



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

– **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

Desarrollo de una plataforma web de gestión y planificación
de redes de sensores inalámbricos definidas por software

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

AUTOR/A: Lorente Sierra, David

Tutor/a: Sempere Paya, Víctor Miguel

Cotutor/a externo: Vera Perez, Jose

CURSO ACADÉMICO: 2023/2024

Resumen

Las redes de sensores inalámbricas, conocidas como WSN (del inglés Wireless Sensor Network) han cobrado una gran relevancia en muchos ámbitos de la sociedad, desde la monitorización de entornos urbanos o Smart Cities, la caracterización y accionamiento de diferentes procesos industriales o incluso cubrir grandes extensiones forestales o de agricultura para llevar a cabo una labor de precisión. Sobre esto, las redes definidas por software, conocidas como SDN (del inglés Software Defined Networks) aportan mayor determinismo y versatilidad sobre estas redes WSN, permitiendo orquestar de manera centralizada cada intervención en la red por parte de cualquier dispositivo. Estas redes y tecnologías a menudo cuentan con plataformas de gestión dedicadas, o herramientas de planificación adicionales, que dificultan romper la barrera de entrada de estas tecnologías a un usuario menos familiarizado con las tecnologías y comunicaciones que se emplean. Con afán de facilitar el uso de este tipo de sistemas SDWSN, este trabajo tiene como objetivo desarrollar una plataforma de gestión, monitorización y simulación basada en la tendencia de Bring Your Own Device (BYOD), en la que a través de un servidor web se puede realizar la configuración, puesta en marcha, seguimiento de despliegue, simulación de recursos y visualización de datos en tiempo real desde cualquier dispositivo con un navegador. Este desarrollo web estará orientado al frontend del sistema, exponiendo diferentes vistas específicas para cada proceso relacionado con el despliegue y puesta en funcionamiento de una WSN. Además, se propondrá una herramienta de simulación de recursos radio, en la que de manera aislada se puedan validar las configuraciones de manera previa a la configuración del dispositivo físico.

Resum

Les xarxes de sensors sense fils, conegudes com WSN (Wireless Sensor Networks) han cobrat una gran rellevància en molts àmbits de la societat, des de la monitorització d'entorns urbans o Smart Cities, la caracterització i actuació de diferents processos industrials o fins i tot cobrir grans extensions forestals o d'agricultura per a dur a terme una tasca de precisió. Sobre això, les xarxes definides per programari, conegudes com SDN (Software Defined Networks) aporten major determinisme i versatilitat sobre aquestes xarxes WSN, permetent orquestrar de manera centralitzada cada intervenció en la xarxa per part de qualsevol dispositiu. Aquestes xarxes i tecnologies sovint compten amb plataformes de gestió dedicades, o eines de planificació addicionals, que dificulten trencar la barrera d'entrada d'aquestes tecnologies a un usuari menys familiaritzat amb les tecnologies i comunicacions que s'empren. Amb l'objectiu de facilitar l'ús d'aquest tipus de sistemes SDWSN, aquest treball té com a objectiu el desenvolupament d'una plataforma de gestió, monitorització i simulació basada en la tendència de Bring Your Own Device (BYOD), en la qual a través d'un servidor web es pot realitzar la configuració, posada en marxa, seguiment de desplegament, simulació de recursos i visualització de dades en temps real des de qualsevol dispositiu amb un navegador. Aquest desenvolupament web estarà orientat al frontend del sistema, exposant diferents vistes específiques per a cada procés relacionat amb el desplegament i posada en servei de la WSN. A més, es proposarà una eina de simulació de recursos radio, en la qual de manera aïllada es puguen validar les configuracions de manera prèvia a la configuració del dispositiu físic.

Abstract

Wireless Sensor Networks (WSN) have gained significant relevance in many areas of the society, from urban environment monitoring or Smart Cities to the characterization and operation of various industrial processes, or even covering large forest or agricultural areas to perform precision tasks. In this context, Software Defined Networks (SDN) provide greater determinism and versatility to these WSNs, allowing for centralized orchestration of each network intervention by any device. These networks and technologies often have dedicated management platforms or additional planning tools, which make it challenging for users less familiar with the technologies and communications employed to overcome the entry barrier. In an effort to facilitate the use of such SDWSN systems, this work aims to develop a management, monitoring, and simulation platform based on the Bring Your Own Device (BYOD) trend, where configuration, deployment, monitoring, resource simulation, and real-time data visualization can be carried out via a web server from any device with a browser. This web development will be oriented towards the frontend of the system, displaying different specific views for each process related to the deployment and commissioning of the WSN. Additionally, a radio resource simulation tool will be proposed, where configurations can be validated in isolation before the physical device configuration.

Resumen ejecutivo

La memoria del TFG del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la ingeniería de telecomunicación

| CONCEPT (ABET) | CONCEPTO (traducción) | ¿Cumple? (S/N) | ¿Dónde? (páginas) |
|--|---|---------------------------|------------------------------|
| 1. IDENTIFY: | 1. IDENTIFICAR: | | |
| 1.1. Problem statement and opportunity | 1.1. Planteamiento del problema y oportunidad | S | 12 – 13, 38 – 55 |
| 1.2. Constraints (standards, codes, needs, requirements & specifications) | 1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones) | S | 14 – 55 |
| 1.3. Setting of goals | 1.3. Establecimiento de objetivos | S | 12 – 13, 79 – 97 |
| 2. FORMULATE: | 2. FORMULAR: | | |
| 2.1. Creative solution generation (analysis) | 2.1. Generación de soluciones creativas (análisis) | S | 79 – 126 |
| 2.2. Evaluation of multiple solutions and decision-making (synthesis) | 2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis) | S | 38 – 97 |
| 3. SOLVE: | 3. RESOLVER: | | |
| 3.1. Fulfilment of goals | 3.1. Evaluación del cumplimiento de objetivos | S | 98 - 126 |
| 3.2. Overall impact and significance (contributions and practical recommendations) | 3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas) | S | 98 - 126 |

Tabla de contenido

| | |
|---|----|
| 1. Introducción y objetivos..... | 12 |
| 1.1. Objetivo del trabajo..... | 12 |
| 2. Marco teórico: Redes de sensores inalámbricos (WSN). | 14 |
| 2.1. Tecnologías WPAN: Redes inalámbricas de área personal. | 15 |
| 2.1.1. Bluetooth* | 16 |
| 2.1.2. Estándar IEEE 802.15.4e | 19 |
| 2.1.9. Tabla comparativa de tecnologías de área personal..... | 23 |
| 2.2. Tecnologías WLAN: Redes inalámbricas de área local. | 23 |
| 2.2.1. Tecnologías WiFi: 802.11a/b/g, 802.11n, 802.11ac y 802.11ax | 24 |
| 2.2.2. WiFi HaLow (802.11ab)..... | 25 |
| 2.3. Tecnologías WWAN: Redes inalámbricas de vecindario..... | 25 |
| 2.3.1. Wi-SUN..... | 26 |
| 2.3.2. JupiterMesh | 28 |
| 2.3.3. Wireless M-Bus | 28 |
| 2.4. Tecnologías WWAN: Redes inalámbricas de área amplia. | 30 |
| 2.4.1. Tecnologías de comunicaciones móviles: GSM, GPRS, UMTS, HSPA, HSPA+, LTE y 5G..... | 30 |
| 2.4.2. LET-M y NB-IoT | 32 |
| 2.4.3. LoRa..... | 33 |
| 2.4.4. SigFox | 36 |
| 2.4.5. Comparación de tecnologías LPWAN: Redes de área amplia de baja potencia..... | 36 |
| 3. Estado del arte y análisis del mercado: Herramientas y soluciones para el despliegue y monitorización de WSN. | 38 |
| 3.1. Análisis de mercado de fases del despliegue de una WSN | 41 |
| 3.1.1. Fase de antes del despliegue (Pre-Despliegue) | 41 |
| 3.1.2. Fase de despliegue (Despliegue “In-Site”)..... | 43 |
| 3.1.3. Fase de después del despliegue (Post-Despliegue) | 48 |
| 3.2. Análisis de Mercado de Tecnologías y herramientas de funcionamiento desasistido | 49 |
| 3.2.1. VersaSense..... | 49 |
| 3.2.2. Solución de desarrollo Zigbee de NXP | 51 |
| 3.2.4. Analog Devices (SmartMesh IP Network)..... | 52 |
| 3.2.5. Adeunis..... | 53 |
| 3.2.6. ACKCIO..... | 54 |
| 4. Metodología: Tecnologías y herramientas empleadas para el desarrollo de la aplicación web QDSN..... | 56 |
| 4.1. Herramientas de desarrollo web utilizadas..... | 56 |

| | |
|---|-----|
| 4.1.1. GitLab | 56 |
| 4.1.2. Docker | 59 |
| 4.1.3. Vue.js..... | 65 |
| 4.1.4. Entorno Linux simulado con el subsistema de Windows para Linux (WSL) | 67 |
| 4.2. Elección del Framework..... | 69 |
| 4.3. Entorno de Desarrollo | 72 |
| 4.4. Arquitectura de la aplicación..... | 77 |
| 5. Requisitos de la plataforma a desarrollar para el despliegue y monitorización de redes de sensores inalámbricas..... | 79 |
| 5.1. Definición de requisitos de diseño de comunicaciones | 80 |
| 5.2. Requisitos generales de la interfaz de usuario a desarrollar..... | 83 |
| 5.4. Diseño de la interfaz de usuario a desarrollar | 85 |
| 5.4.1. Flujo de trabajo de un operario desde que arranca el sistema | 87 |
| 5.4.1. Inicio de sesión y perfil de usuario..... | 87 |
| 5.4.2. Menú o navegador entre diferentes ventanas de visualización | 88 |
| 5.4.3. Página principal o página de inicio | 89 |
| 5.4.4. Despliegue de nodos..... | 90 |
| 5.4.5. Asistente de despliegue de nodos | 91 |
| 5.4.6. Simulador | 92 |
| 5.4.7. Estadísticas de red | 93 |
| 5.4.8. Datos en tiempo real..... | 95 |
| 5.4.9. Configuración..... | 96 |
| 6. Resultado y aplicación web final. | 98 |
| 6.1. Pantalla de inicio de sesión (Login) | 98 |
| 6.2. Pantalla principal de la aplicación (MainLayout) | 101 |
| 6.3. Pantalla de inicio (Home)..... | 104 |
| 6.3.1. Primera Tarjeta: Gráfico de Topología de la Red | 105 |
| 6.3.2. Segunda Tarjeta: Estado del Despliegue de la Red (Deploids) | 105 |
| 6.3.3. Tercera Tarjeta: Estado General del Sistema (System)..... | 106 |
| 6.3.4. Cuarta Tarjeta: Estado de la Conexión Ethernet..... | 106 |
| 6.3.5. Quinta Tarjeta: Conectividad WiFi..... | 107 |
| 6.4. Pantalla de despliegue de nodos (Node Deployment)..... | 108 |
| 6.4.1. Tabla resumen de nodos | 108 |
| 6.4.2. Configuración del nodo..... | 110 |
| 6.4.3. Barra de búsqueda | 111 |
| 6.4.4. Botones de acción..... | 112 |
| 6.5. Pantalla del simulador (Simulator)..... | 114 |
| 6.6. Pantalla de estadísticas de la red (Network)..... | 116 |

| | |
|--|-----|
| 6.7. Pantalla de datos en tiempo real (Livestream) | 118 |
| 6.8. Pantalla de configuración del sistema (Configuration) | 120 |
| 7. Conclusiones | 124 |
| Bibliografía | 126 |
| Anexo I: Endpoints de la API REST de la aplicación web. | 131 |
| ➤ Endpoint de Inicio de Sesión:..... | 131 |
| ➤ Endpoint para actualizar el nombre de un nodo en la WSN: | 131 |
| ➤ Endpoint para obtener el estado del gateway: | 132 |
| ➤ Endpoint para obtener información sobre WSN: | 133 |
| ➤ Endpoint para obtener información sobre un nodo específico en la WSN:..... | 134 |
| ➤ Endpoint para obtener la lista de sensores disponibles: | 135 |
| ➤ Endpoint para configurar un nuevo sensor en un nodo en la WSN:..... | 135 |
| ➤ Endpoint para obtener la lista de sensores asociados a un nodo en la WSN. | 136 |
| ➤ Endpoint para eliminar un sensor de un nodo en la WSN..... | 136 |
| ➤ Endpoint para actualizar un sensor en un nodo en la WSN: | 137 |
| Anexo II: Relación del trabajo con los ODS de la Agenda 2030 | 138 |

Índice de figuras

| | |
|---|-----|
| Figura 1: Tecnologías inalámbricas de comunicación organizadas por rango de cobertura | 15 |
| Figura 2: Velocidad en las distintas versiones de Bluetooth | 17 |
| Figura 3: Pila de protocolos para Bluetooth Low Energy | 17 |
| Figura 4: Principales bandas de frecuencia en IEEE 802.15.4..... | 20 |
| Figura 5: Distribución temporal y frecuencial utilizada en TSCH..... | 22 |
| Figura 6: Ejemplo de arquitectura de un sistema de contadores basado en Wireless M-Bus..... | 29 |
| Figura 7: Evolución y especificaciones básicas de las tecnologías celulares..... | 31 |
| Figura 8: Comparación NB-IoT y LTE-M | 33 |
| Figura 9: Solución del proyecto LUSTER | 44 |
| Figura 10: Representación de estadísticas de calidad de TASK..... | 44 |
| Figura 11: Tareas de despliegue mediante equipos móviles..... | 45 |
| Figura 12: Demostración de despliegue mediante aplicación móvil..... | 46 |
| Figura 13: Field Devices de WirelessHART e ISA100.11a | 47 |
| Figura 14: Arquitectura de Field Xpert | 47 |
| Figura 15: Solución de VersaSense | 49 |
| Figura 16: Solución de Desarrollo Zigbee de NXP..... | 51 |
| Figura 17: Starter Kit Analog Devices SmartMesh IP | 53 |
| Figura 18: Solución de Adeunis | 54 |
| Figura 19: Topología de la red de ACKCIO..... | 54 |
| Figura 20: Solución de ACKCIO | 55 |
| Figura 21: Logo de GitLab..... | 56 |
| Figura 22: Logo de Docker | 59 |
| Figura 23: Logo del Framework Vue.js..... | 65 |
| Figura 24: Comparativa de tiempos entre React, Angular y Vue | 70 |
| Figura 25: Comparativa de rendimiento entre React, Angular y Vue | 71 |
| Figura 26: Pasos a seguir para crear un Personal Access Token en GitLab | 72 |
| Figura 27: Diagrama de la arquitectura de la aplicación..... | 78 |
| Figura 28: Flujo de trabajo para el despliegue de una WSN..... | 79 |
| Figura 29: Nodos de sensores del sistema QDSN..... | 79 |
| Figura 30: Diseño conceptual de la herramienta de planificación | 83 |
| Figura 31: Boceto de la página de inicio de session..... | 88 |
| Figura 32: Boceto del botón para información del perfil | 88 |
| Figura 33: Boceto del menú principal de la aplicación | 89 |
| Figura 34: Boceto de la página de inicio..... | 90 |
| Figura 35: Boceto de la página de despliegue de nodos..... | 91 |
| Figura 36: Boceto de la página de asistente de despliegue | 92 |
| Figura 37: Boceto de la página del simulador..... | 93 |
| Figura 38: Boceto de la página de estadísticas de la red | 93 |
| Figura 39: Boceto de la página de la topología de red | 94 |
| Figura 40: Boceto de la página de datos en vivo..... | 96 |
| Figura 41: Boceto de la página de configuración de red..... | 96 |
| Figura 42: Pantalla de inicio de session | 98 |
| Figura 43: Menú vertical de navegación de la aplicación | 101 |
| Figura 44: Funcionalidad de exportar a CSV de la aplicación | 102 |
| Figura 45: Elección de fecha en la funcionalidad de exportar a CSV | 102 |
| Figura 46: Elección de horario en la funcionalidad de exportar a CSV | 103 |
| Figura 47: Vista del desplegable para exportar a CSV tras elegir fechas..... | 103 |
| Figura 48: Funcionalidad de elección de idioma en la aplicación..... | 103 |
| Figura 49: Funcionalidad de cambiar contraseña en la aplicación..... | 104 |

| | |
|---|-----|
| Figura 50: Página de inicio (Home) de la aplicación | 104 |
| Figura 51: Tarjeta que muestra el gráfico de topología de la red | 105 |
| Figura 52: Tarjeta resumen del estado de despliegue de nodos..... | 105 |
| Figura 53: Tarjeta resumen del estado del sistema..... | 106 |
| Figura 54: Tarjeta del estado Ethernet de la aplicación..... | 107 |
| Figura 55: Tarjeta del estado Wi-Fi de la aplicación..... | 107 |
| Figura 56: Página de despliegue de nodos de la aplicación | 108 |
| Figura 57: Diferentes estados de los nodos | 109 |
| Figura 58: Botón de acción en la pantalla de despliegue de nodos | 109 |
| Figura 59: Funcionalidad de cambiar el nombre o periodo de los nodos..... | 110 |
| Figura 60: Funcionalidad de añadir sensores a un nodo en la aplicación | 111 |
| Figura 61: Barra de búsqueda en la pantalla de despliegue de nodos de la aplicación | 111 |
| Figura 62: Botón de lanzar el simulador en la pantalla de despliegue de nodos..... | 112 |
| Figura 63: Botón de lanzar nodos al controlador habilitado | 112 |
| Figura 64: Botón de lanzar nodos al controlador deshabilitado | 112 |
| Figura 65: Botón del asistente de despliegue de nodos..... | 112 |
| Figura 66: Paso 1 del asistente de despliegue de nodos de la aplicación | 113 |
| Figura 67: Paso 2 del asistente de despliegue de nodos de la aplicación | 114 |
| Figura 68: Paso 3 del asistente de despliegue de nodos de la aplicación | 114 |
| Figura 69: Pantalla de simulación de la aplicación en modo informativo | 115 |
| Figura 70: Pantalla de simulación de la aplicación en modo warning | 115 |
| Figura 71: Pantalla de Network de la aplicación..... | 117 |
| Figura 72: Botón para abrir la Topología en la pantalla de Network | 117 |
| Figura 73: Gráfico de topología en la pantalla de Network | 117 |
| Figura 74: Selección de sensores en la página de datos en vivo | 118 |
| Figura 75: Sensores seleccionados en la página de datos en vivo..... | 118 |
| Figura 76: Página de datos en vivo sin seleccionar sensores | 119 |
| Figura 77: Gráfico de Línea en la pantalla de datos en vivo | 119 |
| Figura 78: Gráfico de Barras en la pantalla de datos en vivo..... | 119 |
| Figura 79: Gráfico de Gauge en la pantalla de datos en vivo..... | 120 |
| Figura 80: Pantalla de configuración con los desplegables plegados..... | 120 |
| Figura 81: Configuración de red en la pantalla de configuración | 120 |
| Figura 82: Configuración del punto de acceso Wi-Fi en la pantalla de configuración | 121 |
| Figura 83: Almacenamiento local en pantalla de configuración | 121 |
| Figura 84: Configuración de cliente MQTT en pantalla de configuración | 122 |
| Figura 85: Kafka en pantalla de configuración | 122 |
| Figura 86: OPC-UA en pantalla de configuración | 123 |

Índice de fragmentos de código

| | |
|--|-----|
| Fragmento de código 1: Ejemplo de Dockerfile para una aplicación de Python | 62 |
| Fragmento de código 2: Ejemplo de docker-compose.yml | 64 |
| Fragmento de código 3: Ejemplo de código Vue.js (fichero.vue) | 65 |
| Fragmento de código 4: Ejemplo de código Vue.js (fichero.html) | 65 |
| Fragmento de código 5: Dockerfile de la imagen de Docker “backend-demo” | 74 |
| Fragmento de código 6: Dockerfile de la imagen de Docker “node-simulator” | 74 |
| Fragmento de código 7: Dockerfile de la imagen de Docker “controller-sim” | 75 |
| Fragmento de código 8: Script para correr el backend de la aplicación..... | 75 |
| Fragmento de código 9: Fichero docker-compose.yml para correr la aplicación..... | 76 |
| Fragmento de código 10: Dockerfile del front-end de la aplicación | 77 |
| Fragmento de código 11: Función “signIn” | 99 |
| Fragmento de código 12: Función “connect” del módulo “RealTimeModule” | 100 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Objetivos de cada capítulo del proyecto..... | 13 |
| Tabla 2: Comparativa de la capa PHY en Bluetooth 5 | 18 |
| Tabla 3: Comparación técnica de las diferentes versiones Bluetooth | 19 |
| Tabla 4: Bandas más usadas en IEEE 802.15.4..... | 20 |
| Tabla 5: Comparación de especificaciones WPAN | 23 |
| Tabla 6: Comparación de especificaciones Wi-Fi..... | 25 |
| Tabla 7: Modos de operación Wireless M-Bus | 29 |
| Tabla 8: Tasas de transferencia para los diferentes modos de funcionamiento de LoRaWAN ... | 35 |
| Tabla 9: Comparación de conectividad de tecnologías LPWAN..... | 37 |
| Tabla 10: Requisitos de diseño de comunicaciones | 82 |

1. Introducción y objetivos

El presente documento expone el desarrollo de una aplicación web de gestión, monitorización y simulación basada en la tendencia de Bring Your Own Device (BYOD), en la que a través de un servidor web se puede realizar la configuración, puesta en marcha, seguimiento de despliegue, simulación de recursos y visualización de datos en tiempo real desde cualquier dispositivo con un navegador. Este desarrollo web estará orientado al front-end del sistema, exponiendo diferentes vistas específicas para cada proceso relacionado con el despliegue y configuración de una red de sensores inalámbricos.

