una cantidad máxima de nodos de 232 con alcance de hasta 100 metros [24].

2.1.5.10 ZigBee

ZigBee es estándar de comunicación inalámbrica que fue creado en 2004 por ZigBee Alliance con el propósito de satisfacer necesidades particulares y únicas de redes

inalámbricas de IoT, en los cuales es primordial el bajo consumo energético y comunicación a corto alcance donde no es posible implementar una red cableada o el lugar es de difícil acceso, este estándar está basado en el protocolo IEEE 802.15.4 donde especifica la comunicación a nivel físico y de acceso [25].

La topología de red que puede usarse es en estrella, en malla y en clústeres de árboles, una red ZigBee puede alcanzar 254 nodos dependiendo del tipo de topología, por lo que si se usa una topología de clústeres se tendría 255 clústeres y en cada clúster 254 nodos, dando un total de 64770 nodos como máximo.

Para lograr la comunicación entre los nodos, estos pueden realizar un enrutamiento denominado multi-hop o saltos múltiples, las principales ventajas que tiene es la capacidad de cubrir superficies grandes del entorno que se necesita realizar las mediciones, también, si ocurriera el caso de que un nodo no funcione correctamente o este fuera de servicio, la red busca otro camino para llegar al destino teniendo así una disponibilidad de la red del 100%.

2.1.5.11 XBee

XBee es una tecnología propietaria de Digi International Inc. y es el nombre de una familia de módulos de radio considerablemente compactos para las funciones y ventajas que ofrece, los cuales permiten la comunicación inalámbrica entre estos, además, los módulos tienen entradas y salidas tanto analógicas como digitales, lo cual permite conectarlos con sensores para lectura de mediciones o conectarlos con actuadores para realizar cierta acción, todo esto de forma remota [27].

El estándar IEEE 802.15.4 y ZigBee están relacionados con XBee, puesto que esta se basa en estos dos protocolos, por lo cual XBee permite la creación de redes punto a multipunto o redes punto a punto, al estar basado en ZigBee ofrece características iguales o mejoradas de este protocolo como un consumo bajo de energía permitiendo de esa manera tener ciclos largos de su vida útil, baja latencia para tener tiempos de respuesta óptimos, una tasa de tráfico de datos alta y sincronización en la comunicación.

Se expande el estudio de este estándar con el objetivo de comprender la automatización de la WSN, puesto que, se encontró información más detallada para la realización del presente trabajo para este estándar, por lo que se evidencia que los demás estándares no han avanzado en la automatización de la WSN con lenguajes de programación como lo ha hecho XBee.

2.1.5.11.1 Digimesh

El estándar Digimesh fue desarrollado y patentado por Digi International Inc. y optimizada para trabajar con ecosistema XBee, tiene una topología tipo malla propietaria con comunicación punto a punto para conectividad inalámbrica, este estándar es usado en WSN por varias diferencias y ventajas sobre el resto de los estándares, la red que proporciona el estándar Digimesh cuenta con un solo tipo de nodo, es decir todos los nodos miembros de la red son del mismo tipo como una red homogénea [28].

La principal ventaja que tiene este estándar es el ahorro de consumo energético de los nodos de la red, debido a que todos los nodos son del mismo tipo y no existen nodos coordinadores, ruteadores, finales y ruteadores de frontera, permiten la posibilidad de dormir a los nodos cuando no estén comunicándose y despertando solo cuando se tiene que transmitir información, por esta razón a estos nodos se los conoce como nodos durmientes. Aun estando los nodos en modo sleep permanecen sincronizados en el tiempo.

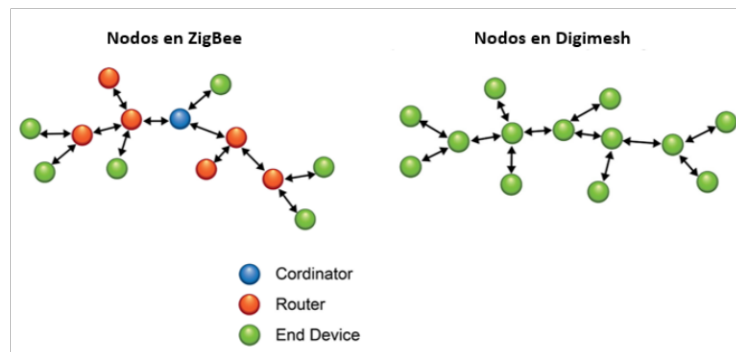


Figura 2.9. Nodos ZigBee vs nodos Digimesh. (Fuente [29])

2.1.5.11.2 Firmware XBee

El firmware es el código del programa que controla los circuitos del módulo, se encuentra almacenado dentro de la memoria y como cualquier programa puede ser actualizado o sustituirse por otro, para realizar estas acciones XBee ofrece diferentes métodos para hacerlo. Dentro de los métodos está el uso de la programación por medio de librerías Python XBee o métodos de configuración con programas como XCTU o interfaces graficas de XBee.

2.1.5.11.3 Ecosistema XBee

Las familias de XBee permiten formar un completo ecosistema de acuerdo con las aplicaciones y el área en el cual los módulos trabajarán. El ecosistema XBee además de ofrecer módulos para proyectos relacionados con IoT también disponen de herramientas que permiten la gestión, conexión y supervisión de la red XBee.



Figura 2.10. Protocolos compatibles con ecosistema XBee. (Fuente: [34])

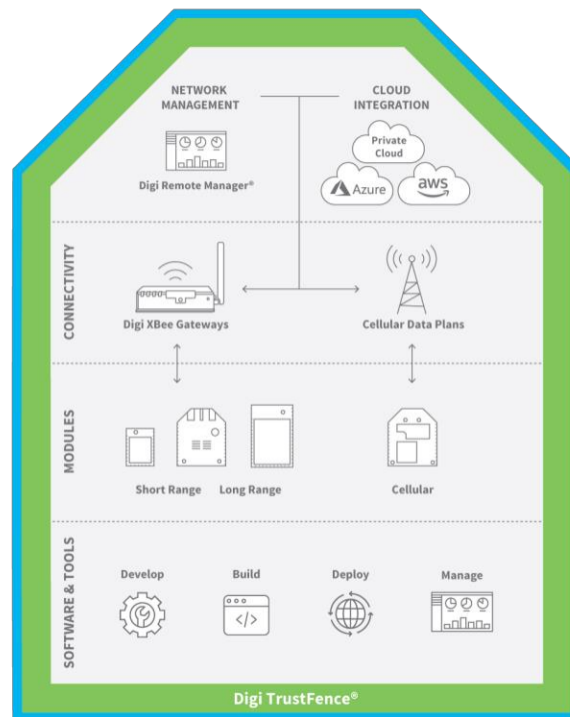


Figura 2.11. Ecosistema XBee. (Fuente: [34])

2.2 Python

Python fue publicado en febrero del año 1991 y es un lenguaje de programación de alto nivel con una curva de aprendizaje corta en comparación con otros lenguajes, esto se debe a que es fácil de entender y rápido de aprender por parte del programador, además, la

ejecución de las líneas de código lo hace en tiempo real o al finalizar la programación del código, ya que no compila el código como lenguajes de programación de bajo nivel, a esto se lo conoce como scripting pues existe un intérprete que procesa las líneas de código para que la máquina entienda y ejecute [37]. La sintaxis de este lenguaje es entendible para el programador y las librerías disponibles en su biblioteca permite que su uso sea extensible para cualquier aplicación como procesamiento de datos, análisis de datos, automatización de procesos, inteligencia artificial, desarrollo web, etc.

2.2.1 Utilización de Python para automatización de redes de telecomunicaciones

El uso de Python para la automatización de las redes permite aumentar la eficiencia de la red, con la automatización de procesos y tareas de supervisión, despliegue de nuevo equipamiento e identificación pronta de problemas relacionadas con la red. La programación de la red a través de scripting da paso a desarrollar la automatización mediante redes definidas por software o SDN por sus siglas en inglés, siendo este un punto importante para la automatización de una red de gran tamaño, pues ofrece la gestión de toda la red de manera remota, fácil, simple, flexible y disminuyendo costos operativos para la administración de los dispositivos de la red. Para llevar a cabo la automatización de la red se hacen uso de APIs, las cuales se revisarán a detalle más adelante.

La automatización de las redes de telecomunicaciones ha sido posible con la integración de las redes SDN, la cual permite la programabilidad de la red con lenguajes de programación como Python o JSON, actualmente es común observar la automatización de redes de telecomunicaciones, dado es el caso de las redes SD-WAN (Software Defined – Wide Area Network) la cual está diseñada para que la red se ajuste dinámicamente a las variaciones fluctuantes de la WAN, omitiendo así la intervención humana de un administrador de la red [42], [43]. Sin embargo, para redes WSN la situación de la automatización se complica, por las limitadas características de hardware que tienen los nodos sensores, por esta razón cuando surgió MicroPython se trató de extender la automatización a este tipo de redes también.

2.2.2 MicroPython

MicroPython es una versión con funciones reducidas del lenguaje de programación Python, lo que permite optimizar las funciones que dispone MicroPython para funcionar en

microcontroladores o en ambientes restringidos, lo que lo hace ideal para trabajar en aplicaciones o realizar funciones orientadas a los nodos en la WSN, donde el microcontrolador se encuentra limitado por el procesamiento de este.

Características de MicroPython:

- Dispone de gran diversidad de librerías para su programación.
- El editor de código que tiene es sencillo de usar y es propio para este lenguaje.
- Es extensible, es decir que puede integrarse líneas de comando de otros lenguajes de programación de bajo nivel al código realizado en MicroPython.

MicroPython al estar orientado a la programación de microcontroladores permite su utilización en los módulos XBee, los cuales dan lugar a la automatización de la WSN, considerando las limitaciones de hardware y procesamiento que tienen los nodos.

2.2.3 Modos de operación de módulos XBee para programación con Python

Los módulos XBee permiten la programabilidad de la red con Python, por tal motivo se estudia los modos de operación a profundidad de estos módulos con el propósito de entender la manera en la que trabaja.

Para desarrollar la programación de la red inalámbrica de sensores los módulos XBee pueden trabajar de forma independiente o conectado a un microcontrolador, de manera que el módulo al trabajar de forma independiente envía directamente los datos al nodo coordinador, por otro lado, al trabajar conectado a un microcontrolador los datos pueden ser procesados o no por este para luego enviarlos por el puerto serie al módulo XBee para ser transmitidos inalámbricamente.

Cuando un módulo XBee está conectado a un microcontrolador por medio de puerto serie se debe configurar el modo de operación, siendo estos:

2.2.3.1 Modo transparente

En este modo la información recibida por el puerto serie del módulo XBee se envía inalámbricamente sin modificación y de igual manera cuando el módulo XBee recepta inalámbricamente la información la envía sin ningún cambio al puerto serie.

Trabajar en este modo tiene limitaciones como que no puede comunicarse con más de un módulo a la vez, puesto que el módulo que envía el mensaje debe ser configurado con la dirección destino del módulo al que se le envía el mensaje, por esta razón es ideal para trabajar con una comunicación punto a punto, ya que si se desea enviar un mensaje a otro módulo es necesario volver a configurar al módulo origen con la dirección del módulo destino.