En primer lugar, se busca tener un marco teórico sobre las redes de sensores inalámbricos y exponiendo los diferentes tipos de redes clasificándolas en cuanto a rango de cobertura, además, se realiza un análisis de mercado, con el objetivo de obtener una visión clara para poder realizar el diseño y posterior desarrollo de la aplicación web.

En segundo lugar, se realiza el estudio de mercado, buscando herramientas similares y tecnologías que serán de utilidad para el desarrollo del sistema de despliegue de redes de sensores inalámbricos.

Finalmente, se realiza el desarrollo de la aplicación, para lo que se hace un estudio de las herramientas utilizadas, se seleccionan las mejores herramientas para el desarrollo y se expone la arquitectura del código fuente de la aplicación.

1.1. Objetivo del trabajo

Este documento tiene como objetivo desarrollar una aplicación de gestión, monitorización y simulación basada en la tendencia de Bring Your Own Device (BYOD), en la que a través de un servidor web se puede realizar la configuración, puesta en marcha, seguimiento de despliegue, simulación de recursos y visualización de datos en tiempo real desde cualquier dispositivo con un navegador. Este desarrollo web estará orientado al frontend del sistema, exponiendo diferentes vistas específicas para cada proceso relacionado con el despliegue y configuración de una red de sensores inalámbricos. Además, se propondrá una herramienta de simulación de recursos radio, en la que de manera aislada se puedan validar las configuraciones de manera previa a la configuración del dispositivo físico.

Además, cada sección de este documento está diseñada con un propósito particular para lograr los objetivos previamente mencionados a través del desarrollo del trabajo. A continuación, se detallan estos objetivos específicos:

| Capítulo del proyecto | Objetivo |
|-----------------------|--|
| 1. Introducción | El objetivo del primer capítulo es proporcionar al lector un resumen del proyecto, destacando los diversos objetivos que se persiguen, junto con una explicación exhaustiva de la metodología utilizada para elaborar las diferentes partes. |

| | |
|--|---|
| <p>2. Marco teórico: Redes de sensores inalámbricos (WSN).</p> | <p>El segundo capítulo se enfoca en un análisis detallado de las redes de sensores inalámbricas, que son el objeto de estudio en este proyecto. Específicamente, ofrece al lector una visión general de las principales redes de sensores inalámbricos disponibles en el mercado, clasificándolas según su rango de cobertura: WPAN, WLAN, WNAN y WWAN.</p> |
| <p>3. Estado del arte y análisis del mercado: Herramientas y soluciones para el despliegue y monitorización de WSN.</p> | <p>El objetivo del tercer capítulo es proporcionar una visión representativa de las redes de sensores inalámbricas disponibles en el mercado. Primero, se realiza un estudio de mercado que considera las etapas de despliegue de estas redes. Luego, se examinan las herramientas similares que existen en el mercado.</p> |
| <p>4. Metodología: Tecnologías y herramientas empleadas para el desarrollo de la aplicación web QDSN.</p> | <p>El cuarto capítulo tiene objetivo exponer las diferentes herramientas utilizadas para llevar a cabo el desarrollo del proyecto. Además, se indican pasos concretos para poner a punto el entorno de desarrollo, así, un desarrollador ajeno al proyecto sería capaz de continuar el desarrollo siguiendo este documento.</p> |
| <p>5. Requisitos de la plataforma a desarrollar para el despliegue y monitorización de redes de sensores inalámbricos.</p> | <p>El objetivo del quinto capítulo es detallar los requisitos del sistema, tanto de software como de hardware. Además, en este apartado se presenta un diseño de la interfaz de usuario que se va a desarrollar.</p> |
| <p>6. Resultado y aplicación web final.</p> | <p>El propósito del capítulo sexto es presentar una serie de capturas de pantalla de las distintas vistas de la aplicación desarrollada. Asimismo, se proporciona una explicación detallada sobre la estructura y el proceso de desarrollo realizado.</p> |
| <p>7. Conclusiones</p> | <p>El objetivo del último apartado de este documento es recopilar las diferentes conclusiones que se ha llegado en el desarrollo del trabajo.</p> |

Tabla 1: Objetivos de cada capítulo del proyecto. (Fuente: Elaboración propia)

2. Marco teórico: Redes de sensores inalámbricos (WSN).

Las redes de sensores inalámbricos (WSN en adelante) representan un área especializada dentro de las tecnologías de comunicación inalámbrica, diseñadas para la monitorización y recopilación de datos en entornos físicos. Estas redes están compuestas por nodos de sensores distribuidos en una ubicación específica, cada uno equipado con sensores para medir variables ambientales como temperatura, humedad, presión, luz, movimiento, entre otros.

Cada nodo de sensor en una WSN es un dispositivo autónomo, generalmente de tamaño reducido, que comprende un procesador, memoria, uno o varios sensores, un transceptor de radio y una fuente de energía, típicamente una batería. Estos nodos de sensores trabajan juntos para recopilar datos del entorno circundante y transmitirlos a una estación base o un nodo central para su análisis y procesamiento ulterior.

Una de las características fundamentales de las WSN es su topología y distribución. Los nodos de sensores pueden organizarse en diferentes configuraciones topológicas, como en malla, en árbol, en estrella o en enjambre, dependiendo de los requisitos específicos de la aplicación y las condiciones del entorno. Esta distribución estratégica de los nodos de sensores permite una cobertura eficiente del área de interés y una comunicación efectiva entre los nodos.

La comunicación y el enrutamiento de datos son aspectos críticos en las WSN. Los nodos de sensores deben ser capaces de transmitir datos de manera eficiente entre sí y hacia la estación base, utilizando enlaces inalámbricos. Para lograr esto, se emplean diversos protocolos de comunicación y enrutamiento que determinan la ruta óptima de transmisión, teniendo en cuenta factores como el consumo de energía, la congestión de la red y la fiabilidad de la comunicación.

La eficiencia energética es otro aspecto crucial en el diseño y funcionamiento de las WSN, ya que muchos nodos de sensores dependen de baterías con recursos limitados. Para maximizar la vida útil de la batería, se implementan técnicas de gestión de energía, como el ahorro de energía en el nodo, el enrutamiento de bajo consumo de energía y el control de la transmisión, que ayudan a reducir el consumo de energía durante las operaciones de la red.

Las WSN encuentran aplicaciones en una amplia variedad de campos, incluyendo la monitorización ambiental, la agricultura de precisión, la salud, la seguridad, la logística y la gestión de infraestructuras. Estas redes proporcionan datos en tiempo real sobre el entorno físico, permitiendo la toma de decisiones basada en datos y el monitoreo continuo de condiciones críticas.

Sin embargo, el diseño y despliegue de WSN también plantea desafíos significativos. Además de la gestión de la energía, otros desafíos incluyen la escalabilidad de la red, la seguridad de los datos, la confiabilidad de la comunicación, la sincronización de los nodos y la integración con sistemas existentes. Además, las condiciones del entorno, como la interferencia electromagnética, el desvanecimiento de la señal y la variabilidad del terreno, pueden influir en el rendimiento y la operación de la red.

En resumen, las Redes de Sensores Inalámbricos representan una herramienta poderosa para la monitorización y recopilación de datos en entornos físicos, con una amplia gama de aplicaciones y desafíos específicos que deben abordarse para su implementación exitosa. Estas redes juegan un papel crucial en la recolección de información en tiempo real y la toma de decisiones en una variedad de campos, contribuyendo al avance de la tecnología y la mejora de nuestra comprensión del mundo que nos rodea.

En la Figura 1 se presenta una categorización de tecnologías de comunicación inalámbrica basadas en su alcance, abarcando desde conexiones de corta distancia hasta enlaces que pueden cubrir varias decenas de kilómetros.

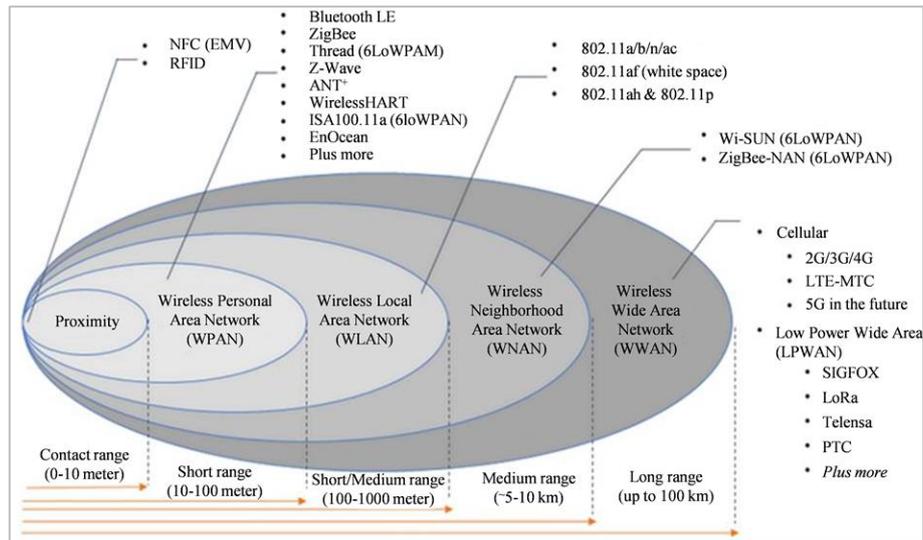


Figura 1: Tecnologías inalámbricas de comunicación organizadas por rango de cobertura. (Fuente: [1])

A continuación, se presenta un análisis de diversas tecnologías de comunicación inalámbrica clasificadas según su categoría basada en el rango de cobertura. Para ello se mostrarán las prestaciones más significativas de cada una de estas tecnologías, además de mostrar el interés que tienen en el ámbito del “Smart Metering”¹.

2.1. Tecnologías WPAN: Redes inalámbricas de área personal.

Las redes WPAN son sistemas de comunicación diseñados para conectar dispositivos en un espacio reducido, generalmente de unos pocos metros. Estas redes facilitan la conectividad entre dispositivos personales y electrónicos, permitiendo una interacción fluida y eficiente. Su alcance es corto, típicamente entre 1 y 10 metros, lo que las hace ideales para entornos personales. Además, se caracterizan por su bajo consumo energético, esencial para dispositivos portátiles, y son económicas en comparación con otras soluciones de red. La facilidad de implementación es otra de sus ventajas, ya que no requieren una infraestructura compleja, lo que simplifica su instalación y mantenimiento.

Entre las principales ventajas de las WPAN se encuentran su flexibilidad, ya que permiten conectar múltiples dispositivos sin necesidad de cables, y su escalabilidad, siendo fáciles de expandir con la adición de más dispositivos. Además, muchas de estas tecnologías son compatibles entre sí, lo que facilita la integración de diferentes dispositivos en una misma red.

No obstante, presentan algunos desafíos, como la posibilidad de interferencias con otros dispositivos que operan en frecuencias similares y la limitación de su alcance, adecuado para espacios personales, pero no para largas distancias. Además, requieren medidas de seguridad adecuadas para proteger la información transmitida.

Las WPAN son fundamentales para la conectividad personal y las aplicaciones de IoT ya que su capacidad para conectar dispositivos de manera eficiente y su bajo consumo energético las hacen ideales para el ámbito del "Smart Metering" y la automatización del hogar, mejorando la gestión energética y la comodidad en el entorno personal.

Entre las tecnologías más comunes de WPAN se encuentran las siguientes:

¹ El "smart metering" o "medición inteligente" se refiere a la tecnología que utiliza medidores electrónicos avanzados para registrar y comunicar de manera automática el consumo de servicios públicos como electricidad, agua o gas. Estos dispositivos recopilan datos en tiempo real o casi real y los envían a los proveedores de servicios y a los consumidores. Esto permite una gestión más eficiente del suministro, facilita la facturación precisa y ofrece a los usuarios información detallada sobre su consumo, lo que puede ayudar a mejorar la eficiencia energética y reducir costos.

2.1.1. Bluetooth*

Bluetooth® es el nombre comercial que se le ha dado a la tecnología de comunicaciones de área personal (WPAN) definida en IEEE 802.15.1, estándar donde se especifican los requisitos a nivel de acceso al medio. Esta tecnología fue creada por el Bluetooth Special Interest Group (SIG) con el objetivo de intercambiar información de voz y datos entre diferentes dispositivos cercanos, utilizando para ello una de las bandas ISM (Industrial, Scientific and Medical) no licenciadas, la banda de 2,4GHz. Aunque esta tecnología se encuentra liderando en la industria de accesorios para teléfonos móviles, en los últimos años se ha estado posicionando como tecnología de bajo consumo para dispositivos wearables² además de equipos destinados a hogares o edificios inteligentes, debido a que en los últimos años se presentaron las nuevas características de bajo consumo con la versión de Bluetooth 4.0 o Bluetooth Low Energy (BLE).

La versión más reciente de esta tecnología es Bluetooth 5, cuyas características y especificaciones están enfocadas para habilitar la interconexión entre dispositivos del Internet de las Cosas (IoT, Internet of Things). Las versiones anteriores no conseguían despertar el interés para ser utilizado en dispositivos IoT debido a su alto consumo que provocaba que fuese inviable para los pequeños sistemas embebidos³ que se utilizan en este ámbito.

Las principales ventajas que ofrece Bluetooth 5 respecto a su predecesor, Bluetooth v4.2, es que puede llegar a tener un alcance hasta 4 veces superior, conseguir velocidades 2 veces más elevadas además de permitir que la longitud de los mensajes de señalización sean hasta 8 veces más largos por lo que hay mayor capacidad para enviar datos que en las versiones anteriores del protocolo.

En las versiones de Bluetooth v4.0/4.1 el formato de los paquetes de baja energía es el mismo, teniendo un tamaño de payload⁴ de hasta 27 octetos, al igual que la tasa de la modulación, de 1Ms/s, lo que significa que un transmisor tiene la capacidad de transmitir mil símbolos en un segundo. Con estas características se podría llegar a alcanzar un throughput⁵ de 0.305Mb/s. En la siguiente versión, Bluetooth v4.2, la tasa de modulación se mantiene igual que en las versiones anteriores, pero el formato de los paquetes es diferente, dejando un tamaño de 251 octetos para el payload del paquete. Esto da una capacidad de throughput de 0.803Mb/s.

Sin embargo, para la versión de Bluetooth 5, la tasa de modulación tiene dos opciones. La primera es de 1Ms/s, igual que en las versiones anteriores de Bluetooth v4.0/v4.1/v4.2. La segunda opción permite alcanzar una tasa de 2Ms/s en la modulación, por lo que añadiendo el formato de paquete que tiene la versión 4.2 puede llegar a alcanzar un throughput de 1.4Mb/s. En

² Los dispositivos wearables, o dispositivos portátiles, son una categoría de tecnología electrónica que se puede usar cómodamente en el cuerpo. Estos dispositivos incluyen relojes inteligentes, pulseras de actividad, gafas inteligentes, ropa inteligente y otros accesorios equipados con sensores y capacidades de conectividad. Los wearables permiten monitorear y registrar diversos datos, como actividad física, frecuencia cardíaca, sueño y otros parámetros de salud, además de ofrecer funciones como notificaciones, GPS y pagos móviles.

³ Los sistemas embebidos son sistemas de computación especializados que integran hardware y software para realizar funciones específicas dentro de dispositivos más grandes, como electrodomésticos, automóviles y equipos médicos, optimizando eficiencia y confiabilidad.

⁴ El payload, en el contexto de redes y comunicaciones, se refiere a la parte de los datos transmitidos que contiene la información útil o el contenido real que se desea enviar. Excluye los encabezados, metadatos y otros datos adicionales necesarios para la transmisión y el control. En otras palabras, es la carga útil de un paquete de datos, la información efectiva que se quiere comunicar.

⁵ El throughput, o rendimiento, es la cantidad de datos que se pueden transmitir exitosamente a través de un sistema o red en un período de tiempo determinado. Se mide comúnmente en bits por segundo (bps) o en paquetes por segundo (pps) y es un indicador clave de la capacidad y eficiencia de un sistema de comunicación o procesamiento de datos.

la Figura 2 se muestra una comparación entre las velocidades que se alcanzan en cada una de las versiones de Bluetooth.

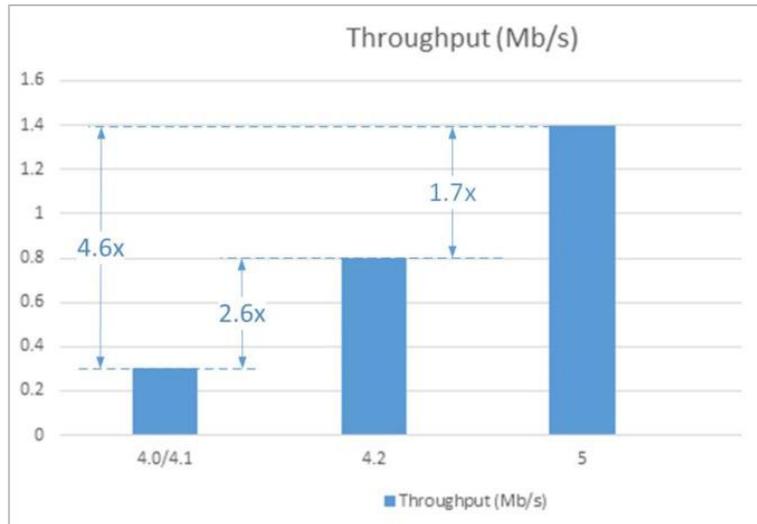


Figura 2: Velocidad en las distintas versiones de Bluetooth. (Fuente: ITI)

La tecnología Bluetooth contempla la pila completa de protocolos, tal y como se puede ver en la Figura 3. Bluetooth 5 añade dos nuevas variantes del nivel físico (PHY) a las especificaciones utilizadas en Bluetooth v4.0. Cada una de estas variantes tiene sus propias características pensadas para un objetivo concreto. Cada una de estas variantes de la capa física tiene un nombre propio: LE 1M, LE 2M y LE Coded.

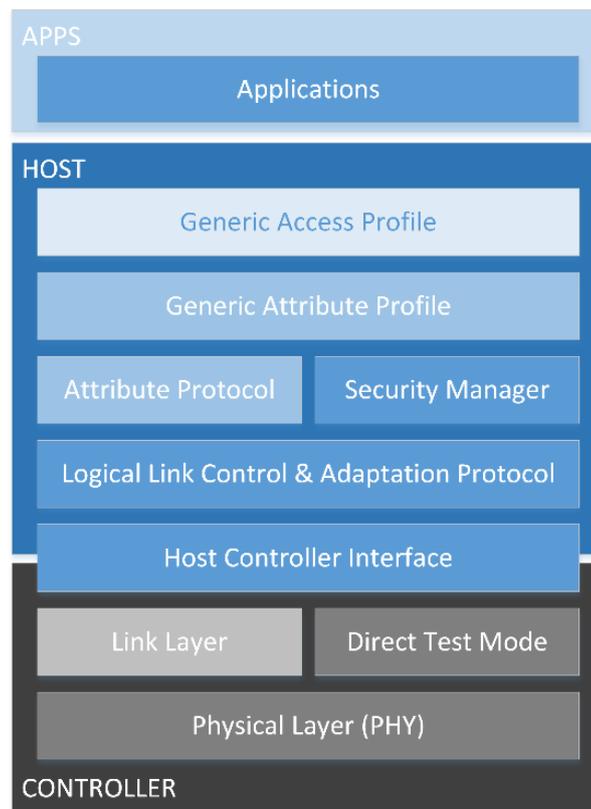


Figura 3: Pila de protocolos para Bluetooth Low Energy. (Fuente: ITI)

LE 1M es la capa física utilizada en Bluetooth v4.0. En esta se utiliza una modulación por desplazamiento de frecuencia gaussiana (GFSK)⁶ y tiene una tasa de símbolo de 1 mega símbolo por segundo (Ms/s). Este modo de funcionamiento sigue estando disponible en la versión 5 de las especificaciones debido a la compatibilidad con versiones anteriores, por lo que su implementación es obligatoria.

El modo LE 2M es el que permite a la capa física operar a 2Ms/s para conseguir así una mayor tasa de transferencia de datos que la versión LE 1M especificada en Bluetooth v4.0.

La variante más reciente de la capa física, conocida como LE Coded, permite extender el rango de cobertura a casi cuatro veces el de Bluetooth 4, consiguiendo esto sin incrementar la potencia de transmisión requerida. El rango de cobertura que se puede alcanzar con dispositivos que utilicen Bluetooth 4 está en torno a los 350-500 metros, según los experimentos mostrados en este documento, ya que en las especificaciones técnicas la distancia es de 100 metros. El interés que tiene aumentar el alcance de esta tecnología está motivado por las ventajas que ofrece en ámbitos como Smart Home, siendo este uno de los objetivos que se buscan con Bluetooth 5. Para que se consiga aumentar el alcance de estos dispositivos, Bluetooth 5 añade un sistema de corrección de errores que permite reducir la tasa de errores respecto a Bluetooth 4, que solo implementa el sistema de detección de errores.

En la especificación se detalla que estos equipos disponen de un controlador, llamado Host Controller Interface⁷ (HCI), que permite ejecutar un proceso que permita seleccionar el modo de funcionamiento del nivel físico que se desee en cualquier instante de tiempo, lo que le permite alcanzar tasa de 2Ms/s o mayor alcance según lo requiera la aplicación. La Tabla 2 presenta las principales métricas asociadas a las capacidades de Bluetooth 5.

| | LE 1M | LE Coded (S=2) | LE Coded (S=8) | LE 2M |
|---|-------------|----------------|----------------|----------|
| Tasa de símbolos (Ms/s) | 1 | 1 | 1 | 2 |
| Tasa de datos (Mbit/s) | 1 | 0,5 | 0,125 | 2 |
| Detección de errores | CRC | CRC | CRC | CRC |
| Corrección de errores | No | FEC | FEC | No |
| Multiplicador del rango (valor aprox.) | 1 | 2 | 4 | 0.8 |
| Requerimientos Bluetooth 5 | Obligatorio | Opcional | Opcional | Opcional |

Tabla 2: Comparativa de la capa PHY en Bluetooth 5. (Fuente: ITI)

⁶ La Frecuencia Gaussiana (GFSK) es una técnica de modulación digital que utiliza una forma de onda gaussiana para suavizar las transiciones de frecuencia, reduciendo el ancho de banda y minimizando interferencias. Es empleada en comunicaciones inalámbricas como Bluetooth por su eficiencia espectral y robustez frente al ruido.

⁷ El Host Controller Interface (HCI) es una especificación que define una interfaz estándar para la comunicación entre el host de un sistema y su controlador de hardware, como en los sistemas Bluetooth. HCI facilita el intercambio de comandos, eventos y datos entre el software del host y el controlador, permitiendo una gestión eficiente de dispositivos y conexiones.

En la Tabla 3 se muestra una comparativa-resumen sobre las principales características que tienen cada una de las versiones de Bluetooth, desde la clásica hasta la última versión.

| Características | Bluetooth Clásico | Bluetooth 4.x | Bluetooth 5 |
|---|---------------------|---------------------|---------------------|
| Frecuencia radio (MHz) | 2400 – 2483.5 | 2400 – 2483.5 | 2400 – 2483.5 |
| Distancia/rango (metros) | Hasta 100 | Hasta 100 | Hasta 200 |
| Técnica Acceso al Medio | Salto en frecuencia | Salto en frecuencia | Salto en frecuencia |
| Tasa de transferencia nominal (Mb/s) | 1-3 | 1 | 2 |
| Latencia (ms) | < 100 | < 6 | < 3 |
| Topología de red | Piconet, scatternet | Star-bus, mesh | Star-bus, mesh |
| Soluciones multi-salto | Scatternet | Si | Si |
| Tamaño del mensaje | Hasta 358 bytes | 31 bytes | 255 bytes |

Tabla 3: Comparación técnica de las diferentes versiones Bluetooth. (Fuente: ITI)

2.1.2. Estándar IEEE 802.15.4e

El estándar IEEE 802.15.4e es una extensión del IEEE 802.15.4⁸, diseñado para mejorar las comunicaciones en redes WPAN, especialmente en aplicaciones de IoT y automatización. Este estándar se enfoca en optimizar la eficiencia energética, la fiabilidad y la escalabilidad de las redes, características fundamentales en entornos como el Smart Metering. Con mejoras significativas en la eficiencia energética, permite a los dispositivos operar durante más tiempo sin necesidad de reemplazo de baterías, reduciendo los costos de mantenimiento.

Además, soporta topologías de red avanzadas, como configuraciones en malla y jerárquicas, que aumentan la cobertura y la fiabilidad de la comunicación. Los mecanismos de priorización de tráfico aseguran que los datos críticos se transmitan con prioridad, y el soporte para operaciones determinísticas permite tiempos de respuesta predecibles, lo cual es vital en aplicaciones industriales y de monitoreo en tiempo real.

Una de las características más destacadas es la introducción de TSCH (Time-Slotted Channel Hopping), una técnica que mejora la robustez de la red al minimizar interferencias y favorecer la coexistencia con otras redes. En el contexto de Smart Metering, IEEE 802.15.4e ofrece confiabilidad en las comunicaciones, mayor eficiencia energética y capacidad para manejar un gran número de dispositivos, haciéndolo ideal para entornos urbanos densos. Esta flexibilidad permite su uso en diversas configuraciones, desde hogares hasta grandes instalaciones industriales, convirtiéndolo en una opción atractiva para la implementación de redes de medidores inteligentes y dispositivos de automatización.

⁸ El estándar **IEEE 802.15.4** define las especificaciones para redes inalámbricas de área personal de baja tasa (LR-WPAN). Está diseñado para aplicaciones que requieren bajo consumo de energía, baja velocidad de datos y comunicación a corta distancia, como las utilizadas en domótica, sensores y dispositivos IoT. Es la base para protocolos como Zigbee y utiliza técnicas como la modulación directa por secuencia para mejorar la fiabilidad de la comunicación.

En la primera versión de este estándar, publicada en 2003, únicamente estaban definidas dos opciones para la capa física (PHY) además de emplear un control de acceso al medio (MAC) simple y efectivo. El objetivo de este protocolo es el de permitir el establecimiento de comunicaciones en redes de área personal empleando para ello una baja tasa de transferencia de datos, también denominadas Low Rate – Wireless Personal Area Networks (LR-WPAN). El uso de estas redes permite implementar comunicaciones de bajo costo, lo que hace que esta tecnología sea ideal para dispositivos con restricciones en energía y capacidad de procesamiento, facilitando la comunicación entre ellos.

La capa física establece los mecanismos necesarios para gestionar el transceptor de radio, incluyendo la detección de energía, la evaluación de la calidad del enlace, la selección del canal de comunicación y la evaluación del uso del canal. Además, se encarga de la transmisión y recepción de mensajes en el nivel físico de la pila de protocolos.

IEEE 802.15.4 contempla diferentes bandas de frecuencia que pueden ser utilizados en el establecimiento de las comunicaciones, así como diferentes alternativas en cuanto a modulación. Aunque hay diferentes alternativas, las bandas de frecuencia más utilizadas en esta tecnología son las mostradas en la Tabla 4:

| Banda de frecuencia | Nº Canales | Ancho de banda |
|---------------------|------------|----------------|
| 2450 MHz | 16 canales | 5 MHz |
| 915 MHz | 10 canales | 2 MHz |
| 868 MHz | 1 canal | 2 MHz |

Tabla 4: Bandas más usadas en IEEE 802.15.4. (Fuente: ITI)

En la Figura 4 se muestran las bandas de frecuencia utilizadas en IEEE 802.15.4.

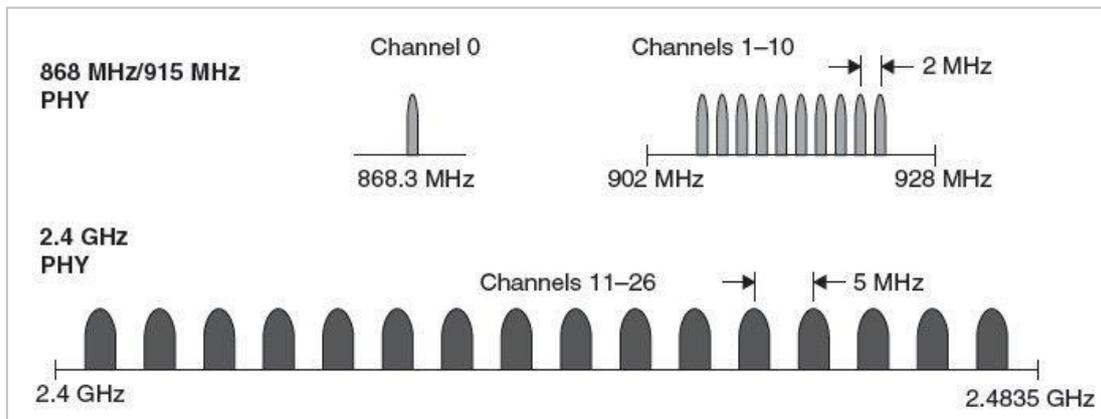


Figura 4: Principales bandas de frecuencia en IEEE 802.15.4. (Fuente: [2])

Por otro lado, la capa MAC es la encargada de administrar los mensajes de señalización y balizamiento, los accesos a los recursos radio o canales, la administración de los recursos temporales que necesiten garantía de disponibilidad, validar las tramas según la norma, retransmitir los mensajes de acuse de recibo, así como de la asociación y disociación de la red. Además, en este nivel es posible implementar mecanismos de encriptación para añadir ciertos niveles de seguridad al intercambio de información.

Inicialmente solo estaban definidos dos métodos de acceso al medio que permitían señalar los límites en los que establecer la comunicación utilizando para ello unas balizas que delimitaban las zonas de transmisión. Sin embargo, estas primeras versiones del protocolo presentaban numerosas limitaciones cuando se introducía la tecnología en escenarios con interferencias más agresivas, como pueden ser los escenarios industriales, entre otros. Algunas de estas limitaciones son las que se presentan a continuación:

- Retardo de los mensajes no acotado.
- Fiabilidad limitada en la comunicación.
- No se proporciona protección contra interferencias o pérdidas de señal.
- El consumo de los nodos retransmisores alto, ya que muchas aplicaciones necesitaban mantener los dispositivos retransmisores activos todo el tiempo.

Por estos y otros motivos, las versiones iniciales del estándar IEEE 802.15.4 no eran particularmente efectivas en situaciones donde se requerían alta fiabilidad y robustez. Es por esto, por lo que en 2012 se lanza una enmienda que soluciona todos estos problemas, aportando mayor fiabilidad, robustez y flexibilidad a la tecnología. Esta enmienda, llamada IEEE 802.15.4e, está planteada para poder trabajar en entornos condicionados por las interferencias, ya que posee algunos mecanismos que permiten variar la frecuencia utilizada para establecer la comunicación, consiguiendo así que las interferencias que afectan a unos canales no afecten tanto al funcionamiento normal de las comunicaciones.

Algunas de las mejoras que añade esta modificación del estándar tienen que ver sobre la mejora de los mecanismos de bajo consumo energético, de forma que los dispositivos puedan trabajar con un ciclo de trabajo muy bajo, alrededor del 1%. También se añaden mejoras a la hora de describir y formar las tramas de comunicación, mejorando la información que estas transportan, utilizando para ello el encapsulado de la información relevante de los Information Elements (IE) para la formación y mantenimiento de las redes inalámbricas. También se introducen mejoras en cuanto a los mecanismos de señalización, introduciendo las Enhanced Beacons (EB), que proveen una mayor flexibilidad y mantienen la compatibilidad con las balizas originales.

La modificación de la norma introduce importantes mejoras en los mecanismos de acceso al medio, permitiendo diferentes métodos de conexión a la red adaptados a las necesidades específicas de cada aplicación. Estos nuevos mecanismos superan las limitaciones de los métodos previamente establecidos. Entre las principales adiciones de la enmienda se encuentran DSME⁹, LLDN¹⁰ y TSCH¹¹:

Este último mecanismo de acceso al medio, el modo TSCH, consigue reducir considerablemente el ciclo de trabajo de los dispositivos, mejorando con ello el consumo energético, gracias a la división del espacio temporal en slots que irán sincronizados con un nodo coordinador, lo que permite que los dispositivos solo consuman energía en aquellas fracciones de tiempo en las que necesitan transmitir o recibir información. Además, añade un mecanismo de salto en frecuencia que, por un lado, aporta una protección frente a interferencias y, por otro lado, permite la reutilización del espectro radioeléctrico añadiendo más capacidad para transmitir información, gracias a la diversidad en frecuencia.

En la Figura 5 se puede ver como se distribuyen los recursos radio, tanto en el plano temporal como en el frecuencial. Temporalmente un slotframe se irá repitiendo de forma periódica en el tiempo. Este slotframe estará formado por un determinado número de slots temporales en los que poder establecer la comunicación, denominados en la figura como ‘TSX’. El número de slots que forman el slotframe determinará la frecuencia con la que se repiten estos slots, y por tanto será un factor limitante en el ciclo de trabajo del dispositivo. Por otra parte, en el plano frecuencial se

⁹ Está diseñado para aplicaciones que requieren alta disponibilidad, eficiencia, escalabilidad y robustez. Combina el acceso aleatorio con una división temporal de los recursos de radio, y ofrece dos opciones distintas para la diversidad de canales.

¹⁰ Desarrollado para aplicaciones que necesitan baja latencia, como robots y sistemas de control de grúas.

¹¹ Especialmente adaptado para entornos industriales, este modo optimiza el consumo energético y proporciona alta resistencia a interferencias, asegurando robustez en la comunicación.

dispone de diferentes canales en los que poder establecer comunicación, de tal forma que cada vez que se repita un slot se utilizará un canal diferente de manera secuencial, permitiendo reducir así el impacto de las interferencias en la comunicación. La frecuencia de transmisión vendrá determinada por la siguiente ecuación:

$$f = F\{(ASN + ch_{offset}) \cdot mod C\} \quad (1)$$

f: canal físico de transmisión;

ASN: valor del slot absoluto (cantidad de slot transcurridos desde el despliegue);

ch_{offset}: permite usar diferentes canales dentro de un mismo slot;

mod C: es el modulo de número total de canales empleados;

F {...}: se encarga de seleccionar un canal de una secuencia predefinida;

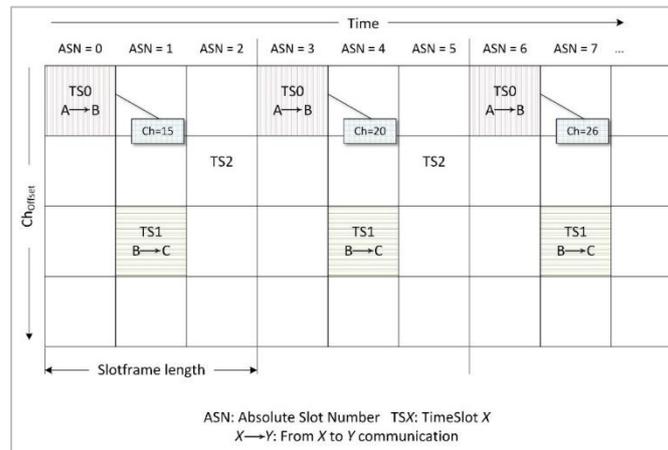


Figura 5: Distribución temporal y frecuencial utilizada en TSCH. (Fuente: [2])

Debido a todas estas características, el estándar IEEE 802.15.4 es uno de los estándares más utilizado para la formación de WSNs, consiguiendo alcances de hasta 10 metros con una tasa de transferencia de 250 Kbps. La utilización de este estándar, junto a otros protocolos como RPL (Routing Protocol for Low-Power and Lossy Networks), que resuelve la parte de encaminamiento para redes mallas, y el protocolo 6LoWPAN, que posibilita el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4, hacen que esta tecnología sea una de las principales soluciones para construir WSN, que sirvan de tecnología habilitadora para aplicaciones relacionados con IoT.

Es por ello por lo que numerosas tecnologías ha construido soluciones propias, con una definición completa de la pila de protocolos, sobre las definiciones originales que aporta el estándar IEEE 802.15.4. Ejemplos de estas tecnologías son el caso de ZigBee, inicialmente pensada para entornos de domótica y Smart Home; WirelessHART, ISA100.11a y WIA-PA, desarrollados para trabajar en entornos industriales; y el caso más reciente de la tecnología Thread, que se basa en las especificaciones de IEEE 802.15.4 para formar redes para aplicaciones IoT en escenarios de Smart Home.

2.1.9. Tabla comparativa de tecnologías de área personal

En la Tabla 5 se resumen algunas de las características más relevantes en cuanto a las limitaciones de cada una de las tecnologías mencionadas. Se han omitido las características de IEEE 802.15.4 debido a que forman parte de otras tecnologías completas como ZigBee o Thread, que ofrecen un conjunto más amplio de funcionalidades y aplicaciones.

| Especificaciones | Bluetooth | Bluetooth 5 | ZigBee | Thread | Z-Wave |
|-------------------------------|----------------|----------------|----------|----------|-------------|
| Tipo de redes | Star-bus, mesh | Star-bus, mesh | Mesh | Mesh | Mesh |
| Distancia (m) | Hasta 100 | Hasta 200 | 10-20 | 20 - 30 | 100 con LOS |
| Banda freq. | 2,4 GHz | 2,4 GHz | 2,4 GHz | 2,4 GHz | 915 MHz |
| Throughput | 1-3 Mb/s | 2 Mb/s | 110 kbps | 250 kbps | 40 kbps |
| Consumo | Alto | Bajo | Bajo | Bajo | Bajo |
| Conectividad IP nativa | No | No | No | Si | Si |

Tabla 5: Comparación de especificaciones WPAN. (Fuente: ITI)

2.2. Tecnologías WLAN: Redes inalámbricas de área local.

Las redes WLAN son sistemas diseñados para ofrecer conectividad inalámbrica en un área geográfica limitada, típicamente dentro de un edificio, campus o entorno similar. Estas redes pueden cubrir distancias que van desde pocos hasta varios cientos de metros, dependiendo de la configuración y los dispositivos utilizados.

Una característica común de las WLAN es el uso de bandas de frecuencia sin licencia, como las frecuencias de 2.4 GHz y 5 GHz, que están disponibles para su uso público. Estas bandas permiten desplegar tecnologías que operan dentro de esta categoría, como Wi-Fi, de manera eficiente y sin la necesidad de adquirir licencias específicas para su uso.

Inicialmente, las WLAN fueron diseñadas para ofrecer altas tasas de transferencia de datos, lo que las hizo ideales para aplicaciones que requerían la transmisión rápida de información, como transferencias de archivos grandes o streaming de contenido multimedia. Sin embargo, con el avance de la tecnología, para conseguir enviar datos con tasas de transferencia más moderadas han surgido versiones de baja potencia que lo permiten, y, además, proporcionan consumos energéticos reducidos.

Estas versiones de baja potencia son especialmente útiles en aplicaciones de IoT (Internet de las Cosas) y dispositivos móviles que operan con batería, donde la duración de esta es crítica. La capacidad de transmitir datos de manera eficiente y con un consumo energético mínimo es crucial para extender la vida útil de la batería y garantizar un funcionamiento continuo y confiable de los dispositivos conectados.

2.2.1. Tecnologías WiFi: 802.11a/b/g, 802.11n, 802.11ac y 802.11ax

El conjunto de estándares conocido como IEEE 802.11¹² está orientado a las comunicaciones inalámbricas en redes locales. Aunque "Wi-Fi" es una marca registrada por la Wi-Fi Alliance, esta tecnología se basa en las especificaciones de IEEE 802.11 con una evolución a lo largo del tiempo para ofrecer distintas versiones. Las versiones iniciales, aunque ahora obsoletas, fueron diseñadas para proporcionar conexiones inalámbricas de corto alcance en entornos domicilios u oficinas. Sin embargo, con el aumento en la cantidad de dispositivos y la necesidad de mayores rangos de velocidad, el estándar ha ido evolucionando significativamente.

Las versiones 802.11b y 802.11a consiguieron velocidades de hasta 11 Mbps y 54 Mbps, respectivamente. La versión 802.11b operaba en la banda de 2.4 GHz, mientras que 802.11a lo hacía en la banda de 5 GHz. Estas versiones fueron lanzadas casi al mismo tiempo, sin embargo, no eran compatibles entre ellas ya que la versión en la banda de 2.4 GHz tenía un alcance aproximado de 50 metros en interiores, mientras que la de 5 GHz tenía un alcance reducido a unos 10-20 metros.