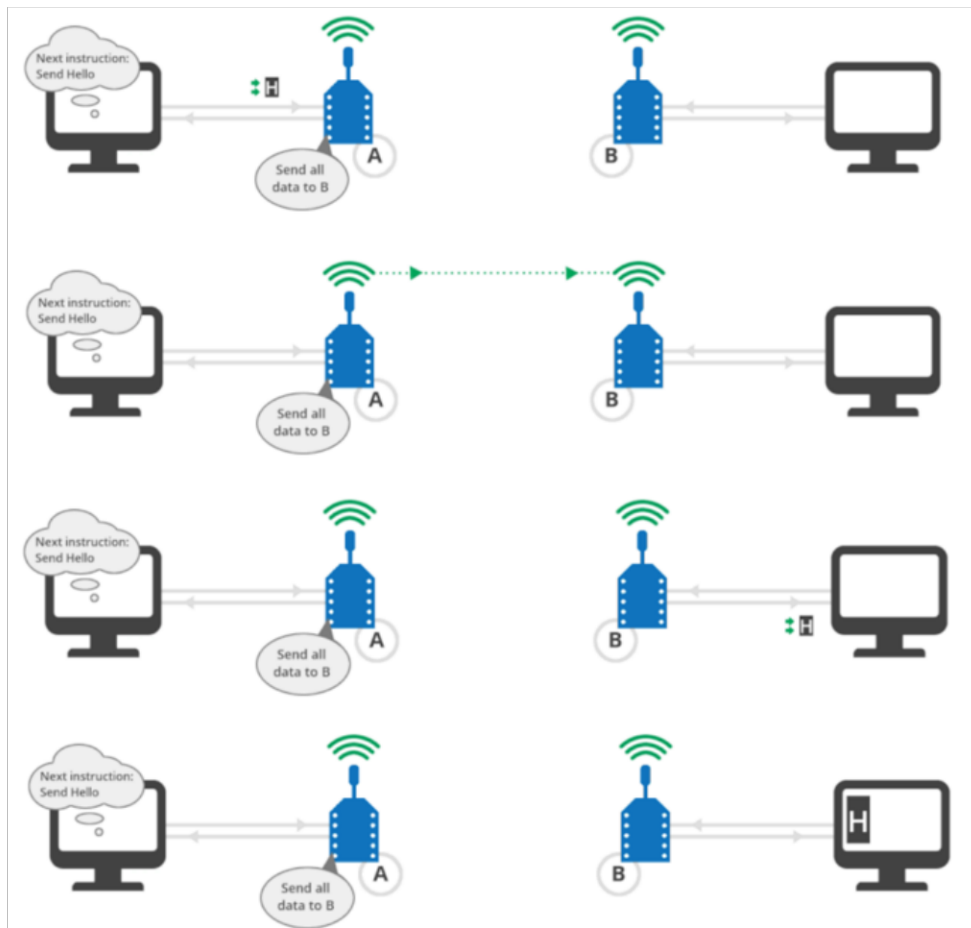


Figura 2.12. Secuencia de la comunicación en modo transparente. (Fuente: [44])

2.2.3.2 Modo API (Application Programming Interface)

El modo API permite una comunicación organizada con una interfaz estructurada que ofrece grandes ventajas en comparación con el modo transparente. La comunicación que se realiza en este modo es con la ayuda de paquetes o llamados también tramas API, además, permite una conexión punto a multipunto lo que a su vez permite crear una red mucho más grande, lo que es ideal en una red inalámbrica de sensores. Como principal característica cabe mencionar que este modo permite una configuración remota del módulo

y por consecuencia una automatización de la red de sensores con el lenguaje de programación Python.



Figura 2.13. Comunicación en modo API. (Fuente: [44])

A continuación, se muestra una tabla con las ventajas y desventajas que tienen estos modos de operación:

Tabla 2.3. Ventajas y desventajas de los modos de operación. (Fuente: [44])

Modo de Operación	Ventajas	Desventajas
Transparente	<ul style="list-style-type: none"> • Interfaz fácil y sencilla de utilizar. • Envío y recepción de datos directamente. • Comunicación bidireccional sin complicación. 	<ul style="list-style-type: none"> • No puede leer o escribir remotamente la configuración de la red. • Antes de enviar un mensaje a otro destino primero debe configurarse la dirección destino en el módulo origen. • En la trama recibida no se incluye la dirección del módulo origen. • En la trama recibida no se tiene información de la transmisión ni el éxito o fracaso.

		<ul style="list-style-type: none"> • No tiene funciones avanzadas.
API	<ul style="list-style-type: none"> • Permite leer y escribir la configuración de elementos dentro de la red. • Transmisión de datos a varios destinos. • El módulo destino conoce la dirección del módulo origen gracias a la trama enviada. • Funciones avanzadas (diagnóstico de la red y actualización del firmware). • La trama recibida incluye información de la transmisión y éxito o fracaso. 	<ul style="list-style-type: none"> • La trama es muy compleja y tiene un formato específico para su transmisión. • Los paquetes están estructurados por lo que es necesario analizarlos, tanto para obtener como enviar datos. • Los paquetes enviados contienen información adicional y datos de control, por lo que los datos recibidos son diferentes a los enviados.

2.2.3.2.1 Estructura de la trama API

Debido a que estándar XBee utiliza API para la automatización de la red, se detalla la estructura de la trama para comprender su funcionamiento. Los paquetes enviados de forma estructurada se consideran tramas en el modo API son enviados y recibidos por el puerto serie del módulo XBee, además, la trama contiene información de la transmisión, el mensaje inalámbrico, calidad de la señal, dirección de destino y dirección de origen de los datos. Al ser configurado el módulo XBee en modo API todos los datos enviados y recibidos deben pasar por el puerto serie, ya que estos datos se encuentran estructurados dentro de una trama que contiene información adicional como transmisión y eventos u operaciones del dispositivo. La trama especifica que el puerto serie del módulo XBee descartará toda información que se encuentre antes del delimitador de inicio (Start delimiter). Por lo que si la trama no es recibida con la estructura correcta o el cálculo de la suma de verificación

(Checksum) falla los datos son descartados, respondiendo el módulo destino al módulo origen con una trama que contiene el motivo de la falla.

Start delimiter	Length		Frame data								Checksum
			Frame type	Data							
1	2	3	4	5	6	7	8	9	...	n	n+1
0x7E	MSB	LSB	API frame type	Frame-type-specific data							Single byte

Figura 2.14. Trama API. (Fuente: [44])

Los campos de la trama API son:

- **Delimitador de inicio (Start delimiter)**

Es el primer byte de la trama, siempre tiene un valor de 0x7E que consta de una secuencia especial de bits. El objetivo de este campo es facilitar la detección de una nueva trama que llega por el puerto serie.

- **Longitud (Length)**

Especifica la longitud de bytes de la trama de datos excluyendo los campos: longitud, delimitador de inicio y suma de verificación. Tiene un valor de 2 bytes compuestos por el byte más significativo llamado como MSB (The Most Significant Byte) y el byte menos significativo llamado como LSB (The Least Significant Byte).

- **Trama de datos (Frame data)**

Este campo contiene la información a transmitir o recibir según sea el caso, el valor del campo está definido por la cantidad de datos, además, está compuesta por dos subcampos. El subcampo tipo de trama API (API frame type) en el cual se especifica el tipo de trama y como se organiza la información en el subcampo de datos. El subcampo de datos se encuentra solamente la información a transmitir o recibir.

- **Suma de verificación (Cheksum)**

Este campo es el último byte de la trama y permite comprobar que la información enviada no haya sido modificada por el medio físico o algún tercero. Para el cálculo se excluyen los primeros tres bytes de la trama es decir los campos longitud y delimitador de inicio.

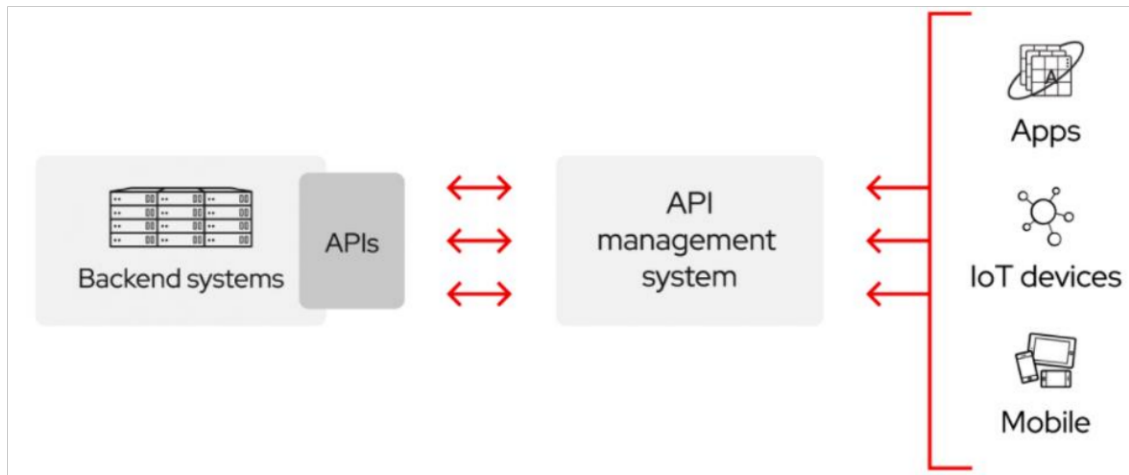


Figura 2.15. Esquema de funcionamiento de las APIs. (Fuente: [45])

2.3 Automatización de redes

La automatización de la red consiste en eliminar tareas manuales para gestionar y/o configurar la red utilizando herramientas de software o scripts con lenguajes de programación de alto o bajo nivel que la red pueda ejecutar de forma autónoma o semiautónoma. Siendo el objetivo de la automatización optimizar los recursos de la red y a su vez aumentar su eficiencia.

La automatización de redes de telecomunicaciones tiene varios años de desarrollo e implementación, junto con la evolución de la inteligencia artificial y el aprendizaje autónomo de esta, la automatización crea un ciclo cerrado del cual aprende con ayuda del aprendizaje automático para solucionar problemas sin intervención humana, aumentando la disponibilidad del servicio de la red, eficiencia operativa y disminuyendo errores en la red causados por gestión y/o configuración manual.

La automatización de las redes de telecomunicaciones nace de la necesidad de gestionar, configurar y monitorear la red con gran cantidad de dispositivos y datos que al realizarlo de forma manual no es posible abastecer los requerimientos de la red aumentando la probabilidad de error humano que afectan en el rendimiento, eficiencia y confiabilidad de

la red. Actualmente, el 95% de los cambios en la red se lo realizan manualmente por lo que tiene un gran impacto negativo en el tiempo que la red permanece inactiva por problemas presentados debido a la configuración manual [46].

Se tiene un amplio campo de procesos y tareas que pueden llegar a automatizarse con la automatización de la red, de los cuales se enlistan a continuación [47]:

- Pruebas de funcionamiento de dispositivos remotamente.
- Verificación de la configuración cargada.
- Administración de inventario.
- Planificación de la red.
- Despliegue de dispositivos físicos o virtuales de la red.
- Recopilación de datos (dispositivos, tráfico de red, software, topología y servicios en tiempo real) con posterior análisis y predicción a futuro del comportamiento de la red.
- Actualización del software a varios dispositivos de la red de forma remota.
- Supervisión de la red con el propósito de garantizar los SLAs del servicio al cliente final.
- Solucionar problemas presentados en la red.
- Verificación de cumplimiento de la seguridad de la red.
- Generación de reportes, alertas y alarmas.

A medida que todos los servicios web y de telecomunicaciones se conectan a la nube las redes se ven obligadas a evolucionar hacia la automatización, puesto que esta facilita la conexión a la nube y servicios mediante API, además de aumentar la eficiencia de la red gracias a la programabilidad de esta con la integración de la inteligencia artificial, los beneficios que ofrece la automatización de la red en la gestión y/o configuración son diversos.

2.3.1 Conceptos base para la automatización en redes

Para entender la automatización de redes de telecomunicaciones es necesario comprender conceptos claves que mencionaremos con frecuencia en el presente trabajo.

La integración de la inteligencia artificial y el aprendizaje automático permite que la automatización de la red analice metadatos que junto con la programabilidad de esta proporcione análisis predictivos de su comportamiento, con el objetivo de realizar correcciones de los problemas presentados a tiempo o antes de que ocurran de manera autónoma y eficiente.

2.3.1.1 Inteligencia artificial

La inteligencia artificial (IA) desarrolla un papel importante en la automatización de la redes alámbricas e inalámbricas de telecomunicaciones, puesto que esta simula y es equivalente a un experto humano en las tareas y procesos de la red, solucionando la complejidad que las redes representan al acelerar su crecimiento [48].

2.3.1.1.1 Aprendizaje automático

El aprendizaje automático o Machine Learning (ML) es la base por la cual la IA puede funcionar exitosamente, gracias al uso de algoritmos que utiliza ML para analizar el comportamiento de la red con el propósito de aprender y obtener predicciones de esta, lo cual facilita la pronta respuesta a problemas con la red sin la necesidad de una persona que este monitoreando ni de acciones manuales que solventen el problema presentado [49].

Ejemplo del aprendizaje automático lo notamos todos los días con nuestros smartphones, como noticias y publicidad personalizada, el aprendizaje automático analiza el comportamiento del usuario mediante la interacción con el smartphone, de esta manera se recopila los datos de perfil para ofrecer publicidad, aplicaciones o información orientada específicamente para el usuario.