El estándar 802.11g intentó combinar las ventajas de las versiones anteriores, operando en la banda de 2.4 GHz y alcanzando velocidades de hasta 54 Mbps, aprovechando los anchos de banda descritos en los estándares anteriores.

La versión 802.11n, se diseñó para enriquecer el rendimiento de las versiones anteriores, aumentando el número de anchos de banda soportados e introduciendo la tecnología MIMO¹³, lo que permite comunicaciones en ambas bandas (2.4 y 5 GHz).

Con la llegada del estándar 802.11ac, se implementó la tecnología de doble banda, lo que permite conexiones simultáneas tanto en la banda de 2.4 GHz como en la de 5 GHz. Esta versión es retro compatible con estándares anteriores y puede alcanzar velocidades de hasta 1300 Mbps en la banda de 5 GHz y 450 Mbps en la banda de 2.4 GHz.

El estándar más reciente, IEEE 802.11ax, presenta mejoras significativas en capacidad, cobertura y rendimiento en entornos con alta densidad de dispositivos, además de optimizar la eficiencia energética. Estas características lo hacen ideal para diversas aplicaciones, como hogares inteligentes y dispositivos IoT. Entre las mejoras notables de esta versión se incluyen: técnicas de modulación OFDMA, esquemas de modulación hasta 1024-QAM, MU-MIMO bidireccional y una mayor duración del símbolo para mejorar la eficiencia.

Recientemente, la Wi-Fi Alliance ha adoptado una nueva nomenclatura para las versiones del estándar con el fin de aclarar la confusión causada por los nombres anteriores. En la Tabla 6 se presentan las características principales de las versiones recientes de Wi-Fi. Las velocidades de transmisión indicadas en la tabla son aproximadas y pueden variar significativamente según la capacidad MIMO del sistema; con un mayor número de antenas, es posible alcanzar tasas de decenas de Gbps con Wi-Fi 6.

¹² IEEE 802.11 comprende una serie de especificaciones para redes locales inalámbricas (WLAN), creado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). Este conjunto de normas define los requisitos para la transmisión de datos utilizando tecnologías de radiofrecuencia dentro de las bandas de frecuencia de 2.4 GHz y 5 GHz.

¹³ Del inglés "Multiple Input Multiple Output", es una tecnología de comunicación que utiliza múltiples antenas tanto en el transmisor como en el receptor para mejorar la capacidad y la eficiencia de la transmisión de datos en WSN.

| Nuevos nombres | Estándar | Frecuencia | Velocidad teórica |
|----------------|----------|-------------|-------------------|
| Wi-Fi 6 | 802.11ax | 2.4 / 5 GHz | > 3.5 Gbps |
| Wi-Fi 5 | 802.11ac | 2.4 / 5 GHz | 1.3 Gbps |
| Wi-Fi 4 | 802.11n | 2.4 / 5 GHz | 600 Mbps |

Tabla 6: Comparación de especificaciones Wi-Fi. (Fuente: ITI)

2.2.2. WiFi HaLow (802.11ab)

WiFi HaLow, también conocido como IEEE 802.11ah, es una extensión del estándar IEEE 802.11 diseñada específicamente para aplicaciones de IoT (Internet of Things) y dispositivos de bajo consumo energético. Este estándar opera en bandas de frecuencias sin licencia por debajo de 1 GHz, lo que permite una mayor penetración de señal y un alcance extendido en comparación con las redes WiFi más tradicionales.

El principal objetivo de WiFi HaLow es proporcionar una conectividad robusta y de larga distancia, capaz de alcanzar velocidades de hasta 347 Mbps. Esta capacidad de transferencia de datos, aunque inferior a estándares WiFi de frecuencias más altas, es suficiente para muchas aplicaciones de IoT que requieren comunicaciones moderadas pero confiables.

Además de su alcance extendido y velocidad moderada, WiFi HaLow está diseñado para ser altamente eficiente en términos de consumo de energía. Esta característica lo hace ideal para dispositivos IoT que operan con baterías y tienen requisitos estrictos de eficiencia energética. Los medidores inteligentes, por ejemplo, podrían beneficiarse significativamente de esta tecnología, permitiendo una operación continua durante períodos prolongados sin la necesidad frecuente de reemplazo de baterías.

2.3. Tecnologías WNAN: Redes inalámbricas de vecindario.

Las Redes Inalámbricas de Vecindario (WNAN) representan una categoría de tecnologías diseñadas para proporcionar conectividad dentro de áreas urbanas compactas o comunidades específicas. Estas redes están optimizadas para cubrir distancias relativamente cortas, típicamente desde unos pocos cientos de metros hasta algunos kilómetros, lo que las hace ideales para implementaciones en vecindarios y zonas residenciales densamente pobladas. Su capacidad para ofrecer una cobertura localizada pero extensa es especialmente ventajosa en aplicaciones como el Smart Metering, donde dispositivos de medición inteligente necesitan comunicarse eficientemente dentro de una misma área geográfica.

En términos de eficiencia energética, las WNAN están diseñadas para consumir poca energía, lo cual es crucial para mantener operativos los dispositivos de manera continua durante períodos prolongados sin intervención humana frecuente. Además, muchas de estas redes pueden configurarse como redes en malla (mesh networks), permitiendo que cada dispositivo actúe como un nodo de la red, facilitando así la comunicación entre los diferentes dispositivos de medición y los puntos de recolección de datos.

Desde el punto de vista de la seguridad y la fiabilidad, las implementaciones de WNAN suelen incluir protocolos robustos para proteger la integridad de los datos transmitidos, garantizando así que la información recogida por los medidores inteligentes sea precisa y segura. Estas redes también son compatibles con el ecosistema más amplio de Internet de las Cosas (IoT), lo que permite integrar fácilmente los medidores inteligentes con otros sistemas de gestión de servicios urbanos, optimizando así la eficiencia operativa y mejorando la calidad de los servicios ofrecidos a los usuarios finales.

Las redes inalámbricas de vecindario ofrecen una solución versátil y escalable para la conectividad dentro de áreas urbanas y residenciales, siendo fundamentales para aplicaciones críticas como el Smart Metering. Su capacidad para proporcionar cobertura localizada con bajo consumo de energía y alta fiabilidad las convierte en una opción ideal para despliegues que requieren comunicación eficiente y continua entre dispositivos distribuidos en un entorno geográfico específico.

Entre las tecnologías más comunes de WNAN se encuentran las siguientes:

2.3.1. Wi-SUN

Wi-SUN (Wireless Smart Utility Network) es un nuevo estándar de comunicaciones diseñado para interconectar dispositivos en aplicaciones de Smart Grid, basándose en los niveles físicos y de control de acceso al medio definidos en IEEE 802.15.4g. Esta tecnología emergente se destaca por su capacidad para soportar comunicaciones a largo alcance y bajo consumo de energía, siendo especialmente adecuada para entornos de servicios públicos como el suministro de electricidad, agua y gas. Detrás de Wi-SUN hay una amplia colaboración de empresas bajo la Wi-SUN Alliance, que promueve su adopción en aplicaciones residenciales y urbanas como el Smart Grid y las Smart Cities. Entre las aplicaciones destacadas se incluyen la medición avanzada, automatización distribuida, iluminación municipal, aparcamiento inteligente y sensorización medioambiental, entre otras. Una de las características clave de Wi-SUN es su capacidad para operar en bandas de frecuencia sin licencia, lo que facilita su implementación y reduce los costos asociados con la adquisición de espectro de frecuencia. Esto es especialmente beneficioso para despliegues a gran escala en entornos urbanos y rurales, donde la conectividad extendida y la penetración a través de estructuras físicas son cruciales para garantizar una cobertura confiable.

Wi-SUN utiliza el estándar IEEE 802.15.4g, que define una capa física y un protocolo de acceso al medio (MAC) optimizados para aplicaciones de redes de servicios públicos. Esta especificación permite que los dispositivos Wi-SUN establezcan comunicaciones robustas en entornos desafiantes, como áreas urbanas densamente pobladas o lugares con obstáculos físicos.

En términos de aplicación en Smart Metering, Wi-SUN permite la conexión de medidores inteligentes a una red de infraestructura compartida, lo que facilita la recopilación remota de datos de consumo y la gestión eficiente de servicios. Esto no solo mejora la precisión de la medición y la facturación, sino que también habilita la implementación de servicios avanzados de gestión de energía y agua basados en datos en tiempo real.

Además, Wi-SUN es compatible con tecnologías de red en malla (mesh networking), lo que significa que los dispositivos pueden comunicarse entre sí y con puntos de acceso centrales redundantes, mejorando la redundancia y la confiabilidad del sistema. Esta capacidad de autoorganización y autoconfiguración también simplifica la expansión y mantenimiento de la red a medida que crece el número de dispositivos conectados.

Las especificaciones de la capa física de la tecnología Wi-SUN están basadas en IEEE 802.15.4g¹⁴, mientras que las especificaciones de la capa MAC implementan funciones definidas en IEEE 802.15.4e. Wi-SUN además incluye las especificaciones del resto de niveles de la pila completa de protocolos con el objetivo de ser una tecnología completa que de soporte a diferentes aplicaciones.

¹⁴ El estándar IEEE 802.15.4g especifica las características físicas y las técnicas de modulación para redes de área personal inalámbricas (WPAN) enfocadas en aplicaciones de redes de servicios públicos y Smart Grid. Diseñado para soportar comunicaciones a largo alcance y bajo consumo de energía, IEEE 802.15.4g optimiza la transmisión de datos en entornos desafiantes, como áreas urbanas densamente pobladas y entornos rurales, facilitando la conectividad confiable y eficiente para dispositivos como medidores inteligentes y otros dispositivos IoT en infraestructuras críticas.

En el estándar 802.15.4g se definen tres modelos diferentes de capa física: MR-FSK, MR-OFDM y MR-OQPSK. El primero de estos modos se ha escogido como modulación FSK (Frequency Shift Keying) debido a que está ampliamente extendido en multitud de sistemas en los Estados Unidos, mientras que MR-OFDM permite alcanzar mayores tasas de transferencia de datos y MR-OQPSK es similar al nivel físico definido en las versiones originales de 802.15.4, utilizada para formar redes de sensores inalámbricos.

- MR-FSK soporta tasas de transferencia de datos desde 2.4 kbps hasta los 200 kbps. Este valor dependerá de la banda de frecuencias utilizada, ya que en cada región hay una serie de frecuencias permitidas, coincidiendo únicamente la banda de 2.4 GHz en todo el mundo. Además, dado que la modulación FSK está ampliamente extendida en muchas otras aplicaciones, el estándar permite definir modos propietarios a nivel físico. A continuación, se muestran las bandas de frecuencia permitidas en el modo MR-FSK.
 - 169.4 - 169.475 MHz, 863 - 870 MHz (Europa)
 - 450 – 470 MHz, 896 – 901 MHz, 901 – 902 MHz, 928 MHz, 960 MHz, 1427 – 1518 MHz (Bandas licenciadas de US)
 - 902 – 928 MHz (Bandas no licenciadas de US)
 - 470 – 510 MHz, 779 – 787 MHz (China)
 - 917 - 923.5 MHz (Corea)
 - 920 – 928 MHz, 950-958 (Japón)
 - 2400 - 2483.5 MHz (Mundial)

- MR-OFDM soporta tasas de transferencia desde 50 kbps hasta 800 kbps. Las diferentes tasas de transferencia se pueden conseguir utilizando diferentes parámetros de configuración de la capa física MR-OFDM, teniendo hasta cuatro modos de funcionamiento y diferentes métodos de modulación como BPSK, QPSK, y QAM. Las bandas de frecuencia definidas para MR-OFDM son las siguientes:
 - 863-870 MHz (Europa)
 - 902-928 MHz (Bandas no licenciadas de US)
 - 470-510 MHz, 779-787 MHz (China)
 - 917-923.5 MHz (Corea)
 - 920-928 MHz, 950-958 MHz (Japón)
 - 2400-2483.5 (Mundial)

- MR-OQPSK está definido para mantener la compatibilidad con los equipos que utilizan la capa física original de IEEE 802.15.4 y operan en las bandas de 780 MHz, 915 MHz y 2450 MHz. Las tasas de transferencia soportadas en este modo están comprendidas entre los 20 kbps y los 250 kbps.

Por otro lado, la capa MAC ha sido redefinida para mejorar su funcionamiento respecto a su definición original de IEEE 802.15.4. Tanto en el estándar IEEE 802.11-2007 como en IEEE 802.15.3-2003 se definen los mecanismos de IEs (Information elements), que además ha sido adoptado en la versión de IEEE 802.15.4e, que facilita el intercambio de datos con el resto de las capas de pila de los protocolos. En IEEE 802.15.4e hay definidos dos tipos de IEs, los elementos de cabecera (MHR – MAC Header) y los de la carga útil que van adjuntos a los datos que se pasan a los niveles superiores. Este mecanismo es realmente útil para que los dispositivos se puedan comunicar y cambiar sus capacidades de la capa física, permitiendo que trabajen de una manera más eficiente. Sin las funcionalidades que aporta la utilización de los IEs, las capacidades de la capa física no se podrían intercambiar eficientemente.

2.3.2. JupiterMesh

La ZigBee Alliance se ha destacado en el desarrollo de tecnologías de comunicación que buscan extender el alcance más allá de lo que ofrecen las redes ZigBee tradicionales. En este contexto, JupiterMesh emerge como una solución avanzada capaz de implementar redes malladas inalámbricas de bajo consumo energético, especialmente diseñadas para aplicaciones industriales de IoT. La robustez en la comunicación es una prioridad fundamental para JupiterMesh, permitiendo configurar redes con diversas tasas de transferencia y alcances extendidos, adaptándose así a aplicaciones en Neighborhood Area Networks y soluciones orientadas a Smart Grid y Smart Cities.

Similar a ZigBee, JupiterMesh se basa en estándares abiertos para proporcionar una solución integral. A nivel físico y de acceso al medio, utiliza las especificaciones de IEEE 802.15.4e y 802.15.4g, incorporando funciones y mecanismos de robustez como el modo TSCH¹⁵ definido en estos estándares. Además, implementa el protocolo 6LoWPAN para la interoperabilidad con redes IPv6 y el protocolo de encaminamiento dinámico RPL¹⁶, facilitando una gestión eficiente y flexible de la red en entornos dinámicos y variados.

Por estar basada en el estándar IEEE 802.15.4g, JupiterMesh tiene la capacidad de operar en varios modos de capa física, permitiéndole trabajar en frecuencias por debajo de 1 GHz. Esta característica posibilita alcanzar mayores distancias y ofrecer diversas tasas de transferencia de datos, lo cual la diferencia significativamente del protocolo ZigBee definido por la ZigBee Alliance. Esta flexibilidad y eficiencia hacen de JupiterMesh una opción atractiva para despliegues que requieren conectividad robusta y escalable en entornos industriales y urbanos avanzados.

2.3.3. Wireless M-Bus

Wireless M-Bus, también conocido como Wireless Meter-Bus, es un estándar europeo (EN 13757-4) diseñado para facilitar la comunicación inalámbrica entre medidores, registros de datos, concentradores y pasarelas en sistemas de contadores inteligentes. Desarrollado para satisfacer las necesidades del mercado europeo, este estándar proporciona una solución eficaz para la interconexión y la lectura remota de contadores en diversos sectores como electricidad, gas, agua y calefacción. Derivado del estándar M-Bus original, que utiliza comunicaciones cableadas, Wireless M-Bus aprovecha las frecuencias de radio de 868 MHz, 433 MHz y 169 MHz para establecer conexiones inalámbricas robustas y confiables entre dispositivos.

Wireless M-Bus opera en diferentes modos de frecuencia según las necesidades específicas de las aplicaciones, cada modo hace uso de las diferentes frecuencias para enviar la información de los contadores, tal y como se muestra en la Tabla 7. Utiliza el modo estacionario (S) para la transmisión esporádica de datos por parte de los medidores, el modo de transmisión frecuente (T) para envíos múltiples diarios y el modo compacto (C) para tasas de transferencia más altas que el

¹⁵ El modo TSCH (Time-Slotted Channel Hopping) es una técnica de acceso al medio utilizada en redes inalámbricas, especialmente en estándares como IEEE 802.15.4e. Este modo divide el tiempo en ranuras de tiempo predefinidas y asigna canales de frecuencia específicos a cada ranura. Los dispositivos en la red sincronizan sus operaciones de transmisión y recepción según este calendario de ranuras y canales, lo que permite una comunicación más eficiente y predecible al reducir las colisiones y minimizar las interferencias. Además, TSCH mejora la robustez de la red al ofrecer una mayor inmunidad frente a las condiciones adversas del canal y al soportar redes de área personal inalámbricas (WPAN) en entornos industriales y otros entornos críticos.

¹⁶ RPL del inglés Routing Protocol for Low-Power and Lossy Networks, es un estándar diseñado para redes de baja potencia y pérdida de paquetes, como las redes de sensores y las redes de área personal inalámbricas (WPAN). RPL facilita la creación automática de rutas eficientes entre los nodos de la red, optimizando el consumo de energía y adaptándose a cambios en la topología de la red. Utiliza métricas de ruta como el número de saltos y la calidad del enlace para tomar decisiones de encaminamiento, lo que permite una comunicación confiable y eficiente en entornos dinámicos y heterogéneos de IoT y Smart Grid.

modo T. Además, emplea el modo de banda estrecha (N) para lograr un mayor alcance y el modo de recepción frecuente (R) para la lectura de múltiples contadores en diferentes canales de frecuencia.

Esta tecnología se caracteriza por su simplicidad y eficiencia en la construcción de redes en topología básica de estrella, facilitando la comunicación directa entre los medidores y un concentrador central. Incorpora medidas de seguridad robustas para proteger la integridad de los datos transmitidos y garantizar un funcionamiento confiable incluso en entornos desafiantes y con restricciones de energía. Aunque inicialmente no es compatible con IPv6 ni con redes malladas, Wireless M-Bus continúa evolucionando para mantenerse alineado con las demandas crecientes de conectividad y eficiencia en el sector de los servicios públicos y de gestión de recursos.

| Modo | Frecuencia (MHz) | Notas |
|-------------------------------|------------------|---|
| S (Stationary) | 868 | Los contadores envían datos pocas veces al día |
| T (Frequent Transmit) | 868 | Los contadores envían datos varias veces al día |
| C (Compact) | 868 | Mayor tasa de transferencia que el modo T |
| N (Narrowband) | 169 | Mayor alcance, sistema de banda estrecha |
| R (Frequent Receive) | 868 | El concentrador lee de múltiples contadores en diferentes canales de frecuencia |
| F (Frequent Tx and Rx) | 433 | Comunicaciones bidireccionales frecuentes |

Tabla 7: Modos de operación Wireless M-Bus. (Fuente: ITI)

Debido a las bandas de frecuencia en las que trabaja, de forma adicional a las especificaciones del estándar de Wireless M-Bus, son necesarias una serie de requerimientos específicos para cada región de Europa, lo que hace que sea más complicado obtener una arquitectura común y que aparezca una necesidad de que la tecnología sea flexible y modular.

Sus principales características que destaca, frente a otras tecnologías destinadas a aplicaciones IoT para contadores inteligentes, se centran en su relativa simplicidad. Está diseñada para construir una topología de red básica en estrella, utilizando las bandas de frecuencia sub-GHz que ofrecen un largo alcance. Además, aunque esto supone ventajas en otras tecnologías, Wireless M-Bus destaca entre sus características la sencillez de su pila de protocolos y el hecho de que no son compatibles con IPv6 ni redes malladas.

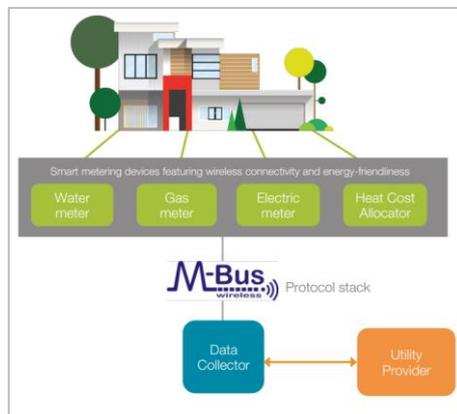


Figura 6: Ejemplo de arquitectura de un sistema de contadores basado en Wireless M-Bus. (Fuente: ITI)

En conclusión, Wireless M-Bus es una solución tecnológica clave para la gestión avanzada de contadores inteligentes en Europa, ofreciendo una plataforma robusta y estandarizada que optimiza la recopilación de datos, mejora la eficiencia operativa y promueve servicios más personalizados para los usuarios finales en una variedad de entornos residenciales, comerciales e industriales.