2.3.1.1.2 Aprendizaje profundo

El aprendizaje profundo o Deep Learning (DL) es la evolución del aprendizaje automático con modelos más complejos, DL utiliza redes neuronales que permite alcanzar mayores niveles de abstracción de la información. Para comprender de mejor manera este concepto se realiza la analogía con las vivencias y experiencia que tiene una persona, esto le permite aprender y de cierta manera ser un experto en algún área específica, el aprendizaje profundo permite a la inteligencia artificial acceder a una gran cantidad de datos que le permiten aprender, cuanta más información disponga DL para su aprendizaje mayor será

la capacidad de rendimiento de la red, es decir, la IA obtendrá la suficiente experiencia para optimizar y aumentar la eficiencia de la red de acuerdo con datos analizados [50].

Un ejemplo práctico de acuerdo con la evolución tecnológica del aprendizaje profundo es en la robótica, puesto que los robots pueden aprender de su entorno lo que aumenta las funciones del robot, como desplazarse evitando obstáculos, realizar tareas que requieran delicadeza en la fuerza o, al contrario.

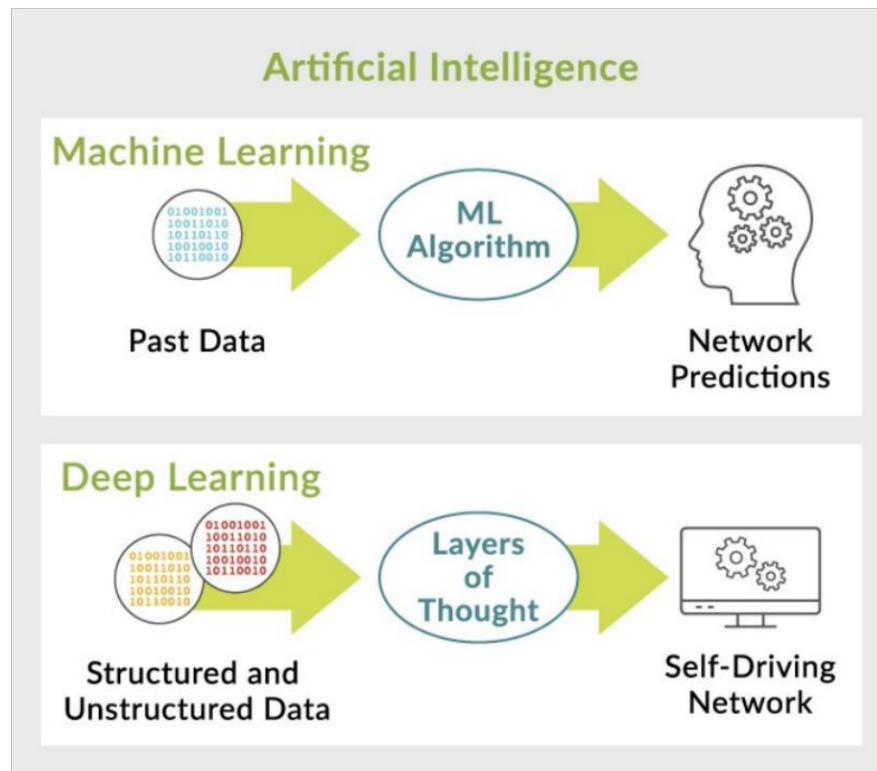


Figura 2.16. Funcionamiento de la IA con ML y con DL. (Fuente: [49])

2.3.1.2 Redes Definidas por Software

La red definida por software (Software Defined Networking - SDN) es una tecnología que permite el control del tráfico de la red y comunicación de la infraestructura dentro de la red a nivel de hardware mediante controladores basado en software o APIs con el objetivo de centralizar la gestión, configuración y programabilidad de la red. De esta forma evoluciona la manera en que se gestiona la red, obviando procesos manuales e individuales a los dispositivos de hardware dentro de la red como enrutadores y conmutadores.

La implementación de las redes SDN permite que exista mayor eficiencia en el control del tráfico en topologías complejas y con gran cantidad de dispositivos con la centralización de

la gestión y configuración, en contra parte al realizarlo de forma tradicional como desventaja se tiene la administración de los dispositivos de manera manual e individualmente [51].

La principal diferencia entre las redes SDN y redes tradicionales es la infraestructura, es decir, una red SDN está basada en software, por otro lado, la red tradicional se encuentra basada en hardware lo que lo limita, al estar basado en software permite que la red sea escalable, flexible, configuración centralizada y aumento de capacidad de una forma rápida y sencilla, sin la necesidad de cambiar el hardware de la red como sucede con la red tradicional.

Las ventajas que ofrece las redes SDN se detallan a continuación [53]:

- Seguridad: al estar centralizada la red permite la visibilidad de todos los dispositivos que componen la red, por lo que, al ser infectado un dispositivo es posible colocarlo en cuarentena para que no infecte el resto de la red.
- Infraestructura: al estar una red definida por software permite que la infraestructura de esta se modifique en tiempo real, direccionando recursos virtuales para optimización de la red o prioridad de aplicaciones.
- Control: como se ha mencionado la SDN permite centralizar la gestión y configuración de la red, por lo que el control de la red se encuentra localizado en un punto, optimizando los recursos y tiempo de configuración.
- Flexibilidad: la red SDN admite protocolos de código abierto con los cuales se puede comunicar con diferentes tipos de dispositivos de hardware gracias a un controlador central.
- Costos: los costos operativos de las redes definidas por software son mucho menores a las redes tradicionales, puesto que, no necesita de dispositivos de hardware orientados a un solo uso ya que la SDN funciona sobre servidores de uso comercial.

2.3.1.2.1 Programabilidad de la red

Las redes SDN puede llevarse a cabo con softwares especializados para la automatización o mediante API, en esta última es necesaria la programabilidad de la red, ya que para la utilización de las APIs pueden usarse lenguajes de programación de alto o bajo nivel con

el objetivo de automatizar tareas o gestión de la red. Para que una red sea programable debe contar con herramientas que permitan lograr este propósito, más adelante se definirá este concepto para ampliar su comprensión.

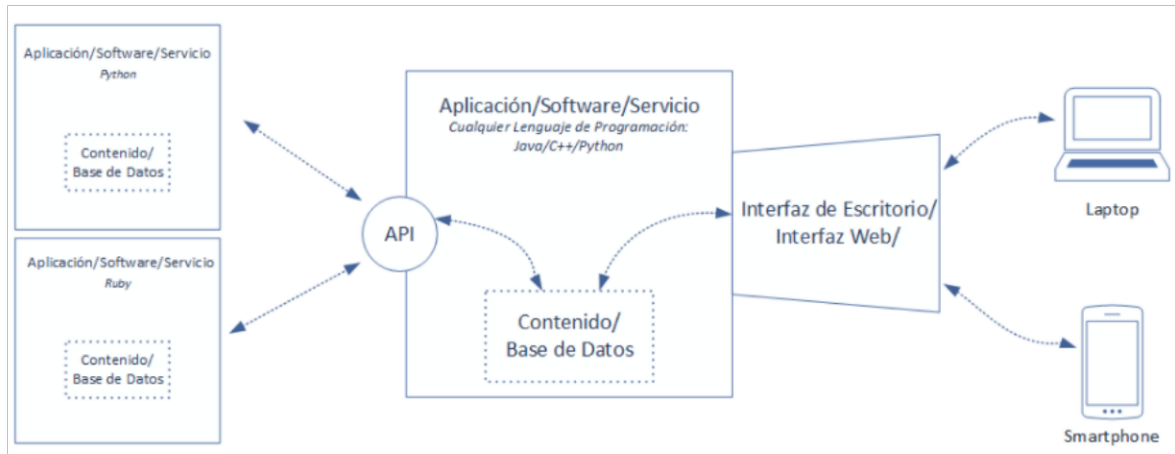


Figura 2.17. Programabilidad de la red mediante API. (Fuente: [54])

2.3.1.3 Virtualización de las funciones de la red

La virtualización de las funciones de la red o NFV (Network Functions Virtualization) es la posibilidad de virtualizar funciones o servicios de la red, tales como firewalls, balanceadores de carga y enrutadores que en redes tradicionales son dispositivos de hardware propietario. La NFV permite empaquetar estos servicios en una o más máquinas virtuales en un solo hardware (servidores) [55].

La integración de las funciones de la red en un solo hardware permite una mejora en la adaptación de la red y agilidad de las funciones, puesto que, de acuerdo con el crecimiento de la red pueden asignarse máquinas virtuales a la red sin la necesidad de requerir recursos de hardware adicionales que involucran costos, tiempo y adaptación a la red ya existente.

La utilización de NFV en las redes de telecomunicaciones permite ejecutar varias funciones de la red en un solo hardware estándar (servidor), sin la necesidad de dispositivos con elevadas prestaciones a nivel de hardware y que son usados solamente para una función de la red. Como beneficio de concentrar las funciones de la red mediante máquinas virtuales en uno o varios servidores se reducen costos generales y se agilizan las funciones de la red.

2.3.2 Automatización de las redes de telecomunicaciones

Dentro de la red pueden llegar a automatizarse varios elementos y con ello varias formas con que puede automatizarse la red, mediante programación o productos de software de automatización también conocidas como herramientas de software. El funcionamiento que tienen es distinto entre estos dos, además de su complejidad [56].

Al usar la programabilidad de la red es posible usar lenguajes de programación que den la facilidad de automatizar tareas de la red sin muchas complicaciones, usando scripts por ejemplo que puedan ser ejecutados varias veces por el administrador, cada script contiene dentro de sí una lista de comandos a ejecutarse, dado es el caso de la configuración de dispositivos de la red. También dentro de esta manera de automatizar la red entran los códigos abiertos como Python o lenguajes heredados como Perl, pero debido a la complejidad de este último la utilización de lenguajes de programación de bajo o alto nivel se ha hecho popular en la automatización de redes de telecomunicaciones.

La utilización de productos de software de automatización permite consolidar varias tareas de la red desde una interfaz intuitiva, fácil de manejar y sin necesidad de conocimientos de lenguajes de programación, sin embargo, existen herramientas de software que permiten programar la red de acuerdo con las necesidades de la red. Esto gracias a la utilización de APIs integradas dentro del software de automatización y la habilitación de estas en los dispositivos de hardware de la red. Las herramientas de automatización son rápidas de integrar a cualquier red de telecomunicaciones.

2.3.3 Ventajas y desventajas de la automatización de la red

Automatizar cualquier red de telecomunicaciones ofrece beneficios tanto económicos como de ingeniería, obteniendo más ventajas que desventajas su implementación.

2.3.3.1 Ventajas

- Aumento de la eficiencia de la red.
- Aumento de la disponibilidad y confiabilidad de la red.
- La red no presenta incoherencias.
- Disminución de errores humanos.

- Disminución de tiempo en el despliegue de la red al permitir configuración masiva de los dispositivos de la red de forma remota.
- Reducción de costos operativos y de mantenimiento.
- Concentración de los recursos de la red.
- Estabilidad de la red.
- Permite que las redes sean escalables y flexibles.
- Simplifica tareas y procesos rutinarios realizados por el administrador de la red.

2.3.3.2 Desventajas

- Redes de telecomunicaciones que debido a su hardware lograr su automatización es compleja, tales como la WSN.
- Inversión inicial alta de acuerdo con el tipo de red a automatizar.

2.3.4 Automatización de operaciones de la red

Dentro de las operaciones de la red que pueden llegar a automatizarse están, la configuración de red, balanceo de la carga y conmutación.

Al realizarse la configuración de la red manualmente se introduce la probabilidad de error humano, lo cual provoca incoherencias dentro de la red e inestabilidad de servicios que requieren disponibilidad de más del 99.999%, por este motivo la automatización soluciona este tipo de problemas y al momento de suceder reduce el tiempo al mínimo de resolución.

El manejo manual del equilibrio de la carga disminuye el rendimiento de la red junto con el aumento del tiempo de conmutación al tratar de balancear la carga, para solucionar este problema se utiliza la automatización con IA que permitirá buscar la mejor ruta de conmutación para equilibrar la carga incrementando de esta forma el rendimiento de la red con rapidez.

2.3.5 Definición de Herramientas para automatización de redes de telecomunicaciones

Las herramientas para la automatización de redes son plataformas, programas o en su forma más básica, librerías de un lenguaje de programación de bajo o alto nivel que permiten la automatización de la red. Existen varias herramientas para automatización de redes de telecomunicaciones, estas pueden variar según las opciones de automatización que dispone, por fabricante, lenguaje de programación usado, pagadas o gratuitas [57].

Dado el desarrollo de las redes y el crecimiento acelerado que se tiene, el uso de estas herramientas facilita los procesos y tareas de gestión de toda la red que a su vez aumentan su eficiencia y disminuyen los costos operativos. Estas herramientas deben permitir la gestión y automatización de la red, puesto que deben ser capaces de detectar los dispositivos de la red para gestión de configuración, resolución de problemas presentados de forma autónoma, maximizar eficiencia de funcionalidades y reducción de errores humanos dentro de la red.

El propósito de las herramientas es facilitar que la automatización de la red sea posible de la manera más simple y sencilla con la capacidad de automatizar todos o gran parte de los procesos de gestión, configuración y monitoreo de la red, existen fabricantes como Cisco, Red Hat, Juniper, entre otros que ofrecen softwares que están diseñados para automatizar las redes de telecomunicaciones con infraestructura propietaria [58], es decir que el software de automatización ofrecido por un fabricante A no será compatible con la arquitectura de red de un fabricante B, al ocurrir esto surgen las plataformas pagadas o gratuitas donde se implementa la automatización mediante scripting, si el firmware de los dispositivos de la red admite scripting será posible el uso de estas plataformas para la automatización de la red con programación en Python [59].

Al no existir un estándar de la automatización de las redes de telecomunicaciones cada fabricante ofrece la manera que mejor le parezca para optimizar la automatización de la red y a la necesidad de sus dispositivos, a pesar de esto, cumplen con lo que debe tener una herramienta para automatizar la red.

2.3.6 Automatización de redes inalámbricas de sensores

La automatización de redes inalámbricas de sensores permite la automatización de procesos y tareas relacionadas con la gestión de las redes, esto implica la configuración masiva de nodos, estado de la red, alertas de sucesos y comunicación con APIs que

permitan la integración de servicios web para el tratamiento y análisis de los datos de los sensores. Cabe mencionar que la gestión de redes es un grado básico de la automatización de esta.

La automatización es posible gracias a la programabilidad de la red, los lenguajes de programación con lo que se puede desarrollar la automatización depende de los fabricantes, sin embargo, con el auge del lenguaje Python por su popularidad, facilidad de uso y una curva de aprendizaje corta, hace que este lenguaje sea usado para implementar la automatización de la red.

Las ventajas que trae la automatización principalmente en redes WSN es la disminución de costos operativos, errores humanos e incoherencias de la red, en su contraparte aumenta la eficiencia de la red, tiempos de respuesta y en casos avanzados con la integración de la nube por medio de APIs es posible implementar la inteligencia artificial con el propósito que la red pueda ser autónoma en la toma de decisiones dentro de la red.

La integración de las API en las redes inalámbricas de sensores permite la posibilidad de automatización de la WSN, ya que al poder integrar la red a la web admite el acceso remoto desde cualquier parte del mundo para gestionarla y configurarla remotamente, es así como el fabricante Digi International permite la automatización de la red inalámbrica de sensores con sus módulos XBee.

Sin embargo, en el campo de redes inalámbricas de sensores la evolución de estas hacia la automatización aún no está claramente definida, parte de la recesión es la popularidad y aplicación que tienen las demás redes de telecomunicaciones, la WSN a pesar de haber sido desarrolladas hace varias décadas atrás y con la implementación que tiene en la actualidad, aún no se ha llevado a cabo la automatización de esta red con la misma velocidad que las demás redes de telecomunicaciones.

El principal desafío que afronta la posibilidad de automatizar la WSN son las características limitantes del hardware, como consumo energético y capacidad de procesamiento, sin embargo, para llevar a cabo la automatización de este tipo de redes es necesario la integración de SDN, sin embargo, si bien existen varios estudios en los cuales se estudia la integración de SDN a redes inalámbricas en el transcurso de las últimas décadas [60], no se ha dado la suficiente importancia a la integración en redes WSN, esto puede deberse a la complejidad que presenta esta integración, ya sea a nivel de hardware o a nivel de software.

Al revisar las especificaciones técnicas de los nodos sensores se observa que se encuentra limitada por el procesamiento que tiene y el consumo limitado de energía que dispone, principalmente la integración de SDN a la WSN tiene como principal desafío mejorar la eficiencia energética adaptándose a las características de los nodos sensores, puesto que a mayor capacidad de procesamiento que tengan estos mayor será el consumo de energía y limitando a otros procesos dentro del nodo, dando como consecuencia fallas en la comunicación o medición de los parámetros físicos. A nivel de software es necesario la implementación de protocolos que permitan la integración de SDN que mejoren la eficiencia energética y no requieran de un consumo elevado de las prestaciones de procesamiento disponible del nodo.

A pesar de las limitantes que presenta la integración de SDN a la WSN lograr su optimización y perfección es difícil, ya que las redes definidas por software están orientadas a otros tipos de redes de telecomunicaciones que no presentan limitantes como las descritas anteriormente, sin embargo, es posible la integración realizando modificaciones en los protocolos, en la arquitectura de las capas usadas en la WSN y mejoramiento del hardware de los nodos sensores con el propósito de lograr la integración eficaz de SDN en la WSN.

3. METODOLOGÍA

La metodología usada para el presente trabajo de titulación es de carácter documental y exploratorio con el objetivo de profundizar en el estudio de la programabilidad de redes de sensores para su automatización, además de analizar el estado actual de la automatización aplicada a redes inalámbricas de sensores con un lenguaje de programación fácil y sencillo como lo es Python.

3.1 Investigación documental

Para cumplir con el objetivo general y los objetivos específicos del presente trabajo de titulación se escogió la metodología documental, lo que lleva a establecer su definición. La investigación documental es un proceso por el cual la información de un tema en específico es buscada, recolectada y organizada para un posterior análisis, reflexión e interpretación de esta, realizándose así la construcción del conocimiento a partir de la documentación disponible.

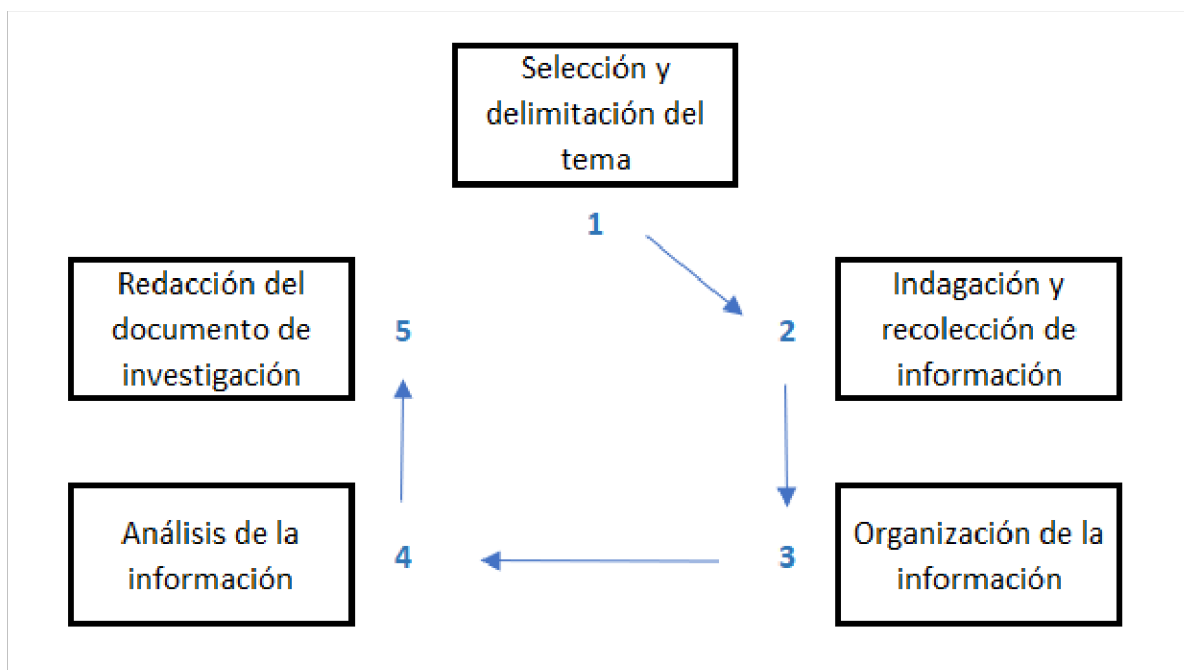


Figura 3.1. Fases de la investigación documental. (Fuente: [61])

Por consiguiente, el método de investigación documental en general es el proceso en el cual se establece una investigación exhaustiva y minuciosa sobre un tema en específico,

permitiendo la recolección de información relevante de varias fuentes bibliográficas con el propósito de concentrar y sintetizar toda la información revisada.

3.1.1 Tipo de enfoque

Para la investigación documental de este trabajo se establece un enfoque cualitativo, que será de ayuda para el proceso de investigación con un criterio selectivo de las fuentes de información consultadas.

Debido a que para el desarrollo del presente trabajo no se tienen variables medibles, sino más bien, se tiene información cualitativa con lo cual se pretende cumplir con el objetivo general y los objetivos específicos del apartado 1.2 del documento fue el motivo por lo que se escogió el tipo de enfoque cualitativo.

3.1.2 Tipo de trabajo

El nivel de investigación que tiene el trabajo es de tipo exploratorio, ya que, por el momento no se tiene información consolidada que sea suficiente sobre la configuración, funcionamiento y mantenimiento para la automatización de redes inalámbricas de sensores con el lenguaje de programación Python, ya sea esta información de origen físico o electrónico.

Adicional, se pretende realizar un análisis del estado actual y profundizar en la investigación de posibles herramientas que permitan la automatización y gestión de redes WSN disponibles en el mercado de las telecomunicaciones o a su vez que permitan ser desarrollados por la comunidad como el caso de código abierto de Python.

3.1.3 Técnica de recolección de información

La recolección de información es el proceso por el cual se concentra la información disponible y necesaria para estudiar un tema determinado, además, apoya a la discriminación de la información confiable y no confiable, permitiendo así obtener información veraz al momento de establecer las conclusiones del trabajo.

Al usarse un tipo de enfoque cualitativo en el presente trabajo, se da la pauta a utilizar la técnica de análisis documental de la información, lo cual permite analizar y entender la información recolectada para tener un resultado favorable y óptimo.

3.1.4 Instrumentos de recolección de información

El instrumento de recolección de información utilizado en el presente documento es el estudio y análisis de contenido documental. En base a la documentación revisada de varios tipos de documentos y de diferentes fuentes se construye la información presentada en este documento, analizando minuciosamente con el objetivo de no ser redundante ni proporcionar información errada e inconsistente.

4. AUTOMATIZACIÓN DE LA WSN

4.1 Situación actual de la automatización de la WSN

En la actualidad la mayor parte del mercado de las redes inalámbricas de sensores se encuentra entre la industria y consumidores civiles, esta última, es una razón por la cual no ha evolucionado la automatización de la WSN, ya que, al ser utilizadas en su mayoría para aplicaciones de hogar o control de accesos, no existe la necesidad del cliente en automatizar la gestión de la WSN.

De acuerdo con la investigación se evidencia que los nodos al estar orientados para aplicaciones de IoT en hogares no presentan los mismos problemas que existen en ambientes hostiles, por ejemplo, en la industria, por tal razón, los nodos para el segmento de mercado hogar tienen una probabilidad mucho menor a fallar o presentar errores en la comunicación, no existe la necesidad de solventar de forma rápida y autónoma problemas presentados a lo largo de su vida útil.

Caso contrario a la realidad de la industria, donde el entorno que rodea al nodo de una red inalámbrica de sensores es hostil y muchas veces con interferencias que perturban la señal emitida por el nodo transmisor al medio aire en la comunicación entre los nodos. La industria para todos sus procesos necesita de un alto nivel de precisión y eficiencia que garantice la producción sin pérdidas de recursos monetarios o materia prima.

Por tal motivo, desde el comienzo de la implementación de redes WSN en la industria y hasta hoy, diferentes fabricantes han creado alianzas o trabajan de forma independiente con el objetivo de desarrollar esta tecnología, puesto que tienen varias aplicaciones en todos los campos de la industria y de esta forma implementarla para optimizar sus procesos industriales.

En el presente, el fruto del desarrollo de las industrias en alianzas o independientes, son nuevos estándares de comunicación inalámbrica con varias mejoras y beneficios para determinadas aplicaciones sobre todo y principalmente en la industrial, algunos de estos estándares son licenciados y otros abiertos a la comunidad para su desarrollo y avance para mejoras de los protocolos.