2.4. Tecnologías WWAN: Redes inalámbricas de área amplia.

Las Redes Inalámbricas de Área Amplia (WWAN) son un tipo de red que proporciona servicios de comunicación a larga distancia, cubriendo grandes áreas geográficas. Estas redes son fundamentales para la conectividad móvil y son especialmente relevantes en el ámbito del "Smart Metering" debido a su amplia cobertura y capacidad para conectar dispositivos en áreas remotas o de difícil acceso. A continuación, se detallan las características y beneficios de las WWAN en este contexto.

Las redes WWAN son esenciales para el "Smart Metering" debido a su capacidad para proporcionar conectividad en lugares donde otras tecnologías pueden no ser viables. Por ejemplo, en áreas rurales donde la infraestructura de telecomunicaciones es limitada, tecnologías como LTE-M y NB-IoT permiten la transmisión de datos de consumo energético, de agua o gas de manera eficiente. Esto asegura que los datos lleguen a los centros de monitoreo sin interrupciones, mejorando la gestión y el análisis del consumo en tiempo real.

Además de la amplia cobertura, las WWAN ofrecen fiabilidad y escalabilidad. Estas redes pueden soportar una alta densidad de dispositivos conectados, lo que es crucial en áreas urbanas donde la cantidad de medidores inteligentes puede ser considerable. La escalabilidad de las WWAN permite que las redes crezcan y se adapten a las necesidades cambiantes sin requerir inversiones significativas en nueva infraestructura.

Como se estudia en este apartado, las WWAN, y en particular tecnologías como LTE-M y NB-IoT, juegan un papel crucial en el ámbito del "Smart Metering". Su capacidad para cubrir grandes áreas, el bajo consumo de energía, la fiabilidad y la escalabilidad las convierten en soluciones óptimas para la recopilación y transmisión de datos de medidores inteligentes, facilitando una gestión más eficiente y precisa del consumo de recursos.

2.4.1. Tecnologías de comunicaciones móviles: GSM, GPRS, UMTS, HSPA, HSPA+, LTE y 5G

Las tecnologías de comunicaciones celulares han evolucionado significativamente desde sus inicios, convirtiéndose en una pieza fundamental para la conectividad en una amplia variedad de aplicaciones, incluyendo el "Smart Metering". A continuación, se ofrece un análisis detallado de estas tecnologías y su relevancia en este ámbito.

Hoy día, las tecnologías de comunicaciones celulares están ampliamente extendidas. Desde que se añadieran las capacidades de GSM¹⁷ a las originales redes de comunicaciones telefónicas, las redes móviles han sido capaces de transmitir servicios de datos. La primera generación de comunicaciones móviles era analógica y únicamente permitía realizar llamadas de voz, con una velocidad de hasta 2.4 kbps. En su segunda generación se realizó la transformación a tecnología digital, permitiendo intercambios de datos hasta los 64 kbps, y hasta 144 kbps con las

¹⁷ GSM (Global System for Mobile Communications) es una tecnología de comunicación celular de segunda generación (2G) que permite la transmisión de voz y datos. Introducida en la década de 1990, GSM digitalizó las comunicaciones móviles, ofreciendo velocidades de datos de hasta 14.4 kbps y proporcionando una cobertura extensa y fiabilidad. Es ampliamente utilizada en todo el mundo y se ha convertido en un estándar para las comunicaciones móviles.

actualizaciones introducidas con GPRS¹⁸. Con las redes de tercera generación se consiguieron alcanzar tasas de transferencia de hasta 2 Mbps y de hasta 14 Mbps utilizando la tecnología HSDPA¹⁹ (High-Speed Downlink Packet Access). Las redes de cuarta generación, denominadas LTE²⁰, ofrecen velocidades de transferencia que pueden alcanzar hasta 1 Gbps.

Las diferentes tecnologías de comunicaciones móviles están basadas en redes celulares, lo que permite dividir el espacio de cobertura en pequeñas celdas con el objetivo de reutilizar las frecuencias en aquellas celdas no adyacentes. Los requisitos principales que pretende conseguir la tecnología LTE son una alta eficiencia espectral, grandes tasas de transferencia de datos y baja latencia. Al tratarse de una tecnología celular interconectada con una infraestructura robusta, el alcance no es un factor limitante, siempre que se esté dentro del área de cobertura de una estación base. En la actualidad, las tecnologías buscan llegar a la quinta generación de redes, conocidas como 5G²¹, que prometen revolucionar la conectividad en las infraestructuras existentes. Estas redes permitirán alcanzar velocidades de transferencia mucho mayores y ofrecerán la capacidad de conectar una amplia variedad de dispositivos a la infraestructura.



Figura 7: Evolución y especificaciones básicas de las tecnologías celulares. (Fuente: ITI)

¹⁸ GPRS (General Packet Radio Service) es una mejora de GSM que introduce la conmutación de paquetes, permitiendo velocidades de datos más altas (hasta 114 kbps). Esta tecnología de 2.5G es adecuada para aplicaciones de "Smart Metering" que requieren transmisión de datos periódica pero no en tiempo real. GPRS ofrece un equilibrio entre cobertura, costo y velocidad de datos, siendo útil para medidores inteligentes en áreas rurales o semiurbanas.

¹⁹ HSDPA (High-Speed Downlink Packet Access) es una tecnología de mejora para redes de tercera generación (3G), específicamente para UMTS, que aumenta significativamente las velocidades de descarga de datos móviles. Introducida a mediados de la década de 2000, HSDPA permite velocidades de transferencia de datos de hasta 14.4 Mbps, mejorando la eficiencia y la capacidad de la red para manejar servicios de datos más intensivos y ofreciendo una mejor experiencia de usuario en aplicaciones como la navegación web, la transmisión de videos y las descargas de archivos.

²⁰ LTE (Long Term Evolution) es una tecnología de cuarta generación (4G) que ofrece velocidades de datos muy altas (hasta 1 Gbps) y baja latencia, lo que la hace ideal para aplicaciones de "Smart Metering" que requieren transmisión de datos en tiempo real y de gran volumen. LTE proporciona una cobertura amplia y es capaz de manejar un gran número de dispositivos conectados, lo que es crucial para redes de medidores inteligentes en áreas urbanas y suburbanas. Los requisitos principales que pretende conseguir LTE son una alta eficiencia espectral, grandes tasas de transferencia de datos y baja latencia. Con LTE, basta con estar dentro del área de cobertura de una estación base para poder acceder a la red.

²¹ 5G es la tecnología de comunicación celular más avanzada, ofreciendo velocidades de datos extremadamente altas (hasta 10 Gbps), baja latencia y una capacidad masiva para dispositivos conectados. En el contexto del "Smart Metering", 5G permite la integración de medidores inteligentes con otros dispositivos IoT, facilitando la recopilación y análisis de datos en tiempo real. La capacidad de 5G para manejar una gran cantidad de dispositivos y su eficiencia espectral la hacen ideal para aplicaciones de "Smart Metering" en entornos urbanos densos y para futuras expansiones de redes inteligentes.

Las tecnologías de comunicación celular son fundamentales para el "Smart Metering" debido a su capacidad para proporcionar una conectividad fiable y de amplio alcance. La selección de la tecnología adecuada se basa en diversos factores, incluyendo la cobertura necesaria, la cantidad de datos a transmitir, la densidad de dispositivos y los costos asociados.

- **Cobertura Extensa**: GSM y GPRS son útiles en áreas rurales donde la cobertura y la fiabilidad son más importantes que la velocidad de datos.
- **Transmisión de Datos en Tiempo Real**: UMTS²², HSPA²³, HSPA+ y LTE son adecuados para aplicaciones que requieren una mayor capacidad de datos y una menor latencia.
- **Alta Densidad de Dispositivos**: LTE y 5G son ideales para entornos urbanos con una alta densidad de medidores inteligentes y la necesidad de transmisión de datos en tiempo real y de gran volumen.

En resumen, las tecnologías de comunicaciones celulares, desde GSM hasta 5G, ofrecen una gama de opciones para implementar soluciones de "Smart Metering" adaptadas a diferentes entornos y necesidades específicas. La evolución de estas tecnologías ha permitido mejorar la eficiencia, la capacidad y la fiabilidad de las redes de medidores inteligentes, facilitando una gestión más precisa y eficiente del consumo de recursos.

2.4.2. LTE-M y NB-IoT

LTE-M (Long Term Evolution for Machines) y **NB-IoT (Narrowband IoT)** son tecnologías de comunicación celular diseñadas para soportar aplicaciones de Internet de las Cosas (IoT) en redes de baja potencia y largo alcance (LPWAN). Ambas tecnologías aprovechan la infraestructura existente de LTE en redes móviles para ofrecer conectividad eficiente y económica a dispositivos IoT que operan con batería y requieren alta eficiencia energética.

LTE-M se refiere específicamente a la categoría LTE CatM1, diseñada para interconectar dispositivos IoT que necesitan velocidades de datos moderadas y una cobertura amplia. Utiliza las bandas LTE existentes y puede alcanzar velocidades de transferencia de hasta 1 Mbps. Con LTE-M, los dispositivos pueden comunicarse a distancias de aproximadamente 5 km, aprovechando los recursos de las estaciones base LTE instaladas para una eficiencia óptima.

NB-IoT, por otro lado, es una tecnología de banda estrecha que también utiliza la infraestructura LTE existente, pero está diseñada para aplicaciones que requieren aún menos velocidad de datos y un consumo de energía extremadamente bajo. NB-IoT puede alcanzar velocidades de transferencia de hasta 250 kbps, pero destaca por su capacidad de alcanzar distancias mucho mayores, hasta 20 km en áreas rurales. Opera en el espectro de GSM y en partes del espectro no utilizado en las bandas LTE, anteriormente reservadas para enviar mensajes SMS.

²² UMTS (Universal Mobile Telecommunications System) es una tecnología de tercera generación (3G) que ofrece velocidades de datos significativamente mayores (hasta 384 kbps). Esta tecnología permite la transmisión de datos en tiempo real y es adecuada para aplicaciones de "Smart Metering" que requieren una mayor capacidad de datos y una latencia más baja. UMTS proporciona una buena cobertura y capacidad para manejar múltiples dispositivos conectados simultáneamente.

²³ HSPA (High-Speed Packet Access) y su evolución HSPA+ son mejoras de UMTS que ofrecen velocidades de datos aún mayores (hasta 14.4 Mbps para HSPA y hasta 42 Mbps para HSPA+). Estas tecnologías de 3.5G proporcionan una mayor eficiencia espectral y una mejor experiencia de usuario. En el ámbito del "Smart Metering", HSPA y HSPA+ son ideales para aplicaciones que requieren una transmisión de datos más rápida y eficiente, especialmente en entornos urbanos con una alta densidad de dispositivos.

Estas dos tecnologías se apoyan en la infraestructura desplegada de LTE en redes de comunicaciones móviles para poder dar soporte a aplicaciones de bajo consumo de potencia y largo alcance (LPWAN). Ambas soluciones están definidas por el grupo 3GPP²⁴ en la Release²⁵ 13 y están destinadas a dar soporte a aplicaciones para el Internet de las Cosas, cuyos dispositivos suelen depender de baterías y necesitan de una alta eficiencia energética y bajo consumo. A grandes rasgos, la diferencia a la hora de elegir entre una tecnología u otra va a depender de la velocidad que queramos conseguir y de la latencia máxima que esté permitida en la aplicación. En la Figura 8 puede ver una comparación que muestra diferentes aplicaciones LPWAN para ambas tecnologías.

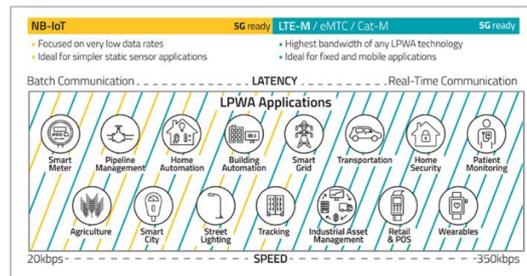


Figura 8: Comparación NB-IoT y LTE-M. (Fuente: [3])

2.4.3. LoRa

En el contexto del Smart Metering, la tecnología LoRa (Long Range) se destaca como una solución eficiente para la comunicación inalámbrica de área amplia (LPWAN). LoRa ofrece un rango de cobertura muy extenso, alcanzando varios kilómetros en entornos urbanos y hasta 15 kilómetros en zonas rurales, dependiendo de las condiciones del terreno y las antenas utilizadas.

Una de las principales ventajas de LoRa es su larga duración de batería, lo que lo hace ideal para dispositivos que necesitan operar durante años sin intervención humana frecuente. Además, la tecnología muestra una buena capacidad para penetrar estructuras densas, lo que es crucial en entornos urbanos complejos.

LoRa opera con un bajo ancho de banda y baja velocidad de datos, características que son adecuadas para aplicaciones de Smart Metering donde se transmiten pequeñas cantidades de datos a intervalos regulares. Esto permite la monitorización remota de medidores de electricidad, agua, gas y otros servicios públicos de manera confiable y eficiente.

En la práctica, LoRa es utilizada por muchas empresas de servicios públicos para implementar redes de medidores inteligentes de manera económica y escalable. Facilita la gestión remota de recursos, la detección temprana de problemas y la optimización de la operación de la red de servicios públicos. Su flexibilidad y capacidad de integración con sistemas existentes hacen de LoRa una opción robusta para aplicaciones críticas como el Smart Metering.

²⁴ El grupo 3GPP (Third Generation Partnership Project) es una colaboración entre organismos de estándares de telecomunicaciones a nivel mundial. Su objetivo principal es desarrollar especificaciones técnicas y estándares para tecnologías de comunicación móvil, incluyendo redes 3G, 4G LTE, 5G y más. El 3GPP garantiza la interoperabilidad y la evolución continua de las tecnologías móviles, permitiendo a fabricantes y operadores implementar soluciones globales compatibles y avanzadas.

²⁵ En el contexto de las tecnologías de comunicación móvil, especialmente definidas por el 3rd Generation Partnership Project (3GPP), una "Release" se refiere a una versión específica de las especificaciones y estándares que define las características, funcionalidades y mejoras de una determinada tecnología. Cada Release representa un conjunto de especificaciones técnicas que pueden incluir nuevas funcionalidades, mejoras de rendimiento, correcciones de errores y soporte para nuevos tipos de dispositivos o servicios.

LoRa fue desarrollada para abordar la creciente demanda de tecnologías de comunicación que requieren bajo consumo energético y amplio alcance. Para lograr estos rangos de cobertura sin depender de potencias de transmisión elevadas, LoRa emplea una tecnología de modulación conocida como Chirp Spread Spectrum²⁶ (CSS). Esta técnica, creada por Semtech²⁷, permite la recuperación de señales que están por debajo del umbral de ruido.

En Europa hay una serie de frecuencias definidas en las que poder utilizar LoRa dentro de las bandas ISM, 868 MHz, 433 MHz y 169 MHz. Las redes LoRa están pensadas para formar topologías en estrella, en las que los dispositivos finales están directamente conectados a un coordinador o pasarela de comunicaciones. LoRaWAN²⁸ es la arquitectura que soporta la tecnología LoRa y puede hacer uso de esta u otras modulaciones.

En las especificaciones están definidas diferentes tasas de transferencia que van desde los 250 bps hasta los 50 kbps, según su ancho de banda y el factor de ensanchamiento de los símbolos utilizado por la modulación definida en LoRa. Dependiendo de la configuración utilizada se pueden llegar a tener alcances de hasta unos 10 km para la configuración con un Spreading Factor más alto.

La capacidad total que se puede conseguir en un único canal LoRa de 125 kHz se puede obtener sumando la tasa de bits de todos los spreading factor, tal y como se muestra en la siguiente ecuación:

$$\text{Capacidad}_{LoRa} = (SF12 + SF11 + SF10 + SF9 + SF8 + SF7 + SF6)$$

$$\text{Capacidad}_{LoRa} = (293 + 573 + 976 + 1.757 + 3.125 + 5.468 + 9.375) = 21.531 \text{ bps}$$

La carga útil que puede transmitir un mensaje LoRa está limitada a 255 bytes transmitidos a nivel físico. Sin embargo, cuando se utiliza los valores de SF más altos se necesita un mayor tiempo en aire para transmitir cada uno de los bytes, por lo que no se recomienda enviar mensajes muy largos ya que se pueden dar errores de sincronización entre dispositivos, así como otras variaciones del canal radio debido a interferencias, que puedan degradar la recepción de los mensajes LoRa.

En las especificaciones de LoRaWAN se definen las diferentes funciones a nivel MAC y a nivel de red que permiten hacer uso de las técnicas de modulación LoRa para establecer comunicaciones entre un Gateway centralizado y los nodos finales. En la definición de arquitectura de LoRaWAN podemos encontrar cuatro tipos diferentes de dispositivos:

²⁶ Chirp Spread Spectrum (CSS) es una técnica de modulación utilizada en comunicaciones inalámbricas donde las señales de radiofrecuencia varían en frecuencia de manera lineal a lo largo del tiempo. Esta técnica, también conocida como espectro ensanchado por chirrido, ayuda a mejorar la eficiencia espectral y la resistencia al ruido, siendo útil en aplicaciones que requieren una transmisión robusta y confiable de datos en entornos desafiantes.

²⁷ Semtech Corporation es una empresa estadounidense especializada en la fabricación de semiconductores y soluciones de alto rendimiento para aplicaciones de comunicaciones analógicas y mixtas, tecnologías de radiofrecuencia, protección de circuitos y gestión de energía. Fundada en 1960 y con sede en Camarillo, California, Semtech ha desarrollado una amplia gama de productos que incluyen semiconductores avanzados para aplicaciones de redes de área amplia de baja potencia (LPWAN), como LoRa (Long Range), así como tecnologías para aplicaciones industriales, de consumo, y de infraestructura crítica.

²⁸ LoRaWAN (Long Range Wide Area Network) es un protocolo de comunicación de bajo consumo diseñado para redes de área amplia (LPWAN) que utiliza la tecnología LoRa para facilitar la conectividad de larga distancia para dispositivos IoT. LoRaWAN permite la transmisión de datos de forma bidireccional entre los dispositivos conectados y las aplicaciones en la nube a través de una infraestructura de red gestionada por operadores de telecomunicaciones, empresas o comunidades. Es conocido por su eficiencia energética, su capacidad para operar en áreas extensas y su flexibilidad para soportar diversos tipos de aplicaciones IoT, como "Smart Metering", agricultura inteligente, gestión de activos y más.

- Los dispositivos finales son los que normalmente generarán los datos para transmitirlos a un dispositivo centralizado. Un nodo final puede ser un sensor, un actuador o cualquier otro tipo de dispositivo que no necesite una transferencia de datos de alta velocidad con la red.
- Los servidores de aplicación son los controladores a los que los diferentes nodos estarán conectados. Estos servidores reciben la información enviada por los nodos y además pueden coordinar su funcionamiento enviando comandos de configuración. Su cometido es recibir los mensajes de los nodos además de configurar los enlaces para establecer dicha comunicación.
- El servidor de red es el encargado de administrar todas las comunicaciones que tienen lugar en la red LoRaWAN. Se comunica tanto con los servidores de aplicación como con las pasarelas. Se podría entender como el router de la red LoRaWAN.
- La pasarela o Gateway permite interconectar los dispositivos finales con el servidor de red. En la misma red LoRaWAN pueden existir más de una pasarela, lo que puede provocar problemas de duplicidad si varias pasarelas reciben información del mismo nodo. Por este motivo se han de seguir una serie de reglas:
 - Todos los mensajes recibidos por una pasarela se deben de reenviar hacia el servidor de red y dado que puede haber múltiples pasarelas, existe la posibilidad de que se reciba el mismo mensaje más de una vez.
 - Cuando el servidor de red quiere enviar un mensaje a un nodo final, elige a la pasarela con la mejor conexión con el dispositivo final para enviarlo.

En LoRaWAN se definen diferentes tasas de transferencia dependiendo de la banda de operación, el SF utilizado y el ancho de banda dedicado a la transmisión. En la Tabla 8 se muestran las diferentes tasas de transferencia especificadas para la banda de 868 MHz en Europa.

| Configuración | Tasa de transferencia (bit/s) |
|-----------------------------|-------------------------------|
| LoRa: SF12 / 125 kHz | 250 |
| LoRa: SF11 / 125 kHz | 440 |
| LoRa: SF10 / 125 kHz | 980 |
| LoRa: SF9 / 125 kHz | 1.760 |
| LoRa: SF8 / 125 kHz | 3.125 |
| LoRa: SF7 /125 kHz | 5.470 |
| LoRa: SF7 / 250 kHz | 11.000 |
| FSK: 50 kbps | 50.000 |

Tabla 8: Tasas de transferencia para los diferentes modos de funcionamiento de LoRaWAN. (Fuente: ITI)

La carga útil que pueden llevar los mensajes de aplicación está limitada a un máximo de 51 bytes para la tasa de transferencia más baja y a un máximo de 222 bytes para las tasas de transmisión más altas.