Ahora bien, los estándares creados por la industria varían según su aplicación, se tienen protocolos que permiten conectar la WSN a una red TCP/IP sin necesidad de pasar por un gateway que realicen la transformación y asignación de direcciones IP, haciendo más eficiente de esta manera la red inalámbrica de sensores, ya que el mismo protocolo de la

WSN lo hace con la información proporcionada por la red TCP/IP, tal es el caso de los estándares 6LoWPAN, Thread, Wi-Fi y Wi-SUN.

Por otro lado, las redes que no pueden realizar la conexión directa a la red TCP/IP hacen uso de un gateway que realiza el direccionamiento IP. La ventaja de poder conectar una WSN a una red TCP/IP es la capacidad de utilizar APIs y por medio de esta conectarlos a la nube, esta tecnología que hoy por hoy ofrece ventajas en la automatización y a futuro promete llegar a ser completamente una red autónoma gracias a la integración de la inteligencia artificial a la WSN.

Con el auge de los servicios interconectados en línea que existe en la actualidad las API son ampliamente usadas para este propósito con lenguajes de programación como Python, JSON entre otras y es el medio por el cual se interconectan diferentes infraestructuras para obtener información que luego quien solicita la información de manera remota pueda procesarla para brindar un servicio, todo esto salvaguardando la integridad y seguridad de la información.

La automatización de la WSN actualmente aún se encuentra en un estado emergente, puesto que, no se ha encontrado artículos, papers o algún documento por parte de universidades o fabricantes que demuestre un adelanto de este en la WSN. Sin embargo, se encuentran colocadas las bases para desarrollar la automatización en este tipo de redes, las API al permitir acceder a servicios de la nube permite la posibilidad de integrar inteligencia artificial para que la red pueda evolucionar por sí misma.

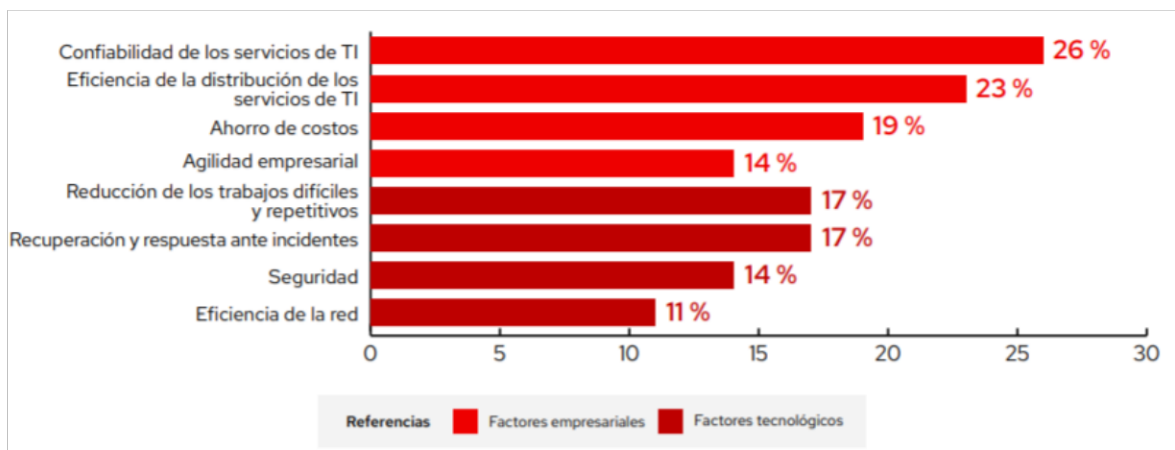


Figura 4.1. Factores que impulsan actualmente al desarrollo de la automatización de la red. (Fuente: [63])

A pesar de que la automatización de la WSN no está completamente desarrollada y aún está en proceso de mejorarse, se ha encontrado en la documentación revisada que el fabricante Digi International ha logrado implementar la automatización de este tipo de redes con tecnologías propietarias y su módulo XBee que al incluir el microcontrolador permite la programabilidad de los nodos sensores de la red, además, el uso de Digi Remote Manager permite la automatización de la WSN integrando este con la nube mediante APIs, logrando así colocar las bases para el desarrollo de inteligencia artificial en la WSN para un análisis predictivo del comportamiento de la red y solucionar problemas presentados en los nodos en tiempo real sin perder confiabilidad.

Digi frente a otros fabricantes ha logrado acercarse mucho más a la automatización de la WSN, sin embargo, igualar el estado actual de la automatización de las redes de telecomunicaciones es complejo, debido a que está muy desarrollado y adelantado, gracias a la implementación de la inteligencia artificial y aprendizaje automático de la red, por otro lado, el mundo cambiante de la evolución tecnológica e industria, a futuro impulsará aún más el desarrollo de la automatización de las redes inalámbricas de sensores solucionando los presentes desafíos que tienen este tipo de redes para ser automatizadas en su totalidad, logrando así alcanzar o a su vez superar la automatización de las redes de telecomunicaciones.

Actualmente, los principales problemas que se tiene con respecto a la automatización de las redes inalámbricas de sensores son el consumo de energía, la capacidad de procesamiento, protocolos que mejoren la eficiencia de los recursos de hardware para minimizar el impacto energético, integración de SDN con el objetivo de centralizar la red para el control, gestión y monitoreo de esta [86]. Como puede observarse, existen varios desafíos que tiene la WSN para lograr una completa automatización de esta, sin embargo, la documentación revisada en el presente trabajo ofrece soluciones viables para lograrlo, de las cuales se las revisarán en capítulo a continuación.

4.2 Estudios relacionados con la automatización de WSN

En el presente trabajo se han encontrado varios estudios actualizados relacionados con la automatización de una red de telecomunicaciones, por lo que a continuación se abordarán estudios relacionados a la automatización de redes inalámbricas de sensores lo más actual posible que se ha encontrado.

En los últimos años se han presentado estudios relacionados con la automatización de la red implementándola en WSN definidas por software como detallan los estudios Software Defined Networking for Improved Wireless Sensor Network Management: A Survey [64] y Software defined wireless sensor networks application opportunities for efficient network management: A survey [65] ambos publicados en el año 2017 en los cuales se detallan la importancia de la revolución de la WSN y WSNAN, debido al crecimiento vertiginoso y popularidad que ha tenido la tecnología del internet de las cosas (IoT). Los estudios analizan la integración de SDN (Software Defined Networking) como una solución que promete brindar una gestión flexible de la WSN, siendo posible el control centralizado de toda la WSN, sin embargo, ambos estudios son puntos de partida para comprender una WSN que esté basada en SDN planteándose como un desafío a futuro lograr esta integración.

La integración de SDN a las WSN implica problemas relacionados con la eficiencia energética, como se recordara por los capítulos anteriores del presente trabajo, las redes inalámbricas de sensores se caracterizan por su bajo consumo de potencia lo que les permite tener ciclos de vida útil de la batería en el orden de las decenas de años dependiendo del estándar usado, al centralizar la WSN con la ayuda de SDN aumenta el consumo energético debido al procesamiento y capacidad de cómputo de los nodos, siendo este una limitante para la implementar SDN, sin embargo, en el año 2015 se propuso un prototipo orientado a la supervisión energética de la WSN con implementación SDN [66] para solventar problemas relacionados a la energía, radio, cómputo y flujo de datos, obteniendo resultados prometedores.

A medida que la WSN aumenta la cantidad de nodos, surgen inconvenientes en la seguridad, por lo tanto, la información transmitida entre los nodos es vulnerable a interceptación y/o modificación, el estudio: A Comprehensive Survey on Various key Management Schemes in WSN [67] publicado en el año 2018 se menciona los diferentes mecanismos de gestión de claves, en este estudio es un documento de apoyo para gestionar la seguridad e integrándolo con programación en Python para automatizar esta gestión en la WSN.

Las redes definidas por software nacieron de la necesidad de reducir la complejidad, administración y gestión para redes inalámbricas, sin embargo, el principal problema que tiene para integrarse a la WSN es el alto consumo de energía y computo, puesto que, las redes inalámbricas de sensores no ofrecen las capacidades a nivel de hardware para lograr una integración totalmente eficiente de SDN. Siendo esta una razón por la que se buscó otro camino para la automatización de la WSN, lo que se propuso automatizar la red

inalámbrica de sensores solamente con tareas necesarias y básicas que no involucre el consumo elevado de energía y procesamiento, con esto se desarrolló la programabilidad de la red con Python en los nodos, logrando automatizar tareas sencillas de la red, como gestión de la configuración de la red, gestión de alertas de sucesos, actualización masiva de los nodos del firmware, acceso remoto al nodo, habilitación y des habilitación de los nodos, mantenimiento programado de la red, entre otros procesos.

De acuerdo con el estudio Software- Defined Networking for Dynamic Control of Mobile Industrial Wireless Sensor Networks [68] realizado en el 2018 propone la integración de SDN a la WSN con nodos sensores móviles en áreas industriales donde el ambiente es susceptible a pérdidas de paquetes en la comunicación, estos nodos móviles se encuentran en constante cambio de posición dentro del entorno de medición en donde esta implementada la WSN dando lugar a pérdida de información en la comunicación entre los nodos. Las simulaciones realizadas de un sistema basado en SDN para lograr la adaptación a la red inalámbrica de sensores con el menor impacto negativo posible referente a la pérdida de paquetes y latencia demostró que es posible centralizar la red para su gestión, sin embargo, al solventar los problemas presentados por la comunicación se evidencia que aumento el consumo de energía, por tal motivo se propone para trabajos futuros encontrar el equilibrio entre la integración de SDN a la WSN con un consumo eficiente de energía.

Como se ha revisado en este subcapítulo la mayor cantidad de estudios relacionados con la automatización de las redes inalámbricas de sensores presentan un problema en común, el consumo de energía elevado de los nodos. Esto provoca pérdida de paquetes, aumento en la latencia, disminución de la capacidad de procesamiento y del ciclo de vida de la red, a pesar de ello, los estudios analizados muestran el potencial de SDN para la automatización de la red y como principal requerimiento para llevarlo a cabo, sin embargo, la utilización de Python en nodos que admitan su programación como es el caso de los módulos XBee llevan a que la automatización de la WSN sean posibles a corto plazo, es necesario recalcar que la automatización realizada a los módulos XBee son de procesos y tareas relativamente simples pero muy útiles, por otro lado, el ecosistema XBee permite ampliar la capacidad de automatización de la red con herramientas de automatización como lo es Digi Remote Manager.

Para finalizar, los estudios analizados referentes a la automatización de la WSN con la integración de SDN aún no son totalmente concluyentes para realizar esta integración con óptimos resultados, actualmente existen varios desafíos por solucionar referentes a la

calidad de servicio, consumo energético, seguridad, latencia y pérdida de paquetes con el propósito de lograr el equilibrio y eficiencia de la red en todo sentido.

4.3 Herramientas para automatización y gestión de WSN

Debido a que la automatización aún está emergiendo se encontraron pocas herramientas para la automatización de redes inalámbricas de sensores, sin embargo, se revisará las herramientas disponibles. Las herramientas tienen como objetivo ayudar al diseñador de la WSN a configurar, monitorear y gestionar la red con el objetivo de facilitar los procesos manuales que son sistemáticos.

4.3.1 StreetLight Control

Es un software de alumbrado público orientado para aplicación de ciudades inteligentes, si bien no es un software de automatización con funciones completas de automatización de la red, permite automatizar algunos parámetros y gestionarlos. Puede verse más información en [69].

Ofrece un análisis en base a los datos recolectados por los nodos de la WSN, gracias a un nivel alto de inteligencia artificial, además, permite generar reportes con el propósito de mejorar el rendimiento de la red. StreetLight Control tiene herramientas de gestión de la red como monitoreo, informes de fallas y estado de la red en tiempo real, configuración masiva de los nodos, actualización del firmware de todos los nodos de la red, además, permite la programabilidad de procesos de mantenimiento de la red.

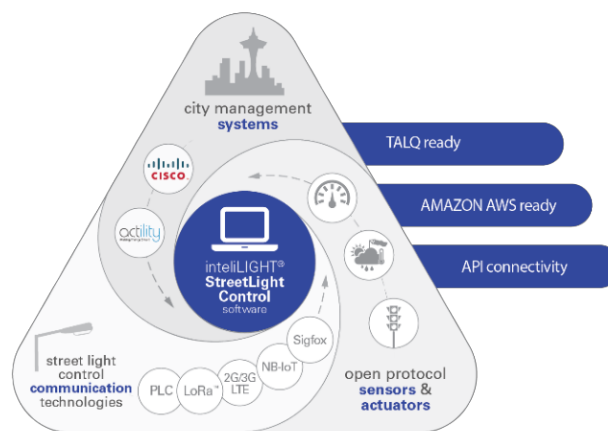


Figura 4.2. Características de StreetLight Control. (Fuente: [69])

4.3.2 Ginjer

Es una plataforma de administración de dispositivos de IoT, esta herramienta permite visualizar el estado de los sensores de la WSN remotamente en tiempo real, los datos obtenidos por la WSN se envían a la plataforma para análisis y generación de informes. La aplicación permite acceso y configuraciones básicas de forma remota gracias a las API utilizadas en el software. Puede verse más información en [70].

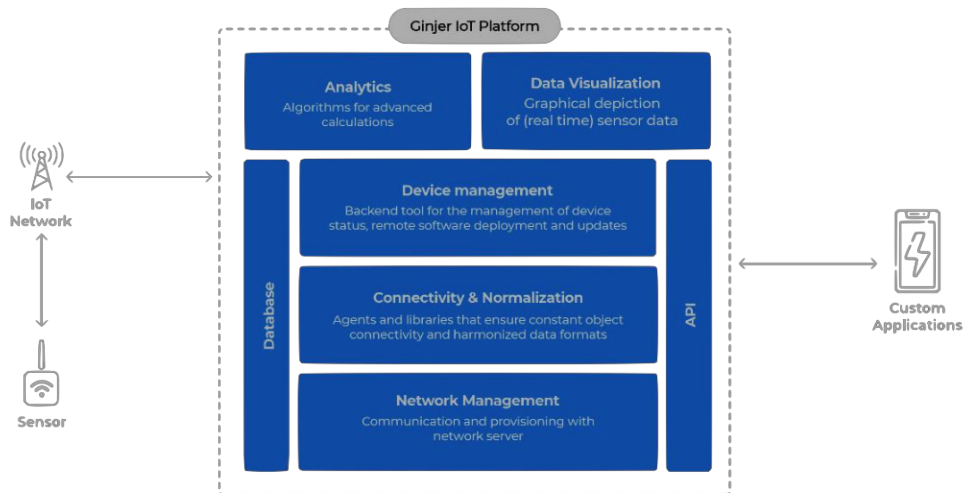


Figura 4.3. Diagrama de bloques de la plataforma Ginjer. (Fuente: [70])

4.3.3 XCTU

Esta herramienta es un software desarrollado por Digi que permite la configuración sencilla, rápida y sistemática de módulos de RF de Digi, además permite realizar pruebas de la configuración y medidas de los parámetros de transmisión en tiempo real.

Las características que ofrece XCTU son:

- Configuración y gestión sencilla de módulos XBee conectado por el puerto serie o de forma remota, es decir conectado inalámbricamente.
- Actualización del firmware del módulo.
- Consola de comunicación con el módulo mediante comandos AT o comandos API.
- Permite guardar sesiones de consola y abrirlas en otro PC con XCTU instalado.
- Incluye terminal de MicroPython.

- Realiza pruebas de alcance entre dos módulos que estén en la misma red.
- Explorador de firmware.
- Construcción de tramas API de forma fácil y rápida.
- Decodificación de tramas API para observar información.
- Compatible con Windows, MacOS y Linux.
- Analizador de espectro en bandas de frecuencia 2.4 GHz y 900 MHz.
- Herramientas de rendimiento para los módulos XBee.

El terminal de MicroPython permite la programación de scripts y automatizar tareas de gestión de la red en los módulos Xbee, además, se ofrece tutoriales y documentación con información sobre la codificación en Python en los recursos de Digi o GitHub.



Figura 4.4. Interfaz gráfica de XCTU. (Fuente: [71])

4.3.4 Digi Remote Manager

Esta herramienta es la más completa que dispone Digi, puesto que, ofrece gestión, configuración, despliegue de los nodos y automatización de la WSN. Como principales características del software se listan a continuación:

- Configuración remota de un grupo o todos los nodos de la red.
- Comprobación de configuración de los nodos con el propósito de que no hayan sido reconfigurados arbitrariamente o configuración no intencionada.
- Automatización de tareas de gestión y actualización de un grupo o todos los nodos, programadas eventual o periódicamente.
- Programar tareas de creación de informes con métricas y estado de la red inalámbrica de los sensores.
- Establecer alertas para de métricas variantes y/o funcionamiento correcto de los nodos.
- Conjunto de APIs para la integración de aplicaciones, análisis, recolección y visualización de datos.
- Programación de funcionalidades para los dispositivos de borde en la red con el lenguaje Python, de esta manera realizarlo para todos los dispositivos de borde o un grupo seleccionado.
- Integra seguridad y privacidad de datos.
- Diagnostico en tiempo real.
- Acceso a dispositivos que se encuentren fuera de banda.



Figura 4.5. Aplicaciones de utilización para Digi Remote Manager. (Fuente: [72])

Hasta el momento Digi Remote Manager ha sido el software más completo para gestión de una red inalámbrica de sensores, además, ofrece la herramienta de automatización de la red con APIs y scripts de Python. Esto siendo posible de forma remota desde un computador, tableta o smartphone.

4.4 Automatización de redes inalámbricas de sensores con Python

A pesar de la búsqueda de información relacionada con las redes inalámbricas de sensores que implementen automatización basada en Python, las implementaciones encontradas no son suficientes, sin embargo, lo que se ha encontrado tiene relación con la automatización de redes inalámbricas de sensores basadas en la arquitectura ZigBee conocida como XBee las cuales están desarrollados, adaptados y cuentan con herramientas para esta implementación. Lo que no sucede con otros estándares inalámbricos. Encontrándose casos de automatización, los cuales se describirán en este capítulo.

De acuerdo a la documentación revisada, la automatización de una red inalámbrica de sensores admite la programación de lenguajes de bajo o alto nivel dependiendo del fabricante y el estándar con el que funciona, el objetivo del presente trabajo es el estudio para entender la automatización de una WSN con el lenguaje de programación de alto nivel como lo es Python, por lo que, en este capítulo se analizará la documentación disponible sobre el tema y abordaremos la automatización para redes de módulos XBee 3.0, puesto que, al incluir un microcontrolador dentro del módulo no es necesario un microprocesador externo, además, permite programación con MicroPython el cual es una implementación con funciones reducidas del lenguaje Python 3 para microcontroladores.

4.4.1 Ventajas y desventajas de la WSN automatizadas con Python

4.4.1.1 Ventajas

Automatizar la WSN ofrece ventajas desde dos diferentes verticales, el primer vertical es desde el punto de ingeniería.

- Disminución del tiempo de configuración, al utilizar funciones disponibles en las librerías puesto que, basta con llamar a la función para realizar alguna tarea de configuración de un nodo sensor o un grupo de nodos sensores.
- Despliegue fácil y rápido de nuevos nodos dentro de la WSN agilizando el proceso para que el nodo comience su operación.
- Disminuye errores e incoherencias dentro de la red por configuración manual de los nodos sensores, además, la programación de scripts con Python permite detectar incoherencias de la red para corregirlas.
- Mantiene la WSN estandarizada y organizada.
- Permite acceso remoto desde cualquier dispositivo electrónico, al estar integrado con APIs es posible realizar la gestión o administración de la red desde un smartphone.
- Aumento de la disponibilidad de la red.
- Recolección automática de la información de la red.
- Al programar con Python es posible integrar varios servicios y aplicaciones disponibles en línea gracias a las APIs.
- Aumenta la eficiencia de la red.
- Utilización de REST (Representational State Transfer) mediante APIs, esto permite interoperabilidad con otras herramientas disponibles en línea.
- Flujo de datos eficiente.
- Mantenimiento predictivo de acuerdo con el análisis de la información de la red, gracias a la integración de inteligencia artificial.

El segundo vertical se encuentra desde el punto de vista de la industria.

- Disminución de costos operativos, puesto que una red automatizada no necesita de un operario para monitorearla 24/7.
- Disminución de tiempos de espera, aumenta el SLA (Service Level Agreement) del servicio prestado por la empresa.

4.4.1.2 Desventajas

A pesar de que la automatización de WSN ofrece importantes y varias ventajas, también se tienen desventajas.

Desde el punto de vista de ingeniería se tiene:

- Aumento del consumo energético, por lo que disminuye el tiempo de vida útil de la batería.
- Probabilidad de sobrecargar el procesamiento por la programación del código, recordemos que una característica de los nodos sensores es su limitada memoria y bajo procesamiento.

Desde el punto de vista de la industria:

- La inversión económica para implementar la automatización de la red y capacitación del personal.

4.5 Casos de automatización procesos en WSN

La programación en Python de la red hace posible la automatización de tareas o procesos, consecuencia de esto se detallarán proyectos implementados en el ecosistema XBee, en los cuales se desarrolló la automatización usando el lenguaje de programación Python.

Dado que la automatización de redes inalámbricas de sensores es relativamente nueva, no se encontró suficiente información de los casos en los cuales se implemente la automatización de la red con Python, sin embargo, se detalla a continuación la información disponible.

4.5.1 Automatización de WSN con Digi Remote Manager

Las redes inalámbricas de sensores tienen como objetivo recolectar mediciones de los parámetros físicos de su entorno, por lo que, el interés de la industria es procesar estos datos para un análisis y posteriormente toma de decisiones.

La manera que se encontró para realizar el envío de la información recolectada es por medio de programación de código en MicroPython, enviando como flujo de datos a cualquier plataforma disponible en la nube para el procesamiento de la información.

En este trabajo se estudia la implementación de Digi Remote Manager para el almacenamiento y procesamiento de la información mediante APIs. Es decir, el proceso a automatizar que se explica en el trabajo es enviar toda la información recolectada por los nodos sensores de la red, ya que para analizar los datos el administrador de la red necesita crear manualmente el flujo de datos hacia un servidor o unidad de procesamiento de información, lo cual detalla en el trabajo como automatizar esta tarea. De acuerdo con la revisión de las herramientas para automatización y gestión de la WSN, conocemos que Digi Remote Manager es la herramienta más completa para llevar a cabo algunos procesos relacionados con la automatización de la red.

En primer lugar, la configuración del nodo sensor se realiza habilitando el modo API para comunicación remota y permitir la programación de MicroPython en el módulo Xbee 3. Posterior a esto se explica en el trabajo [73] cómo deben cargarse las librerías [74] y [75] en el código para acceder a funciones que facilitan la programación sin saturar el procesamiento del microcontrolador.

En el trabajo se muestra el funcionamiento cuando se envían los datos a Digi Remote Manager y como acceder a la información para generación de reportes gráficos y estadísticos, además, el software trabaja con APIs que permiten compartir la información con otras plataformas disponibles en internet en formato XML para procesamiento especializado de dicha información según sea la aplicación.

Con la automatización del flujo de datos, se observa cómo puede lograrse la centralización de la carga de datos recolectados por los nodos de la WSN, llegando a un nivel de automatización de la red inalámbrica de sensores relativamente sencilla, sin embargo, puede ser la base para futuros trabajos de automatización como lo indica el autor del trabajo de esta automatización.

4.5.2 Notificación de eventos de la WSN vía SMS

La aplicación de notificación de eventos en tiempo real surgió de la necesidad de monitorear los datos de temperatura tomados de un sensor en específico sin importar donde se encuentre la persona interesada, de esta forma se plantea el uso de MicroPython para obtener estos valores programando la frecuencia de actualización, día, hora y cantidad de datos enviados vía SMS, en este trabajo el autor propone automatizar la notificación de eventos de acuerdo a ciertas condiciones en la medición de temperatura del

sensor, enviando una alerta vía SMS con la ayuda de programación en MicroPython del módulo XBee 3.0 [76].

Automatizar una tarea del proceso de monitoreo de la red enviando alertas vía SMS, permite que las alertas sean notificadas en tiempo real y en cualquier lugar con cobertura celular, además, se garantiza que sea tomada en cuenta para una pronta acción. Para la implementación de la notificación del evento, el autor propone apoyarse en el ecosistema XBee, usando el módulo XBee 3 celular LTE-M para permitir la conexión a la infraestructura celular y de esta forma enviar la notificación con la información deseada vía SMS.

De acuerdo con el trabajo analizado, antes de configurar y desarrollar la programación del nodo, es necesario realizar la activación del modo API y habilitación de MicroPython. La API permite notificar eventos de forma remota y la conexión con otros nodos dentro de la WSN.