2.4.4. SigFox

En el ámbito del Smart Metering, SigFox destaca como un operador privado de telecomunicaciones originario de Francia, conocido por su tecnología de comunicación de largo alcance diseñada específicamente para dispositivos de bajo consumo orientados al Internet de las Cosas (IoT). La empresa ha desplegado su propia infraestructura de red en más de 53 regiones alrededor del mundo, utilizando bandas de frecuencia por debajo de 1 GHz, como la banda de 868 MHz en Europa y la de 902 MHz en Estados Unidos.

SigFox emplea tecnología de banda ultra estrecha (Ultra Narrow Band), que maximiza la eficiencia del espectro al operar en anchos de banda muy reducidos. Esto no solo le permite alcanzar un largo alcance, similar a otras soluciones como LoRa, LTE-M y NB-IoT, sino que también proporciona una alta resistencia a las interferencias, asegurando una conectividad robusta y confiable en entornos urbanos densos y en áreas rurales.

La tasa de transferencia de datos de SigFox es limitada, alcanzando hasta 600 bps dependiendo de la región de operación. Este enfoque es ideal para aplicaciones de telemetría donde los dispositivos remotos necesitan enviar pequeñas cantidades de datos de forma periódica y eficiente, como es el caso de los medidores inteligentes en el Smart Metering.

Debido a sus características de baja tasa de transferencia, los dispositivos conectados a redes SigFox tienen limitaciones en la cantidad de mensajes que pueden enviar. Cada dispositivo puede enviar un máximo de 140 mensajes al día, con restricciones adicionales de hasta 7 mensajes transmitidos por hora y 4 mensajes recibidos por día. Los mensajes recibidos por los nodos se utilizan típicamente para configurar parámetros en los dispositivos, aprovechando la capacidad limitada del enlace de bajada para optimizar el consumo energético.

El modelo de negocio de SigFox se basa en el alquiler de su infraestructura de red global, proporcionando una solución rentable y escalable para la conectividad de dispositivos IoT en diversos sectores industriales, incluido el Smart Metering. Esta aproximación permite a las empresas de servicios públicos y otras organizaciones implementar sistemas de medición inteligente que mejoran la eficiencia operativa, reducen costos y permiten una gestión más precisa y proactiva de los recursos.

2.4.5. Comparación de tecnologías LPWAN: Redes de área amplia de baja potencia

En el ámbito de las redes LPWAN, diversas tecnologías emergentes ofrecen soluciones para conectar dispositivos de Internet de las Cosas (IoT) de manera eficiente y efectiva. En esta sección, se presenta una comparación detallada de las tecnologías LPWAN más prominentes: SigFox, LoRa, LTE-M, NB-IoT y 5G, evaluando cuatro criterios clave: alcance, espectro, tasa de bits (bps) y duración de baterías, como puede verse en la Tabla 9.

2.4.5.1. Alcance:

- **SigFox**: Destaca por su largo alcance, que puede extenderse hasta varios kilómetros en entornos urbanos y decenas de kilómetros en áreas rurales.
- **LoRa**: Ofrece un excelente alcance comparable al de SigFox, adaptándose bien tanto a entornos urbanos como rurales.
- **LTE-M y NB-IoT**: Proporcionan un buen alcance que puede variar dependiendo de la infraestructura de red existente.
- **5G**: Aunque principalmente conocido por su alta velocidad, también puede ofrecer un alcance significativo para aplicaciones de IoT en entornos específicos.

2.4.5.2. Espectro:

- **SigFox**: Utiliza un espectro estrecho y dedicado para transmitir datos.
- **LoRa**: Opera en un espectro no licenciado, ofreciendo flexibilidad en términos de frecuencias utilizadas.
- **LTE-M y NB-IoT**: Utilizan el espectro de las redes celulares existentes, permitiendo una integración más sencilla con infraestructuras de telecomunicaciones.
- **5G**: Opera en frecuencias más altas que pueden proporcionar un ancho de banda significativo, pero pueden ser menos eficientes en términos de consumo de energía comparado con tecnologías LPWAN específicas.

2.4.5.3. Tasa de bits (bps):

- **SigFox y LoRa**: Operan típicamente a tasas de bits bajas, ideales para aplicaciones de IoT que transmiten pequeñas cantidades de datos de manera ocasional.
- **LTE-M y NB-IoT**: Ofrecen tasas de bits más altas en comparación con Sigfox y LoRa, adecuadas para aplicaciones que requieren una mayor velocidad.
- **5G**: Destaca por su alta velocidad y baja latencia, siendo adecuado para aplicaciones de IoT que requieren transferencias de datos rápidas y continuas.

2.4.5.4. Duración de baterías:

- **SigFox y LoRa**: Son conocidas por su eficiencia energética, permitiendo una larga duración de batería que puede extenderse desde meses hasta varios años, dependiendo del uso y la configuración.
- **LTE-M y NB-IoT**: Aunque más eficientes que las tecnologías celulares tradicionales, pueden requerir una mayor potencia en comparación con SigFox y LoRa.
- **5G**: Aunque eficiente en términos de ancho de banda, puede requerir un consumo de energía más alto en dispositivos IoT, limitando potencialmente la duración de la batería en comparación con LPWAN.

En conclusión, la elección entre estas tecnologías LPWAN dependerá de las necesidades específicas de cada aplicación de IoT, considerando factores como el alcance requerido, la cantidad de datos a transmitir, la eficiencia energética y la integración con la infraestructura de red existente. Cada tecnología ofrece ventajas distintas que pueden optimizarse según el contexto y los requisitos particulares de cada implementación de IoT.

| | SIGFOX | LoRa | LTE-M | NB-IoT | 5G (objetivos) |
|--------------------------|-----------|-----------|-----------|------------|----------------|
| Alcance | < 13 km | < 11 km | < 11 km | < 15 km | < 15 km |
| Espectro | 900 MHz | 900 MHz | 7-900 MHz | 7-900 MHz | 7-900 MHz |
| Tasa | <100 bps | < 11 kbps | < 1 Mbps | < 200 kbps | < 1 Mbps |
| Duración baterías | > 10 años | > 10 años | > 10 años | > 10 años | > 10 años |

Tabla 9: Comparación de conectividad de tecnologías LPWAN. (Fuente: ITI)

3. Estado del arte y análisis del mercado: Herramientas y soluciones para el despliegue y monitorización de WSN.

Se ha llevado a cabo una investigación exhaustiva en la literatura académica sin encontrar un sistema comparable al que se presenta en este documento. No obstante, existe una gran cantidad de literatura sobre redes de despliegue rápido, un área crucial para la expansión de este tipo de redes y que plantea problemas significativos. Sin embargo, estos problemas se abordan desde enfoques distintos, y hasta la fecha no se ha desarrollado una investigación específica sobre el sistema descrito en este documento. Los estudios académicos previos sobre despliegue rápido en redes de sensores inalámbricos (WSN) se pueden clasificar en varias categorías, entre ellas:

- **Localización**: Algoritmos que permiten a un sensor determinar su ubicación física, reconocer a los nodos vecinos más cercanos y realizar tareas basadas en su posición. Estos algoritmos pueden utilizar GPS, técnicas de triangulación o métodos de inferencia a partir de la intensidad de la señal.
- **Marcado perimetral**: Algoritmos que habilitan a los nodos situados en el borde de la red para reconocer su posición y realizar tareas específicas como la detección de objetos que cruzan el límite de la zona vigilada. Este aspecto es esencial para aplicaciones de seguridad y monitoreo de fronteras.
- **Autocalibración de potencias**: Mecanismos que ajustan la potencia de transmisión de cada nodo al mínimo necesario para mantener la comunicación, aumentando la eficiencia de consumo energético y mitigar los conflictos en la red. Esto contribuye a una mayor eficiencia energética y una mayor duración de los nodos.
- **Asignación automática de direcciones a los nodos**: Son algoritmos que permiten la asignación automática de direcciones de los nodos, eliminando la necesidad de configuración manual. Esto simplifica la instalación y el despliegue de la red, especialmente en entornos grandes o dinámicos.
- **Capacidad de autodescubrimiento**: Son algoritmos que permiten a los sensores comunicar sus capacidades, lo que permite a la WSN y al servidor realizar automáticamente una configuración según las necesidades. Esta capacidad de autodescubrimiento es crucial para la flexibilidad y adaptabilidad de las redes WSN.
- **Gestión eficiente y adaptable de la red**: Son algoritmos que permiten detectar los requisitos de la red y realizar la configuración automática de nodos y la actualización remota. Esto facilita una gestión más eficiente y adaptable de la red, respondiendo a cambios en el entorno o en los requisitos de la aplicación.
- **Sincronización temporal**: Son algoritmos que sincronizan el reloj interno de los nodos de la red para asegurar que todos usan la misma referencia temporal en todas las mediciones. La sincronización precisa es esencial para aplicaciones que requieren coordinación temporal, como el monitoreo ambiental y la seguridad.
- **Herramientas de despliegue**: Esta categoría, que corresponde al enfoque del presente proyecto, incluye soluciones diseñadas para asistir en la colocación física de cada sensor en el área de trabajo. Estas herramientas pueden utilizarse antes, durante o después del despliegue e incluyen:

- **Simuladores de despliegue**: Software que permite modelar y probar diversos escenarios de despliegue antes de implementarlos en el campo. Facilitan la identificación de las mejores ubicaciones para los nodos y la previsión de problemas potenciales.
- **Dispositivos de posicionamiento**: Herramientas físicas que ayudan a colocar los nodos con precisión en el terreno. Incluyen sistemas GPS especializados, herramientas de medición de distancias y niveles, y aplicaciones móviles para asistencia en tiempo real.
- **Sistemas de monitoreo en tiempo real**: Plataformas que permiten supervisar el estado y rendimiento de la red durante el despliegue. Estas herramientas ayudan a identificar y resolver problemas rápidamente, asegurando que todos los nodos estén correctamente ubicados y funcionando.
- **Interfaces de usuario intuitivas**: Aplicaciones y paneles de control diseñados para facilitar la interacción con la red durante el despliegue, proporcionando una vista clara del estado de cada nodo y permitiendo ajustes rápidos y efectivos.

En resumen, aunque no se ha identificado un sistema exacto que coincida con el presentado en este documento, la investigación actual y las tecnologías disponibles proporcionan una base robusta para enfrentar los desafíos asociados al despliegue rápido de redes de sensores inalámbricos. Estas herramientas y métodos continúan avanzando para mejorar la eficiencia operativa, adaptabilidad y robustez de las redes WSN en diversas aplicaciones.

Por ejemplo, diversos estudios sobre despliegue en redes de sensores inalámbricos abordan temas como la localización, la actualización del software de manera automática y a distancia, la configuración autónoma de los nodos y la identificación distribuida de parámetros. Aunque estos estudios ofrecen una visión detallada de los desafíos y soluciones técnicas relacionadas con el despliegue y funcionamiento de nodos en la red, no se enfocan específicamente en herramientas o metodologías destinadas a facilitar el proceso de despliegue. La atención tiende a centrarse en los aspectos técnicos y operativos del despliegue y funcionamiento de los nodos, sin considerar en profundidad el uso de herramientas que podrían simplificar y optimizar la fase inicial de despliegue.

Por lo tanto, el artículo se beneficiaría de una mayor discusión sobre las herramientas prácticas disponibles para los profesionales que buscan implementar redes de manera más eficiente. Implementar redes de sensores es un proceso complejo que enfrenta varios desafíos, como diseñar arquitecturas que garanticen la conectividad entre nodos, optimizar la cobertura de sensores y extender la durabilidad de la red para asegurar su funcionamiento continuo, autónomo y fiable. Con el avance de las tecnologías y el aumento de la complejidad en las redes de sensores inalámbricos (WSN), las tareas de despliegue se están convirtiendo en un desafío significativo tanto para investigadores como para la industria. Aunque se ha puesto un gran énfasis en lograr un funcionamiento autónomo de las WSN, este enfoque aún no se ha aplicado suficientemente a la fase de despliegue inicial, lo que subraya la necesidad de mejorar las prácticas y herramientas para esta etapa crucial.

Existe una gran multitud de estudios y guías como [4], el cual detalla aspectos cruciales como la cobertura, la conectividad y el despliegue en las redes de sensores inalámbricos (WSN). En algunos estudios se examinan en profundidad los problemas fundamentales que afectan a las WSN. Se proporcionan pautas precisas para seleccionar parámetros críticos durante el diseño e

implementación de estas redes. Entre estos parámetros se incluyen la distribución óptima de los nodos sensores para garantizar una cobertura adecuada del área de interés, la configuración de protocolos de comunicación robustos para mantener una conectividad confiable entre los nodos, y estrategias eficientes para gestionar el consumo energético y la vida útil de las baterías de los dispositivos.

En otros estudios, como en [5], se abordan aspectos teóricos y prácticos del despliegue de WSN. Este libro ofrece una comprensión profunda de las teorías fundamentales que respaldan el diseño de redes de sensores inalámbricos, así como recomendaciones prácticas derivadas de experiencias de despliegue en el campo. Se destacan directrices esenciales para una implementación efectiva, que van desde la planificación inicial del despliegue hasta la gestión operativa continua de las redes. Esto incluye la selección adecuada de tecnologías de comunicación inalámbrica, la evaluación de la topología de red más adecuada según el entorno específico de despliegue, y la aplicación de medidas de seguridad para proteger la integridad de los datos recogidos por los sensores.

Aparte del ámbito de investigación, algunos fabricantes que utilizan protocolos basados en el estándar IEEE 802.15.4, como WirelessHART, ISA100.11a, ZigBee o WIA-PA, han desarrollado soluciones y herramientas propias para facilitar la implementación de redes de sensores inalámbricos. Este enfoque surge porque los estándares de estos protocolos no abordan exhaustivamente los desafíos específicos de la fase de despliegue. Sin embargo, estos desafíos son de gran importancia para los usuarios que buscan implementar redes en entornos de automatización y control de procesos industriales.

Los fabricantes ofrecen una variedad de recursos prácticos que incluyen guías paso a paso diseñadas para simplificar el proceso de despliegue. Estas guías proporcionan instrucciones detalladas sobre la configuración inicial, la colocación estratégica de los nodos y la optimización del rendimiento de la red. Además, los fabricantes han desarrollado simuladores electromagnéticos que permiten evaluar la cobertura y la calidad de la señal antes de la instalación física de los dispositivos. Los planificadores de redes son otra herramienta crucial que ayuda a diseñar topologías efectivas y optimizar el uso del espectro de radiofrecuencia disponible.

Para abordar problemas que puedan surgir tras el despliegue, se proporcionan herramientas de monitoreo y depuración que facilitan la identificación y resolución rápida de problemas de conectividad, interferencia o fallos en los nodos. Estas herramientas son esenciales para asegurar el funcionamiento continuo y confiable de las redes de sensores en entornos industriales que exigen altos niveles de fiabilidad.

Entre los protocolos mencionados, WirelessHART destaca por contar con el mayor número de miembros y recursos disponibles, debido a que fue uno de los primeros estándares en establecerse en el mercado. Por otro lado, ISA100.11a está ganando relevancia gracias a su mayor flexibilidad y capacidad de configuración en comparación con WirelessHART. En contraste, ZigBee PRO tiene un alcance más limitado en aplicaciones industriales debido a sus limitaciones originales de diseño. El protocolo WIA-PA²⁹, aunque emergente, está ganando terreno principalmente en el mercado chino, con un potencial de expansión aún por explorar en otros mercados globales.

²⁹ WIA-PA es un estándar diseñado para aplicaciones de automatización industrial. Se caracteriza por ofrecer comunicaciones inalámbricas fiables y seguras en entornos industriales, utilizando tecnología basada en el estándar IEEE 802.15.4. WIA-PA proporciona características específicas orientadas a cumplir con los requerimientos de aplicaciones críticas en sectores como la manufactura, la energía y la automatización de procesos, garantizando robustez, baja latencia y capacidades avanzadas de configuración y gestión de red.

3.1. Análisis de mercado de fases del despliegue de una WSN

En términos generales, el despliegue de una red de sensores inalámbricos (WSN) se compone de la fase de pre-despliegue, en la que se realizan los diseños, determinando las ubicaciones de los sensores y llevando a cabo los ajustes y configuraciones previas al despliegue; la fase de despliegue, en la que se procede a la instalación física de los nodos y se realizan pruebas y configuraciones para integrar los nodos en la red; y, finalmente, la fase de post-despliegue, en la que se valida que el sistema está funcionando correctamente y se hacen configuraciones para asegurar el rendimiento óptimo de la red.

A continuación, se explican en detalle las tres fases del despliegue de una WSN.

3.1.1. Fase de antes del despliegue (Pre-Despliegue)

En las fases previas al despliegue de una red de sensores inalámbricos (WSN), se emplean diversas soluciones tanto en la comunidad científica como en la industria. Estas soluciones incluyen algoritmos y estrategias que buscan determinar la ubicación óptima de los nodos para un sistema específico, enfocándose en aspectos como la conectividad de los nodos, la maximización del área de cobertura y la mejora de la eficiencia energética para prolongar la vida útil de la red. Otras herramientas destacables son los simuladores que también valiosas en esta etapa, ya que proporcionan resultados teóricos preliminares, tales como la cobertura radioeléctrica, la ubicación de los nodos, las estimaciones de duración de la red y la asignación de roles.

A continuación, se presentan algunas de estas estrategias y soluciones, que van desde técnicas especializadas para el despliegue en dos dimensiones hasta enfoques más generales que abordan aspectos críticos de las WSN.

Existen numerosos estudios dedicados al desarrollo de algoritmos y estrategias para optimizar el despliegue de redes de sensores inalámbricos (WSNs). Estos estudios abarcan desde técnicas específicas para el despliegue en dos dimensiones³⁰ hasta enfoques más generales que abordan múltiples aspectos críticos de las WSNs.

En el contexto del despliegue en dos dimensiones, se destacan investigaciones que proponen métodos para maximizar el área de sensorización utilizando modelos poligonales y algoritmos incrementales eficientes, respectivamente. Además, otros estudios exploran algoritmos inspirados en la inteligencia de enjambres³¹, como el algoritmo de optimización de enjambre de luciérnagas, y el protocolo de despliegue basado en la optimización de colonias de hormigas.

Otro enfoque crucial es la maximización de la cobertura radio, que emplean esquemas distribuidos y algoritmos genéticos para mejorar la cobertura del área de interés. Estos estudios son fundamentales para garantizar que los nodos sensores estén estratégicamente ubicados para optimizar la eficiencia de la red.

³⁰ El despliegue de redes de sensores inalámbricos (WSNs) en dos dimensiones se refiere al proceso de distribuir estratégicamente los nodos sensores en un plano bidimensional, como por ejemplo en una superficie terrestre o acuática. Este despliegue tiene como objetivo cubrir eficientemente el área de interés para la recolección de datos, asegurando una adecuada conectividad entre los nodos para la transmisión de información. Las técnicas de despliegue en 2D consideran factores como la densidad de cobertura, la colocación óptima de los nodos para minimizar las zonas sin cobertura y la maximización del alcance de las comunicaciones inalámbricas, todo ello con el fin de optimizar el rendimiento y la eficiencia de la red de sensores.

³¹ Los algoritmos inspirados en la inteligencia de enjambres son técnicas de optimización bioinspiradas que se basan en el comportamiento colectivo de sistemas naturales, como enjambres de insectos, aves o peces. Estos algoritmos son aplicados en el contexto del despliegue de redes de sensores inalámbricos (WSN) para optimizar la ubicación y configuración de los nodos.

Además, se han desarrollado mecanismos para el despliegue que abordan aspectos específicos como la capa física, la gestión del balance de energía para prolongar la vida útil de la red, el control de la topología, la conservación de energía, la probabilidad de detección de objetivos, y la consideración de desvanecimientos de señal y diversidad de frecuencia. Todos estos mecanismos contribuyen a la optimización general de las redes de sensores inalámbricos (WSNs). Dado que estos enfoques tienden a centrarse en aspectos particulares, algunos investigadores han propuesto estrategias multipropósito que buscan abordar y compensar las limitaciones de los enfoques específicos mencionados.

Finalmente, aunque con menor frecuencia, también se investigan estrategias de despliegue para redes de sensores en entornos tridimensionales. Este enfoque resulta especialmente relevante en escenarios donde los nodos no se colocan en un único plano, como en redes de sensores subacuáticas o en entornos forestales.

Estos estudios y enfoques representan una contribución significativa al campo de las redes de sensores inalámbricos, proporcionando herramientas y metodologías que permiten abordar los desafíos complejos asociados con el diseño, implementación y gestión de estas redes en una amplia gama de aplicaciones prácticas y escenarios ambientales.