Aunque la automatización revisada de este trabajo es de una tarea de monitoreo de la red inalámbrica de sensores sea básica, permite entender el potencial que puede lograrse con la programación de la WSN con Python, logrando automatizar gradualmente el monitoreo de todos los nodos de la red.

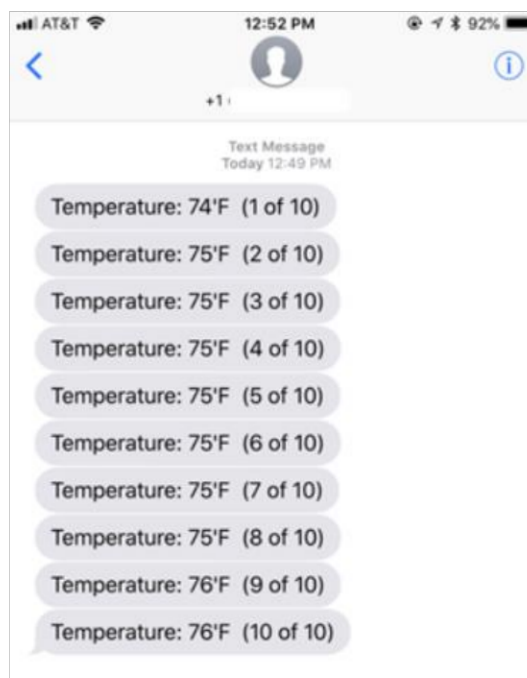


Figura 4.6. Notificación de la temperatura de un nodo sensor vía SMS. (Fuente: [79])

4.6 Desafíos de la automatización de redes inalámbricas de sensores

La automatización de las redes inalámbricas de sensores presenta varios desafíos para su implementación, principalmente la eficiencia energética, al ser una WSN con gran cantidad de nodos aumenta el consumo de energía, por lo que la programación de la red debe solventar este problema debido a los recursos a nivel de hardware limitados de los nodos.

Mantener o mejorar la eficiencia de parámetros de energía, radio, procesamiento y flujos de datos de los nodos al implementar SDN en la WSN es un problema que debe ser analizado a detalle, buscando el equilibrio entre estos parámetros para lograr una óptima integración de SDN en WSN, siendo estos temas para trabajos futuros.

La obsolescencia de equipos es un desafío actual para la automatización de la WSN, es decir, la infraestructura debe estar preparada con los estándares que permiten llevar a cabo la automatización, ya sea a nivel de hardware o software, como hemos visto las redes inalámbricas de sensores están principalmente limitadas por su hardware, por lo que una infraestructura actualizada permite optimizar y adaptar perfectamente la automatización.

La seguridad de la WSN no es lo suficientemente robusta para enfrentar ataques, esto se debe a que entre más segura sea la comunicación más consumo de energía se tendrá, por lo que es un problema que debe tomarse en cuenta para la automatización, al estar centralizada la red se concentra en un punto la seguridad, si llega a ser vulnerado este punto toda la red estará desprotegida.

Actualmente no existe la interoperabilidad con otros protocolos que basen su automatización con otros lenguajes de programación de bajo o alto nivel en las redes inalámbricas de sensores, al superar este desafío la red pueda ser fácilmente escalable y flexible, aprovechando las ventajas que disponen otros lenguajes de programación para la automatización, además, la optimización de protocolos utilizados en la WSN para lograr la eficiencia energética es un desafío colateral de la interoperabilidad.

6. CONCLUSIONES

Aunque la automatización de las redes inalámbricas de sensores ofrece diversas y variadas ventajas tiene limitaciones implementarla. Desde el origen de la WSN se buscó un consumo de energía bajo en los nodos con el objetivo de obtener varios años de vida de la batería, ya que al ser usados en campo que son de difícil acceso se dificultaba cambiar periódicamente la batería para que funcionen. Para solucionar esto se limitaron los recursos del nodo, como memoria, capacidad de procesamiento, seguridad del enlace y parámetros de transmisión. Ahora la integración de la automatización de la red pone en consideración el consumo de energía, siendo la principal limitación que tiene la WSN frente a la automatización junto con el bajo procesamiento de los módulos de los nodos sensores de la red y capacidad de memoria.

Como se revisó en el presente documento la automatización de red puede ser posible con la creación de scripts con un lenguaje de programación de alto nivel como lo es Python, por otro lado, la automatización de redes inalámbricas de sensores las cuales sus nodos no tienen las mejores prestaciones a nivel de hardware limitan su utilización, además, al ser relativamente nueva la automatización para este tipo de redes no existen diversidad de herramientas o nodos sensores que soporten la programación con el propósito de automatizar la red. Sin embargo, empresas como Digi International permiten extender la automatización de la red con Python, gracias sus módulos XBee 3 que integran un microcontrolador que le da la capacidad de hardware necesaria para la automatización.

Las redes de telecomunicaciones están destinadas a evolucionar y la WSN no son la excepción, puesto que, fabricantes como Cisco [77], Red Hat [78] o Juniper [80] ofrecen dispositivos activos de la red orientados para la automatización de esta, la WSN enfrentan limitaciones para ser automatizadas, sin embargo, actualmente puede ser automatizadas de manera básica en comparación con otras infraestructuras de red ofertadas por fabricantes como Cisco. Es cuestión de tiempo para solventar las limitaciones que tiene la WSN con avances tecnológicos a nivel de hardware y software, logrando en futuro cercano la automatización completa de la WSN.

Actualmente en la búsqueda de información con respecto a la automatización de la WSN realizada en el presente documento se evidencia la automatización en este tipo de redes principalmente en el área de gestión y despliegue de nuevos nodos en la red, ejemplo de esto es Digi Remote Manager, la herramienta más completa para la automatización con Python en la WSN. Aunque la información de la automatización de redes inalámbricas de sensores aún es escasa, su implementación ya está en marcha.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1]. O. Américo, "Redes de sensores", Trabajo de especialización, Universidad Nacional de La Plata, La Plata, 2011. [En línea]. Disponible: <http://sedici.unlp.edu.ar/handle/10915/4196>
- [2]. A. Cama, G. Piñeres, Z. Comas, J. Vélez y F. Gómez, "Diseño de una red de monitorización de variables meteorológicas relacionadas a los tornados en Barranquilla-Colombia y su área metropolitana", *Ingeniare*, vol. 25, n.º 4, pp. 585–598, 2017.
- [3]. Silicon Laboratories Inc, *The Evolution of Wireless Sensor Networks*. Silicon Labs, 2013.
- [4]. S. Saigua Carvajal, M. Villafuerte Haro, D. Ávila Pesantez y A. Arellano, "Evaluación de las topologías físicas de redes WSN para la medición de variables ambientales", *Revista Científica y Tecnológica UPSE*, vol. 3, n.º 1, pp. 159–165, diciembre de 2015. [En línea]. Disponible: <https://doi.org/10.26423/rctu.v3i1.84>
- [5]. D. Archila y F. Santamaría, "Estado del arte de las redes de sensores inalámbricos", *Tecnología Investigación y Academia*, vol. 2, n.º 1, pp. 4–14, 2013.
- [6]. D. Naranjo-Hernández, J. Reina-Tosina y L. M. Roa, "Special issue "body sensors networks for e-health applications"", *Sensors*, vol. 20, n.º 14, p. 3944, julio de 2020. Accedido el 16 de febrero de 2022. [En línea]. Disponible: <https://doi.org/10.3390/s20143944>
- [7]. A. I. C. Arce, A. R. B. Tech, A. C. S. Silva y E. J. X. Costa, "Monitorización de rebaños de bovinos a través de redes de sensores inalámbricos", *Archivos de Zootecnia*, vol. 58, n.º 222, pp. 253–263, 2009.
- [8]. ElProCus. "Types of wireless sensor networks: Attacks & their applications". *Electronic Projects for Engineering Students*. [En línea]. Disponible: <https://www.elprocus.com/introduction-to-wireless-sensor-networks-types-and-applications/>
- [9]. M. S. Mahmoud y A. A. H. Mohamad, "A study of efficient power consumption wireless communication techniques/ modules for internet of things (IoT) applications", *Advances in Internet of Things*, vol. 06, n.º 02, pp. 19–29, 2016.

- Accedido el 16 de enero de 2022. [En línea]. Disponible: <https://doi.org/10.4236/ait.2016.62002>
- [10]. J. Miño, "Diseño e implementación de sensores de medición de contaminación del aire basado en una red lorawan en la escuela politécnica nacional", Trabajo de Titulación, Escuela Politécnica Nacional, Quito, 2021. [En línea]. Disponible: <https://bibdigital.epn.edu.ec/handle/15000/21403>
- [11]. Digi International Inc, Digi MicroPython Programming Guide, 5a ed. Digi International, 2021.
- [12]. J. Pérez, E. Urdaneta y Á. Custodio, "Metodología para el diseño de una red de sensores inalámbricos", Universidad, Ciencia y Tecnología, vol. 18, n.º 70, pp. 12–22, 2014.
- [13]. "Redes inalámbricas de sensores inteligentes", en XIII workshop de investigadores en ciencias de la computación, Rosario, Argentina, 6 de mayo de 2011. Rosario: Red de Universidades con Carreras en Informática, 2011, pp. 54–58.
- [14]. Venco. "Qué es ZigBee, cómo funciona y características principales". Venco Electrónica Industrial. [En línea]. Disponible: <https://www.vencoel.com/que-es-zigbee-como-funciona-y-caracteristicas-principales/>
- [15]. J. P. Dignani, "Análisis del protocolo ZigBee", Trabajo de especialización, Universidad Nacional de La Plata, La Plata, 2012. [En línea]. Disponible: <http://sedici.unlp.edu.ar/handle/10915/18349>
- [16]. J. Olsson, 6LoWPAN Demystified. Dallas: Texas Instruments, 2014.
- [17]. S. Kharb y A. Singhrova, "REVIEW OF INDUSTRIAL STANDARDS FOR WIRELESS SENSOR NETWORKS Introduction to Wireless Sensor Network (WSN)", en Csi-2015, New Delhi, 3 de diciembre de 2015. Pune: Bharati Vidyapeeth Deemed University, 2015.
- [18]. F. Labeau, A. Agarwal y B. Agba, "Comparative study of wireless sensor network standards for application in electrical substations", en 2015 International Conference on Computing, Communication and Security (ICCCS), Pamplemousses, 4–5 de diciembre de 2015. IEEE, 2015. [En línea]. Disponible: <https://doi.org/10.1109/cccs.2015.7374135>
- [19]. Lora Alliance, Lorawan and Nb-iot: Competitors or Complementary? New York: ABI Research, 2019.