En las fases previas al despliegue, es común utilizar simuladores que integran diversos algoritmos de optimización. Un ejemplo es WINSS, una herramienta de simulación que se basa en NS-2³² para configurar simulaciones, permitiendo el diseño y prueba de redes que utilizan el estándar IEEE 802.15.4. Existen numerosos simuladores similares para redes de sensores inalámbricos (WSN). Un estudio de B. Musznicki y P. Zwierzykowski proporciona una comparación de varios simuladores y emuladores, tales como MSPsim³³, TOSSIM³⁴, WSim³⁵, SENSE, NetTopo, COOJA y Castalia, entre otros.

En la investigación también se ha hallado la herramienta KleeNet³⁶ que permite descubrir errores debidos a eventos no deterministas, como reinicios o paquetes duplicados, antes de realizar el despliegue. Además, KleeNet extiende las capacidades de KLEE para adaptarse a los entornos específicos de las WSNs.

³² NS-2 (Network Simulator 2) es una herramienta de simulación discreta para redes de comunicación, ampliamente utilizada en el ámbito académico e industrial para la investigación y desarrollo de protocolos de red y tecnologías de comunicación. Permite modelar y simular una amplia gama de redes, desde redes cableadas hasta inalámbricas, proporcionando una plataforma flexible para evaluar el rendimiento y comportamiento de la red.

³³ MSPsim es un simulador del comportamiento de microcontroladores basados en la arquitectura MSP430 de Texas Instruments. Este simulador se utiliza principalmente en el desarrollo y prueba de aplicaciones para WSNs. MSPsim permite a los desarrolladores ejecutar y depurar código en un entorno virtual que reproduce fielmente el funcionamiento del hardware MSP430.

³⁴ TOSSIM es un simulador diseñado para redes de sensores inalámbricos que emula el entorno de ejecución del sistema operativo TinyOS. Utilizado principalmente en la investigación y desarrollo de aplicaciones para WSNs, TOSSIM permite a los desarrolladores simular el comportamiento de redes grandes y complejas, ofreciendo la capacidad de probar y depurar código en un entorno virtual.

³⁵ WSim es un simulador de hardware y redes diseñado para emular microcontroladores y dispositivos de WSNs. Es particularmente útil para el desarrollo y la prueba de aplicaciones en plataformas de sensores como los basados en la arquitectura MSP430. WSim permite a los desarrolladores ejecutar y depurar sus programas en un entorno simulado que reproduce fielmente el comportamiento del hardware y las interacciones de red, lo que facilita la identificación y solución de problemas antes de la implementación en dispositivos físicos.

³⁶ KleeNet es una herramienta de análisis simbólico basada en KLEE, diseñada para la verificación y prueba de aplicaciones en redes de sensores inalámbricos (WSNs). Al extender las capacidades de KLEE, KleeNet permite realizar una exploración exhaustiva y sistemática de los posibles estados de un programa, ayudando a detectar errores y vulnerabilidades.

KLEE es una herramienta de verificación de software que realiza análisis simbólico para generar casos de prueba de alta cobertura y encontrar errores en programas. Desarrollada para funcionar con programas en lenguaje C, KLEE ejecuta todas las posibles rutas de ejecución del programa de manera simbólica en lugar de utilizar valores concretos. Esto permite detectar una amplia variedad de errores, incluyendo fallos de segmentación, fallos de memoria y violaciones de aserciones, proporcionando así una verificación exhaustiva y automatizada del software.

Los fabricantes de dispositivos que utilizan los protocolos industriales basados en IEEE 802.15.4 ofrecen una variedad de herramientas para la simulación y planificación en las fases previas al despliegue. Dado que WirelessHART fue el primer protocolo en entrar en el mercado de la Automatización Industrial³⁷, cuenta con una amplia gama de fabricantes asociados. Empresas como Emerson, Pepperl+Fuchs y Phoenix Contact proporcionan herramientas para el diseño de redes WirelessHART, simulación de la propagación de ondas y conectividad en modelos tridimensionales de plantas, planificación de la ubicación de nodos y estimación del consumo de baterías. Ejemplos de estas herramientas incluyen la “Wireless Planning Tool” de Emerson, “Wireless Network Checker” de Pepperl+Fuchs, “WirelessHART Network Planner Tool” de Phoenix Contact, “AMS Wireless SNAP-ON™ Application” de Emerson, “Life and cost estimators” de Emerson y “Wireless Network Planner” de SoftDEL. Por otro lado, Honeywell ofrece la “RF-Lite Tool” para la planificación de redes que utilizan el estándar ISA100.11a. En cuanto a ZigBee PRO, también se encuentran herramientas como ZEIN, que facilita la planificación de redes ZigBee en entornos interiores.

Aunque hay una amplia gama de opciones para abordar la fase de pre-despliegue, diversos autores coinciden en que los despliegues reales suelen diferir significativamente de las simulaciones previas. Esto se debe a que la complejidad del entorno real introduce variables como obstáculos, condiciones climáticas, interferencias cercanas y desajustes de relojes, lo que puede generar errores durante el proceso de despliegue. Como resultado, el despliegue suele requerir asistencia adicional para corregir estos problemas. Esta complejidad, junto con el tiempo requerido y los conocimientos necesarios para esta fase, representa un desafío importante para los sistemas de despliegue rápido.

3.1.2. Fase de despliegue (Despliegue “In-Site”)

En un escenario ideal, un despliegue rápido debería comenzar en la fase de despliegue, eliminando la etapa de pre-despliegue para reducir tanto el tiempo como la complejidad. No obstante, uno de los principales desafíos durante esta fase es que los motes³⁸ no cuentan con interfaces de usuario para verificar o configurar parámetros, ya que están diseñados principalmente para la recolección de datos o el control de sistemas. Por ejemplo, uno de los motes más avanzados en términos de interfaz de usuario solo tiene tres LEDs que pueden indicar hasta ocho estados diferentes. Sin embargo, existen herramientas de validación para la fase de despliegue (DTV – Deployment Time Validation) que emplean equipos adicionales con sus propias interfaces para realizar tareas de verificación. Estos equipos, que se utilizan durante el proceso de despliegue, no forman parte de la red en sí, sino que sirven como instrumentación adicional para que los técnicos puedan diagnosticar y resolver problemas en el sistema.

³⁷ El sector de la Automatización Industrial se refiere al uso de tecnologías avanzadas para controlar y operar procesos industriales de manera automatizada. Esto incluye el uso de sistemas computarizados, software y hardware especializado para monitorear y controlar maquinaria, procesos de manufactura, producción y otros sistemas industriales.

³⁸ Los "motes" son dispositivos autónomos utilizados en WSNs para la recolección de datos y el control de sistemas. A menudo carecen de interfaces de usuario avanzadas y están optimizados para tareas específicas dentro de la red. Durante el despliegue de la red, herramientas de validación en el tiempo de despliegue (DTV) proporcionan interfaces adicionales para la configuración y diagnóstico, facilitando la instalación y puesta en marcha del sistema.

El proyecto TASK, desarrollado en colaboración entre la Universidad de Berkeley y el MIT, abordó los retos asociados con la fase de despliegue en redes de sensores inalámbricos (WSN) desde sus primeras etapas. TASK diseñó un conjunto de herramientas para monitorización y configuración que asegura el funcionamiento óptimo de la red antes de finalizar la instalación. En esta fase, se utiliza un software instalado en un PDA portátil que permite verificar si los nodos cercanos están operativos mediante un ping directo, aunque esta herramienta no evalúa la conectividad completa de la red.

Por otro lado, el proyecto LUSTER (ver Figura 9) tiene como uno de sus objetivos principales el desarrollo de una herramienta de validación in-situ llamada SeeMote. Esta herramienta forma parte del kit de técnicas y procedimientos denominado SeeDTV, que se encarga de identificar problemas en los dispositivos y de verificar la integridad física y lógica de la red. SeeMote también comprueba la conectividad con el servidor final, un aspecto crucial en la fase de despliegue. El dispositivo emite paquetes de control de manera periódica, y los nodos responden con información sobre su estado de salud, incluyendo identificador local, valor RSSI, batería restante y datos del conversor AD. Esta información se muestra en una pantalla LCD de 1.8 pulgadas, que permite navegar por diferentes menús para obtener información detallada.

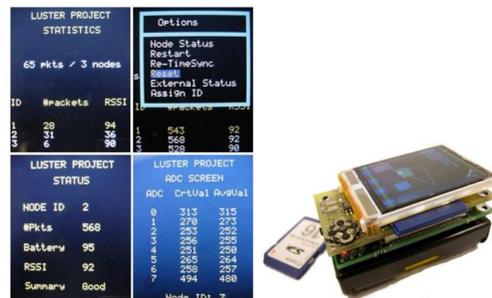


Figura 9: Solución del proyecto LUSTER. (Fuente: ITI)

El equipo SeeMote es lo suficientemente compacto como para llevarlo en un bolsillo y es de 8 a 9 veces más eficiente en consumo que un PDA iPAQ, similar al utilizado en TASK. Sin embargo, está diseñado para redes TASK con topología en estrella (maestro-esclavo) de un solo salto.

Existe un dispositivo de mano (ver Figura 10), basado en las implementaciones de los proyectos TASK y LUSTER. A diferencia de otros dispositivos, su funcionamiento es pasivo y se centra en la visualización de las medidas enviadas por un nodo específico. Este equipo opera capturando paquetes a través de sniffing, es decir, escuchando el tráfico en la red y extrayendo datos de los paquetes transmitidos por las estaciones. Además de mostrar las medidas de los nodos, el dispositivo proporciona información sobre el estado de la red y la conectividad, ayudando a identificar y resolver problemas de manera temprana.



Figura 10: Representación de estadísticas de calidad de TASK. (Fuente: ITI)

Por otro lado, se puede encontrar una suite de herramientas que puede utilizarse desde cualquier dispositivo de mano para evaluar diferentes aspectos de la red de sensores. Estas pruebas incluyen:

- **Conectividad con el padre:** Esta prueba verifica si cada nodo en una red con topología en árbol tiene conexión con su nodo "padre". Se considera que la red tiene conectividad completa si todos los nodos están conectados con sus respectivos padres. Durante la prueba, se generan y envían paquetes entre nodos padre e hijo, y el dispositivo de mano reporta la tasa de pérdida de paquetes y la calidad del enlace. Es responsabilidad del usuario realizar estas comprobaciones para cada dispositivo y repetirlas si se hacen cambios en la red.
- **Cobertura de la red:** Después de asegurar la conectividad con el padre, esta prueba verifica que todos los mensajes enviados desde nodos de origen lleguen a sus destinos correctamente, asegurando así la integridad de la cobertura de la red.
- **Cobertura del dispositivo:** Esta prueba específica evalúa la calidad del enlace y la tasa de pérdida de paquetes entre un mote y un nodo destino particular, proporcionando información crucial sobre la eficacia de la comunicación en la red.

MoteFinder es una herramienta diseñada para interactuar con nodos específicos dentro de una red en despliegue, permitiendo localizar nodos, medir la potencia de la señal y detectar nodos que funcionen incorrectamente o que estén causando problemas en la red. Funciona mediante un PC y un mote con conexión USB.

TestbedProfiler es una suite de aplicaciones desarrollada para asistir en la instalación de bancos de pruebas para WSN. Esta herramienta se utiliza durante el despliegue de los nodos para evaluar su ubicación en términos de conectividad e intensidad de señal. A diferencia de herramientas como SeeDTV, que son dispositivos de mano para inspeccionar una ubicación específica, TestbedProfiler ofrece una visión completa de la red, utilizando un canal de comunicaciones dedicado para recolectar estadísticas de diversos nodos hacia un servidor central. Su limitación principal es que está diseñado específicamente para nodos que soporten el radio Chipcon CC2420.

Otra posibilidad explorada es el uso de nodos sensores móviles para llevar a cabo las tareas de despliegue, permitiendo mejorar la cobertura de manera dinámica. En este enfoque, el área de despliegue se divide en múltiples celdas individuales, y se implementa una estrategia de gestión de potencia asíncrona para optimizar el consumo energético de la red de sensores. Los nodos sensores móviles, que incluyen una variedad de sensores y actuadores, son capaces de desplazarse por el entorno para ajustar su posición y maximizar la cobertura. Este sistema ofrece la ventaja de realizar despliegues automáticos con un alto grado de precisión, adaptándose a las condiciones cambiantes del entorno.

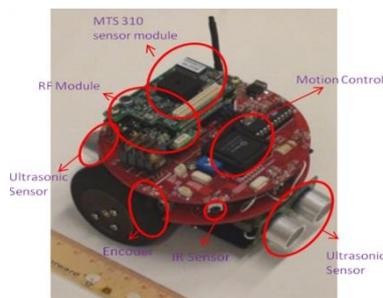


Figura 11: Tareas de despliegue mediante equipos móviles. (Fuente: ITI)

Existe también, un sistema para ayudar en el despliegue de campo, que asiste en el despliegue de redes de sensores inalámbricos, orientando a los técnicos sobre la colocación adecuada de los nodos. Esta herramienta utiliza los motes ITTH desarrollados por el Indian Institute of Technology de Hyderabad y conecta los nodos a un smartphone para monitorizar la calidad del enlace. Una aplicación de Android proporciona datos en tiempo real sobre la calidad del enlace, indicando cuándo y dónde es apropiado instalar cada nodo. El proceso de despliegue se inicia en la estación base y se expande con la colocación de nodos de retransmisión (Relay Nodes) siguiendo las sugerencias de la herramienta. Posteriormente, la estación base recoge y transmite datos de los nodos al smartphone a través de un servidor back-end, facilitando la visualización de la topología de la red. Además, se ha desarrollado un algoritmo que optimiza la duración de la batería, sugiriendo la colocación de nodos de apoyo cerca de aquellos que consumen más energía para extender la vida útil de la red.



Figura 12: Demostración de despliegue mediante aplicación móvil. (Fuente: ITI)

Por último, se ha identificado un conjunto de herramientas para abordar los desafíos que genera el despliegue de una WSN. Este sistema se centra en el uso de un dispositivo portátil (Hand-Held Device) durante el proceso de despliegue y puesta en marcha. El dispositivo portátil consta de dos componentes principales: un dispositivo móvil y una plataforma de hardware para la WSN. Se incluye un simulador de conectividad que permite comparar los resultados obtenidos durante el despliegue con los escenarios simulados, utilizando diversos algoritmos de despliegue. La combinación de las capacidades GPS y Wi-Fi del dispositivo móvil con la conectividad de la plataforma WSN proporciona un sistema de guía automático que facilita la instalación precisa de los nodos. Además, el sistema incorpora una plataforma de pruebas para detectar errores, permitiendo la evaluación de parámetros de los nodos sensores en tiempo real.

En los estándares industriales mencionado se definen dispositivos conocidos como "Handheld Devices"³⁹, cuyo propósito es funcionar como nodos capaces de interactuar con la red para recopilar datos de los nodos individuales y de la red en su totalidad. Estos dispositivos permiten a un operario desplazarse a una zona donde la red no esté funcionando como se espera, y analizar la información recopilada por los nodos en esa área, lo que permite evaluar la salud de sus enlaces. Aunque estos dispositivos están contemplados en los estándares, su implementación queda a discreción de los fabricantes. Por ejemplo, Emerson ofrece el 475 Field Communicator, que cumple la función de los Handheld Devices según el estándar de WirelessHART. Otros fabricantes, como Yokogawa con su YHC5150X Fieldmate Handheld, Endress+Hauser con el Field Xpert SFX370, y Fluke y ABB con el Fluke 754 y el DHH805-A, respectivamente, tienen dispositivos similares.

³⁹ Los "Handheld Devices" son dispositivos portátiles diseñados para ser transportados y utilizados fácilmente con una mano. Suelen incluir funcionalidades como computación, comunicación y captura de datos, y están optimizados para ser utilizados en movilidad. Ejemplos comunes incluyen smartphones, tabletas y dispositivos de mano especializados para aplicaciones específicas como la recolección de datos, la navegación o el control remoto.



Figura 13: Field Devices de WirelessHART e ISA100.11a. (Fuente: ITI)

A diferencia de las herramientas específicas utilizadas en estudios de investigación, estos dispositivos están diseñados para un uso general y permiten configurar una amplia gama de equipos desplegados en planta, tanto inalámbricos como cableados. En la siguiente figura se ilustra el dispositivo Field Xpert SFX370 de Endress+Hauser, donde se pueden observar las diversas conexiones con dispositivos WirelessHART, equipos cableados que utilizan el protocolo HART, sistemas de monitorización, dispositivos Fieldbus, entre otros.

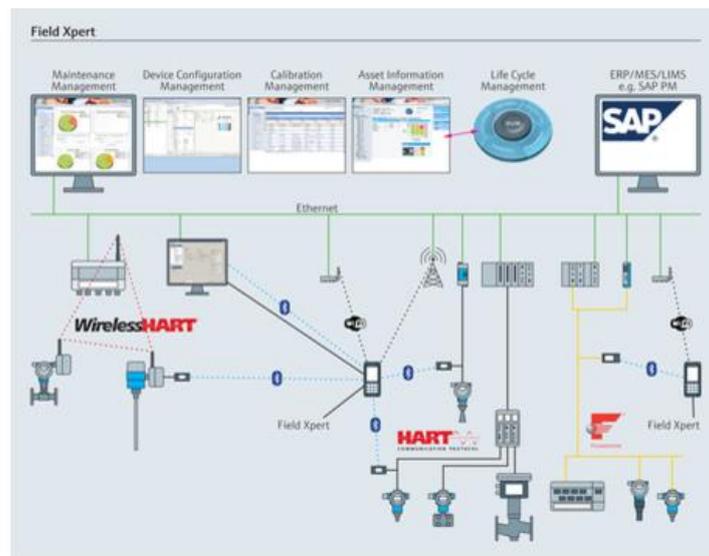


Figura 14: Arquitectura de Field Xpert. (Fuente: ITI)

Además de los dispositivos mencionados, los fabricantes proporcionan diversas guías que son útiles durante el proceso de despliegue, ya que detallan procedimientos iterativos. También se encuentran disponibles herramientas que apoyan al usuario durante la fase de puesta en marcha de redes WirelessHART.

Como se observa, varias investigaciones y herramientas comerciales abordan conceptos relacionados con nuestro proyecto, pero todavía no se ha desarrollado el concepto de nodos que apoyen directamente al usuario en la fase de despliegue. Creemos que esta característica representa una ventaja significativa frente a la competencia existente.

3.1.3. Fase de después del despliegue (Post-Despliegue)

Uno de los principales desafíos en la fase de despliegue es la necesidad de acudir personalmente a cada nodo para realizar ajustes o usar infraestructura adicional para la comunicación con ellos. En algunos artículos proponen el uso de motes con doble radio (802.15.4 + Bluetooth), donde la red 802.15.4 sirve como WSN, y la red Bluetooth forma una Scatternet que facilita la programación y depuración de los nodos.

Para evaluar el funcionamiento adecuado de una WSN después del despliegue, se utilizan diversas herramientas de monitorización, prueba y validación que aseguran que la red opere correctamente. Existen estudios sobre herramientas para diagnosticar fallos en la fase de post-despliegue en WSNs. Estos estudios indican que, aunque existe una variedad de herramientas, no hay una estandarización en este campo, y muchas son aún proyectos de investigación. Las herramientas se clasifican en tres categorías: las basadas en tráfico, como Sympathy y SNIF, que utilizan análisis de datos para detectar errores; las que evalúan el estado de los nodos, como Nucleos o NodeMD, que proporcionan información detallada sobre cada nodo para identificar problemas; y las que analizan el estado global de la red, como Memento o Wringer, que combinan datos de múltiples nodos para ofrecer una visión general del estado de la red.

En el ámbito de herramientas de monitorización, existen opciones como MoteView, un producto de Crossbow que permite visualizar una lista de nodos desplegados, conectarse a bases de datos y analizar las medidas de los sensores. Otras herramientas similares incluyen SpyGlass, TinyViz, Surge Network Viewer, MonSense, Octopus, Trawler (de MotelV Corporation), SNAMP, MeshNetics WSN Monitor, Mica Graph Viewer y Marwis.

NViz es una herramienta de visualización flexible para redes de sensores inalámbricos (WSNs) que supera las limitaciones de soluciones específicas con formatos de paquetes predefinidos, permitiendo configurar parámetros para visualizar datos de cualquier WSN. NSSN ofrece una herramienta de monitorización y captura de paquetes usando nodos especiales, denominados NSSNer nodes, para analizar la red sin interrumpir su funcionamiento. EMMON proporciona un conjunto completo de herramientas para monitorear aplicaciones basadas en WSN, incluyendo planificación de despliegues, análisis de dimensionamiento, simulación de protocolos y programación remota automática de nodos. MINERVA es una arquitectura de depuración para testbeds de WSNs que detecta y soluciona errores a nivel de red y nodo mediante una plataforma de depuración distribuida en cada nodo. Por último, EPMOST es un sistema de monitorización pasivo que utiliza SNMP para reducir el consumo de energía en WSNs.