- [20]. Tekniker. "Redes de sensores". Tekniker | Research and Technology Center. [En línea]. Disponible: <https://www.tekniker.es/es/redes-de-sensores>
- [21]. Thread Group, Thread 1.2 in Commercial White Paper. Thread Group, 2019.
- [22]. D. García. "Redes industriales inalámbricas: La libertad de movimientos en la fábrica". infoPLC. [En línea]. Disponible: <https://www.infoplcn.net/noticias/item/105230-hms-ersoft-redes-industriales-inalambricas-fabrica>
- [23]. Wi-SUN Alliance, WHITE PAPER: COMPARING IOT NETWORKS AT a GLANCE How Wi-Sun FAN Stacks Up Against LoRaWAN and NB-IoT. Wi-SUN, 2019.
- [24]. "Learn - Z-Wave". Z-Wave. [En línea]. Disponible: <https://www.z-wave.com/learn>
- [25]. J. P. Dignani, "Análisis del protocolo ZigBee", Trabajo de especialización, Universidad Nacional de La Plata, La Plata, 2012. [En línea]. Disponible: <http://sedici.unlp.edu.ar/handle/10915/18349>
- [26]. Digi International Inc, Wireless Connectivity Kit Getting Started Guide, 3a ed. Digi International, 2016. [En línea]. Disponible: <https://www.digi.com/resources/documentation/digidocs/pdfs/90001456-13.pdf>
- [27]. Digi International Inc. "Guía de compra de XBee". Digi. [En línea]. Disponible: <https://es.digi.com/blog/post/xbee-buying-guide>
- [28]. Digi International Inc. "¿Qué es la red de malla inalámbrica ZigBee?" Digi. [En línea]. Disponible: <https://es.digi.com/solutions/by-technology/zigbee-wireless-standard>
- [29]. Digi International Inc, Wireless Mesh Networking: Zigbee® vs. Digimesh. Baltimore: Digi International, 2018.
- [30]. Cisco. "Cisco meraki secure edge | ciberc". CiberC Partner Gold Cisco. [En línea]. Disponible: https://sase.ciberc.com/cisco-meraki-secure-edge?utm_term=sd%20wan%20firewall&utm_campaign=Cisco+Meraki+Tráfico+Web&utm_source=adwords&utm_medium=ppc&hsa_acc=3663981768&hsa_cam=15997907258&hsa_grp=135355150049&hsa_ad=576747264608&hsa_src=g&hsa_tgt=kwd-446651967285&hsa_kw=sd%20wan%20firewall&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gclid=CjwKCAiA6Y2QBhAtEiwAGHybPQAXyr5AUbwAy-nlrn4LzghOaf0zpsbVmEPzrBtclW97szldF4O-zRoCsDUQAvD_BwE

- [31]. J. C. García, A. Manotas, R. Acosta y A. Romero, "Revisión del estado del arte de Redes Zigbee en WSN", Investigación y desarrollo en TIC, vol. 5, n.º 1, 2014. [En línea].
Disponible: <https://revistas.unisimon.edu.co/index.php/identific/article/view/2477>
- [32]. Digi International Inc. Introduction to XCTU 6.3. (11 de diciembre de 2015).
Accedido el 21 de diciembre de 2021. [Video en línea]. Disponible:
https://www.youtube.com/watch?v=X_fHDvV_q98
- [33]. Xbee.cl. "¿Qué es XBee?" XBee.cl - Comunicación Inalámbrica para Tus Proyectos. [En línea]. Disponible: <https://xbee.cl/que-es-xbee/>
- [34]. Digi International Inc. "Explore el ecosistema Digi XBee". Digi. [En línea].
Disponible: <https://es.digi.com/xbee>
- [35]. MicroPython. "MicroPython - Python for microcontrollers". [En línea]. Disponible:
<https://micropython.org>
- [36]. MicroPython. "Overview — MicroPython 1.17 documentation". [En línea].
Disponible: <http://docs.micropython.org/en/latest/>
- [37]. Python. "General Python FAQ". 3.10.2 Documentation. [En línea]. Disponible:
<https://docs.python.org/3/faq/general.html#what-is-python>
- [38]. R. Moral. "Xbee-micropython/lib at master · digidotcom/xbee-micropython".
GitHub. [En línea]. Disponible: <https://github.com/digidotcom/xbee-micropython/tree/master/lib>
- [39]. R. Faludi. "Ejemplos de micropython para digi xbee parte 2: Edge computing".
Dispositivos y servicios de IoT industrial (IIoT) para redes M2M | Digi International. [En línea]. Disponible: [https://es.digi.com/blog/post/hands-on-micropython-programming-examples-for-\(1\)](https://es.digi.com/blog/post/hands-on-micropython-programming-examples-for-(1))
- [40]. R. Faludi. "Ejemplos de micropython para digi xbee parte 1: Edge computing".
Dispositivos y servicios de IoT industrial (IIoT) para redes M2M | Digi International. [En línea]. Disponible: <https://es.digi.com/blog/post/hands-on-micropython-programming-examples-for-edge>
- [41]. Python. "Download python". Python.org. [En línea]. Disponible:
<https://www.python.org/downloads/>

- [42]. M. Wadsten. "GitHub - digidotcom/xbee-micropython". GitHub. [En línea]. Disponible: <https://github.com/digidotcom/xbee-MicroPython>
- [43]. Digi International Inc, XBee Python Library Documentation. Digi International, 2021. [En línea]. Disponible: https://xbplib.readthedocs.io/_/downloads/en/1.4.0/pdf/
- [44]. Digi International Inc, Wireless Connectivity Kit Getting Started Guide, 3a ed. Digi International, 2016. [En línea]. Disponible: <https://www.digi.com/resources/documentation/digidocs/pdfs/90001456-13.pdf>
- [45]. Red Hat. "¿Qué es una API?" Red Hat - We make open source technologies for the enterprise. [En línea]. Disponible: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- [46]. Cisco. "¿Qué es la automatización de la red?" Cisco. [En línea]. Disponible: https://www.cisco.com/c/es_mx/solutions/automation/network-automation.html#~¿Por%20qué%20Cisco?
- [47]. Juniper. "¿Qué es la automatización de redes? | Juniper Networks". Juniper Networks. [En línea]. Disponible: <https://www.juniper.net/mx/es/research-topics/what-is-network-automation.html>
- [48]. Juniper, Juniper-Mist Delivers a Fullaccess Solution Built on Artificial Intelligence. Juniper Networks, 2020. [En línea]. Disponible: <https://www.juniper.net/content/dam/www/assets/solution-briefs/us/en/juniper-mist-delivers-a-full-access-solution-built-on-artificial-intelligence.pdf>
- [49]. Juniper. "¿Qué es la inteligencia artificial para redes? | Juniper Networks". Juniper Networks. [En línea]. Disponible: <https://www.juniper.net/mx/es/research-topics/what-is-ai-for-networking.html>
- [50]. NetApp. "¿Qué es el aprendizaje profundo? - Cómo funciona | NetApp". Data Management Solutions for the Cloud | NetApp. [En línea]. Disponible: <https://www.netapp.com/es/artificial-intelligence/what-is-deep-learning/>
- [51]. VMware. "What is Software-Defined Networking (SDN)? | VMware Glossary". VMware. [En línea]. Disponible: <https://www.vmware.com/es/topics/glossary/content/software-defined-networking.html>

- [52]. NIBIB, "Artificial Intelligence (AI)", National Institute of Biomedical Imaging and Bioengineering, pp. 1–2, 2020. [En línea]. Disponible: <https://www.nibib.nih.gov/sites/default/files/2020-06/Artificial-Intelligence-AI.pdf>
- [53]. Red Hat. "¿Qué son las redes definidas por software?" Red Hat - We make open source technologies for the enterprise. [En línea]. Disponible: <https://www.redhat.com/es/topics/hyperconverged-infrastructure/what-is-software-defined-networking>
- [54]. M. Alvarez. "La programabilidad de redes y automatización. introducción - coding networks blog". Coding Networks Blog. [En línea]. Disponible: <https://codingnetworks.blog/es/la-programabilidad-de-redes-y-automatizacion/>
- [55]. Red Hat. "¿Qué es la NFV?" Red Hat - We make open source technologies for the enterprise. [En línea]. Disponible: <https://www.redhat.com/es/topics/virtualization/what-is-nfv#sdn-y-nfv>
- [56]. Red Hat. "¿Qué es la automatización de la red?" Red Hat - We make open source technologies for the enterprise. [En línea]. Disponible: <https://www.redhat.com/es/topics/automation/what-is-network-automation>
- [57]. L. Berisso. "Automatización en Networking – NET4DD". NET4DD – Contribuyendo en la formación de expertos en TI. [En línea]. Disponible: <http://net4dd.com/automatizacion-en-networking/>
- [58]. Cisco. "Cisco DNA Center: Administración y automatización de redes". Cisco. [En línea]. Disponible: https://www.cisco.com/c/es_mx/products/cloud-systems-management/dna-center/index.html
- [59]. Netmiko. "Python for network engineers | articles". Python for Network Engineers. [En línea]. Disponible: <https://pynet.twb-tech.com/blog/automation/netmiko.html>
- [60]. M. Ndiaye, G. Hancke y A. Abu-Mahfouz, "Software defined networking for improved wireless sensor network management: A survey", Sensors, vol. 17, n.º 5, p. 1031, mayo de 2017. [En línea]. Disponible: <https://doi.org/10.3390/s17051031>
- [61]. J. Rizo, Técnicas de investigación documental. Managua: Universidad Nacional Autónoma de Nicaragua, 2015.
- [62]. D. Archila y F. Santamaría, "Estado del arte de las redes de sensores inalámbricos", Tecnología Investigación y Academia, vol. 2, n.º 1, pp. 4–14, 2013.

- [63]. Red Hat, Automatización de la red para todos. Red Hat, 2021. [En línea]. Disponible: <https://www.redhat.com/rhdc/managed-files/rh-network-automation-for-everyone-ebook-f26682-202101-a4-es.pdf>
- [64]. M. Ndiaye, G. Hancke y A. Abu-Mahfouz, "Software defined networking for improved wireless sensor network management: A survey", *Sensors*, vol. 17, n.º 5, p. 1031, mayo de 2017. [En línea]. Disponible: <https://doi.org/10.3390/s17051031>
- [65]. K. M. Modieginyane, B. B. Letswamotse, R. Malekian y A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey", *Computers & Electrical Engineering*, vol. 66, pp. 274–287, febrero de 2018. Accedido el 16 de febrero de 2022. [En línea]. Disponible: <https://doi.org/10.1016/j.compeleceng.2017.02.026>
- [66]. R. Huang, X. Chu, J. Zhang y Y. H. Hu, "Energy-Efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype", *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–12, 2015. Accedido el 16 de febrero de 2022. [En línea]. Disponible: <https://doi.org/10.1155/2015/360428>
- [67]. G. Manikandan y U. Sakthi, "A comprehensive survey on various key management schemes in WSN", en *2018 2nd International Conference on I-Smac (IoT in Social, Mobile, Analytics and Cloud) (I-Smac)*, Palladam, India, 30–31 de agosto de 2018. IEEE, 2018. Accedido el 16 de febrero de 2022. [En línea]. Disponible: <https://doi.org/10.1109/i-smac.2018.8653656>
- [68]. L. L. Bello, A. Lombardo, S. Milardo, G. Patti y M. Reno, "Software- defined networking for dynamic control of mobile industrial wireless sensor networks", en *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, 4–7 de septiembre de 2018. IEEE, 2018. Accedido el 16 de febrero de 2022. [En línea]. Disponible: <https://doi.org/10.1109/etfa.2018.8502457>
- [69]. inteliLIGHT. "StreetLight control software - remote management software". inteliLIGHT®. [En línea]. Disponible: <https://intellilight.eu/intellilight-streetlight-control-software/>
- [70]. Senraco. "PAN india lorawan network operator for iot". Senraco. [En línea]. Disponible: <https://senraco.com>

- [71]. Digi International Inc. "Xctu". Digi. [En línea]. Disponible: <https://es.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>
- [72]. Digi International Inc. "Digi remote manager". Dispositivos y servicios de IoT industrial (IIoT) para redes M2M | Digi International. [En línea]. Disponible: <https://es.digi.com/products/iot-software-services/digi-remote-manager#overview>
- [73]. R. Faludi. "Ejemplos de micropython para digi xbee parte 2: Edge computing". Dispositivos y servicios de IoT industrial (IIoT) para redes M2M | Digi International. [En línea]. Disponible: [https://es.digi.com/blog/post/hands-on-micropython-programming-examples-for-\(1\)](https://es.digi.com/blog/post/hands-on-micropython-programming-examples-for-(1))
- [74]. M. Wadsten. "GitHub - digidotcom/xbee-micropython". GitHub. [En línea]. Disponible: <https://github.com/digidotcom/xbee-MicroPython>
- [75]. R. Moral. "Xbee-micropython/lib at master · digidotcom/xbee-micropython". GitHub. [En línea]. Disponible: <https://github.com/digidotcom/xbee-micropython/tree/master/lib>
- [76]. R. Faludi. "Ejemplos de micropython para digi xbee parte 1: Edge computing". Dispositivos y servicios de IoT industrial (IIoT) para redes M2M | Digi International. [En línea]. Disponible: <https://es.digi.com/blog/post/hands-on-micropython-programming-examples-for-edge>
- [77]. Cisco. "¿Qué es la automatización de la red?" Cisco. [En línea]. Disponible: https://www.cisco.com/c/es_mx/solutions/automation/network-automation.html#~¿Por%20qué%20Cisco?
- [78]. Red Hat, Automatización de la red para todos. Red Hat, 2021. [En línea]. Disponible: [En línea]. Disponible: <https://www.redhat.com/rhdc/managed-files/rh-network-automation-for-everyone-ebook-f26682-202101-a4-es.pdf>
- [79]. Juniper, Juniper-Mist Delivers a Fullaccess Solution Built on Artificial Intelligence. Juniper Networks, 2020. [En línea]. Disponible: <https://www.juniper.net/content/dam/www/assets/solution-briefs/us/en/juniper-mist-delivers-a-full-access-solution-built-on-artificial-intelligence.pdf>