Para casos en los que, tras el despliegue, algunos nodos queden aislados y no se conecten a la red, se han desarrollado protocolos como el de ESA, que aborda la cobertura de áreas faltantes mediante nodos móviles que pueden trasladarse a zonas con mejor cobertura.

Además, hay herramientas especializadas para la monitorización y diagnóstico de redes que usan estándares industriales como WirelessHART, ISA100.11a y ZigBee PRO. Un ejemplo es el SoftDEL Wireless Network Monitor una aplicación que permite monitorizar redes con estos protocolos.

3.2. Análisis de Mercado de Tecnologías y herramientas de funcionamiento desasistido

El desarrollo y la implementación de sistemas de Redes de Sensores Inalámbricos (WSN) han ganado una relevancia significativa en diversas aplicaciones, desde la agricultura de precisión hasta la automatización industrial y las ciudades inteligentes. En este contexto, el sistema de Red de Sensores de Despliegue Rápido (Quick Deployment Sensor Network - QDSN, por sus siglas en inglés), desarrollado en ITI, representa un avance crucial en la capacidad de monitoreo y análisis de datos en tiempo real. Para asegurar un funcionamiento eficiente y desasistido, es esencial considerar un conjunto de tecnologías y herramientas clave que permitan una gestión autónoma y fiable de la red.

Este apartado se enfoca en el análisis de mercado de dichas tecnologías y herramientas, proporcionando una visión detallada de las opciones disponibles y su relevancia en la implementación de QDSN. Se explorarán los protocolos de comunicación, las plataformas de gestión, las tecnologías de recolección de energía, y las soluciones de inteligencia artificial y machine learning⁴⁰, todas ellas fundamentales para el diseño de un sistema robusto y eficiente. Además, se identificarán los principales actores en la industria y las tendencias emergentes que están moldeando el futuro de las WSN.

3.2.1. VersaSense

VersaSense ofrece una solución integral de IoT (Internet de las Cosas) basada en sensores y controladores inalámbricos que permite la monitorización y control eficiente de diversas variables en entornos industriales y comerciales. El sistema VersaSense integra componentes de software, red y hardware para garantizar la seguridad y la captura precisa de datos de los sensores, así como de otros elementos de la instalación.

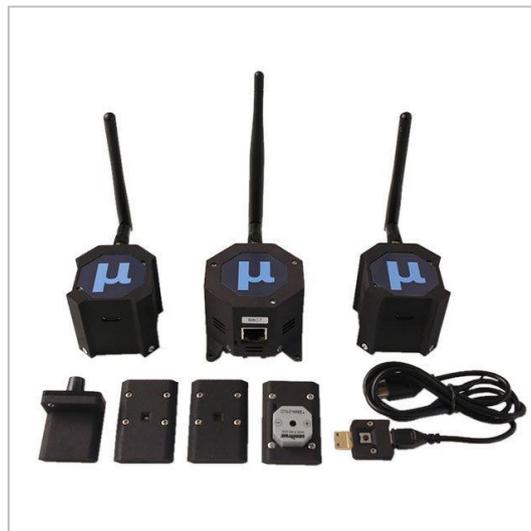


Figura 15: Solución de VersaSense. (Fuente: ITI)

⁴⁰ Machine learning (aprendizaje automático) es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar su desempeño en tareas específicas a partir de datos y experiencias previas, sin ser programadas explícitamente para esas tareas. Utiliza técnicas estadísticas y matemáticas para identificar patrones en los datos y hacer predicciones o tomar decisiones basadas en esos patrones. Los modelos de machine learning pueden ser supervisados (entrenados con datos etiquetados), no supervisados (donde la estructura de los datos se descubre sin etiquetas) o de aprendizaje por refuerzo (donde los modelos aprenden a través de la interacción con un entorno y la recepción de recompensas o castigos).

El núcleo del sistema VersaSense se basa en la tecnología SmartMesh IP de Linear Technology® (Analog Devices), que proporciona una red mesh con una fiabilidad del 99.999%. Esta tecnología asegura una comunicación robusta y de alta disponibilidad, esencial para aplicaciones críticas donde la pérdida de datos no es una opción. Además, la red mesh de VersaSense tiene un alcance efectivo de hasta 50 metros, lo que permite cubrir amplias áreas con una infraestructura mínima.

VersaSense ofrece dos tecnologías de comunicación para adaptarse a diversas necesidades y escenarios de implementación: SmartMesh y LoRaWAN. SmartMesh es ideal para aplicaciones que requieren alta fiabilidad y baja latencia, mientras que LoRaWAN es adecuado para despliegues de larga distancia y bajo consumo de energía.

En cuanto a la variedad de sensores compatibles, VersaSense proporciona una amplia gama que incluye:

- Sensor de Presencia: Detecta la presencia o movimiento de personas u objetos.
- Acelerómetro: Mide la aceleración y las vibraciones en tres ejes.
- Sensor de Temperatura: Monitoriza la temperatura ambiental o de superficies específicas.
- Sensor de Humedad: Mide la humedad relativa del ambiente.
- Micrófono: Captura sonidos y niveles de ruido.
- Barómetro: Registra la presión atmosférica.
- Sensor de Radiación Solar: Mide la radiación solar, requiere alimentación externa.
- Sensor de Luz: Detecta la intensidad lumínica.
- Lector RFID: Lee etiquetas RFID, requiere alimentación externa.
- Relé DC: Controla dispositivos de corriente continua.
- Relé AC: Controla dispositivos de corriente alterna.
- Sensor de Temperatura de Superficie: Mide la temperatura de superficies específicas.
- Sensor de Distancia Ultrasónica: Determina la distancia mediante ultrasonidos, requiere alimentación externa.
- Sensor de Detección de Objetos: Detecta la presencia de objetos en su proximidad.
- Buzzer: Emite señales sonoras para alertas o notificaciones.
- Sensor de Gas: Detecta diversos gases, incluyendo pH, oxígeno, amoníaco y etileno.
- Pinza Amperimétrica: Mide la corriente eléctrica sin necesidad de contacto directo.
- Sensor Láser de Distancia: Mide distancias con alta precisión utilizando tecnología láser.

Estos sensores permiten una monitorización completa y precisa de múltiples parámetros, lo que es esencial para aplicaciones en industrias como la manufactura, la agricultura, la gestión de edificios inteligentes, y más. La versatilidad y la fiabilidad del sistema VersaSense lo convierten en una solución robusta para la implementación de proyectos IoT a gran escala, facilitando la recolección de datos en tiempo real y el control automatizado de procesos.

3.2.2. Solución de desarrollo Zigbee de NXP

La solución de desarrollo Zigbee de NXP es una oferta integral que facilita la creación de dispositivos y sistemas basados en el protocolo Zigbee, diseñado para aplicaciones de bajo consumo energético y redes de corto alcance. Esta solución se compone de varios elementos clave que juntos proporcionan un entorno robusto y eficiente para desarrolladores.



Figura 16: Solución de Desarrollo Zigbee de NXP. (Fuente: ITI)

En el centro de la solución están los microcontroladores y transceptores de NXP, diseñados específicamente para aplicaciones Zigbee. Estos componentes son optimizados para ofrecer un bajo consumo de energía y un rendimiento confiable en redes inalámbricas. Los microcontroladores suelen integrar características avanzadas de procesamiento y comunicación, lo que permite a los dispositivos gestionar complejas tareas de red mientras mantienen una alta eficiencia energética.

La oferta de NXP también incluye pilas de protocolos Zigbee certificadas. Estas pilas de protocolos son fundamentales para la implementación del estándar Zigbee, asegurando que los dispositivos sean interoperables con otros dispositivos Zigbee en el mercado. El software de desarrollo proporcionado por NXP, como el SDK de Zigbee, incluye herramientas y bibliotecas que simplifican la creación y prueba de aplicaciones, acelerando el tiempo de desarrollo y reduciendo la complejidad.

Las herramientas de desarrollo de NXP, como los kits de evaluación y las herramientas de depuración, son esenciales para los ingenieros que trabajan en proyectos Zigbee. Estas herramientas permiten a los desarrolladores diseñar, probar y optimizar sus dispositivos de manera eficiente. Los kits de desarrollo típicamente incluyen placas de evaluación que contienen los microcontroladores y transceptores necesarios, además de interfaces para conectarse a otros dispositivos y redes.

Un aspecto crítico de la solución Zigbee de NXP es la seguridad. Los productos de NXP integran características avanzadas de seguridad para proteger las comunicaciones inalámbricas y garantizar la integridad y confidencialidad de los datos transmitidos. Esto es particularmente importante en aplicaciones sensibles como la automatización del hogar y la medición inteligente, donde la seguridad de los datos es primordial.

Además, los productos de NXP están diseñados para cumplir con los estándares Zigbee y facilitar la certificación de dispositivos. Esto asegura que los dispositivos desarrollados sean compatibles con otros dispositivos Zigbee en el mercado, lo cual es crucial para la interoperabilidad en ecosistemas de IoT (Internet de las Cosas) donde múltiples dispositivos de diferentes fabricantes necesitan comunicarse de manera eficiente y segura.

Las aplicaciones de la solución Zigbee de NXP son diversas y abarcan múltiples industrias. En la automatización del hogar, los dispositivos Zigbee permiten el control de iluminación, termostatos, sistemas de seguridad y otros dispositivos inteligentes, mejorando la comodidad y la eficiencia energética en los hogares. En el campo de la medición inteligente, los sistemas Zigbee permiten la recolección de datos en tiempo real y la gestión eficiente del consumo energético, lo cual es esencial para las compañías de servicios públicos y los consumidores. En entornos industriales, las redes de sensores y actuadores basadas en Zigbee permiten el monitoreo y control eficientes, mejorando la productividad y reduciendo los costos operativos. Además, en el sector de la salud y el bienestar, los dispositivos Zigbee pueden usarse para el monitoreo remoto de la salud, proporcionando datos fiables y en tiempo real que pueden mejorar la atención al paciente.

Para los desarrolladores interesados en utilizar la solución Zigbee de NXP, la compañía ofrece una amplia gama de recursos adicionales. Esto incluye documentación detallada, ejemplos de código, y soporte técnico disponible a través de su sitio web y comunidad de desarrolladores. Estos recursos son valiosos para ayudar a los desarrolladores a superar desafíos técnicos y a maximizar el potencial de sus proyectos basados en Zigbee.

3.2.4. Analog Devices (SmartMesh IP Network)

Analog Devices ofrece una solución integral y avanzada con su maletín de evaluación SmartMesh IP, diseñado para facilitar la implementación y evaluación de redes IoT robustas y altamente confiables. Esta tecnología se basa en el estándar IEEE 802.15.4e, conocido por su capacidad para establecer redes mesh autoformadas y autorreparables, ideales para entornos industriales donde la conectividad inalámbrica debe ser resiliente y de bajo consumo energético.

El maletín de evaluación incluye componentes clave que permiten a los usuarios experimentar directamente con la tecnología SmartMesh IP. El nodo Manager es el centro de control de la red, capaz de gestionar hasta 100 nodos individuales, lo que asegura una gestión eficiente de la comunicación y la redundancia de datos. Este nodo central coordina la autoconfiguración y el enrutamiento dinámico de los datos entre los nodos, optimizando la eficiencia y la fiabilidad de la red.

Para la evaluación práctica, el kit proporciona 5 nodos de evaluación que permiten simular diferentes escenarios de despliegue y validar el rendimiento en condiciones variadas. Cada nodo está equipado con radios inalámbricos de baja potencia que operan en la banda de 2.4GHz, garantizando una cobertura adecuada y una comunicación estable en entornos industriales adversos.

Además del nodo Manager y los nodos de evaluación, el maletín incluye un punto de acceso USB que facilita la conexión del sistema SmartMesh IP a computadoras y otros dispositivos externos para configuración y monitoreo. El dispositivo "Eterna" proporciona alimentación y comunicación UART a los nodos, mientras que el adaptador específico para la placa "Eterna" asegura una integración sin complicaciones con los demás componentes del maletín.

Para garantizar una operación continua y sin problemas, el maletín de evaluación viene completo con pilas de botón y todo el cableado necesario. Esto permite a los usuarios configurar rápidamente la red SmartMesh IP y comenzar las pruebas y evaluaciones de rendimiento con mínima preparación.

El precio del maletín de evaluación, que ronda los 3000 dólares, refleja su naturaleza como una herramienta profesional y especializada. Es una inversión importante pero justificada para empresas y desarrolladores que buscan implementar redes IoT robustas y seguras en aplicaciones críticas como monitorización industrial, gestión de activos en tiempo real, y optimización de recursos energéticos.

En resumen, el maletín de evaluación SmartMesh IP de Analog Devices ofrece una solución completa y avanzada para la exploración y desarrollo de redes IoT basadas en SmartMesh IP. Con su capacidad para gestionar múltiples nodos de manera eficiente, su alta fiabilidad y su facilidad de integración, se posiciona como una herramienta indispensable para aquellos que buscan desplegar infraestructuras de IoT escalables y resilientes en entornos industriales exigentes.



Figura 17: Starter Kit Analog Devices SmartMesh IP. (Fuente: ITI)

3.2.5. Adeunis

Adeunis se destaca en el mercado de las comunicaciones LPWAN con una solución enfocada en la red Sigfox, diseñada para ofrecer dispositivos especializados que cumplen funciones específicas dentro del ecosistema IoT. Similar a empresas como NCD o Libelium, Adeunis se especializa en la creación de dispositivos que abordan necesidades concretas de monitorización y control en diversos sectores industriales y comerciales.

Entre los productos destacados de Adeunis se encuentran varios nodos especializados que cubren diferentes aspectos de la monitorización y medición:

- El **Nodo de Temperatura/s** está diseñado para proporcionar mediciones precisas de la temperatura en entornos industriales y comerciales, permitiendo la monitorización continua y la detección de variaciones térmicas críticas.
- El **Nodo de Pulso** facilita la detección y registro de pulsos o eventos discretos, siendo útil para aplicaciones que requieren el conteo preciso de señales eléctricas o el seguimiento de eventos específicos.
- El **Nodo de Energía** está orientado a la medición del consumo energético, proporcionando datos precisos que permiten optimizar el uso de la energía en instalaciones industriales y comerciales, así como mejorar la eficiencia operativa.
- El **Nodo Modbus** facilita la integración con sistemas y dispositivos que utilizan el protocolo Modbus, permitiendo la comunicación y el control eficiente de equipos industriales compatibles con este estándar.
- El **Nodo de Testeo de Red** se utiliza para evaluar y asegurar la cobertura y la calidad de la red SigFox en diferentes ubicaciones y condiciones, garantizando una conectividad robusta y confiable para aplicaciones críticas.

Estos nodos no solo proporcionan mediciones y datos precisos en tiempo real, sino que también son fáciles de integrar en infraestructuras existentes gracias a su compatibilidad con la red SigFox, que ofrece una cobertura extensa y un bajo consumo energético. Esto hace que los productos de Adeunis sean ideales para aplicaciones donde la eficiencia operativa, la fiabilidad y la escalabilidad son fundamentales, como la gestión de activos, la monitorización ambiental, la agricultura inteligente y la automatización industrial.

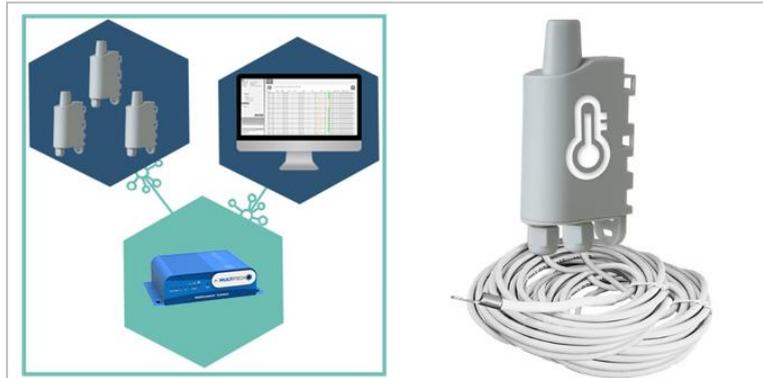


Figura 18: Solución de Adeunis. (Fuente: ITI)

3.2.6. ACKCIO

Ackcio ofrece una solución avanzada de red de sensores inalámbricos diseñada para operar en frecuencias sub-GHz (400 MHz / 700 MHz / 800 MHz / 900 MHz) mediante su tecnología Ackcio Mesh. Esta plataforma está especialmente configurada con una variedad de nodos específicos, cada uno adaptado para cumplir funciones clave en entornos industriales y comerciales.

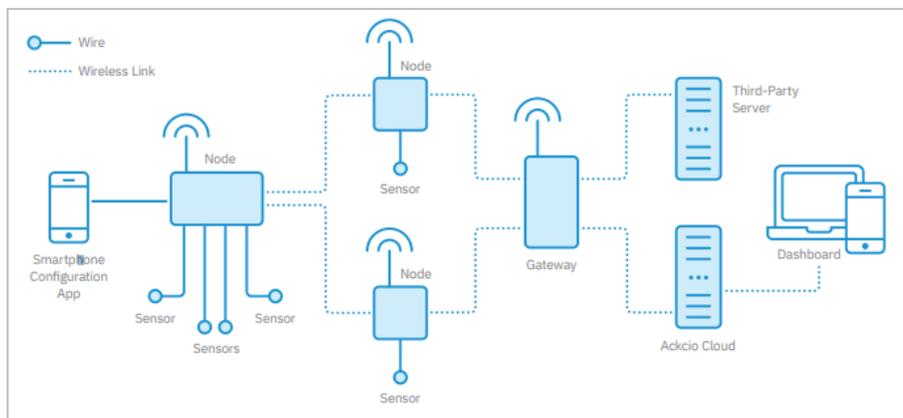


Figura 19: Topología de la red de ACKCIO. (Fuente: ITI)

Los nodos disponibles en el catálogo de Ackcio incluyen:

- **Nodos Analogue:** Diseñados para medir corrientes de 4-20mA con una alta resolución de ADC de 24 bits. Estos nodos son ideales para aplicaciones que requieren una monitorización precisa de corrientes analógicas en tiempo real.
- **Nodos Vibration:** Equipados con sensores cableados capaces de detectar vibraciones en un rango de frecuencias entre 100 y 8000 Hz. Estos nodos son fundamentales para la vigilancia y el mantenimiento predictivo de maquinaria y equipos sensibles a las vibraciones.

- **Nodos Digitales:** Con capacidad para conectar y leer datos según el protocolo Modbus RTU a través de interfaces RS485 y RS232. Estos nodos facilitan la integración con dispositivos industriales compatibles, permitiendo una supervisión eficiente y precisa de datos digitales.
- **Nodo Relay:** Diseñado para controlar dispositivos externos actuando como un relé remoto, proporcionando capacidad de acción sobre otros equipos o sistemas según las necesidades del usuario.

Además, los dispositivos de Ackcio están optimizados para ofrecer largos períodos de operación autónoma con intervalos de muestreo configurables, permitiendo medidas frecuentes. Estos dispositivos tienen una vida útil de hasta 5 años y utilizan una batería estándar de coche de 19000 mAh.

La infraestructura de conectividad de Ackcio se complementa con gateways que ofrecen conectividad 3G/4G y ethernet, asegurando una comunicación fiable y robusta entre los nodos de la red y la plataforma central. Estos gateways están equipados con baterías de respaldo de 18000 mAh, lo que asegura operatividad continua incluso en condiciones adversas.

Ackcio también proporciona una plataforma en la nube y una aplicación móvil que permiten la gestión remota de los datos recolectados por los sensores, ofreciendo a los usuarios acceso instantáneo a información crítica y herramientas de análisis para optimizar operaciones y tomar decisiones informadas en tiempo real.



Figura 20: Solución de ACKCIO. (Fuente: ITI)

En resumen, la solución de Ackcio destaca por su enfoque en la robustez, la precisión y la autonomía de sus dispositivos de red de sensores inalámbricos, proporcionando a las industrias herramientas avanzadas para mejorar la eficiencia operativa, la seguridad y el mantenimiento de activos críticos.

4. Metodología: Tecnologías y herramientas empleadas para el desarrollo de la aplicación web QDSN.

4.1. Herramientas de desarrollo web utilizadas

4.1.1. GitLab

GitLab es una plataforma completa de desarrollo de software que proporciona un conjunto integrado de herramientas para gestionar el ciclo de vida completo del desarrollo de aplicaciones. Se utiliza principalmente para alojar repositorios⁴¹ Git⁴², realizar seguimiento de problemas, implementar pipelines de integración y entrega continua (CI/CD), gestionar el desarrollo colaborativo y mucho más.



Figura 21: Logo de GitLab. (Fuente: [36])

4.1.1.1. Alojamiento de repositorios Git tanto públicos como privados.

Aunque GitLab es una plataforma integral de desarrollo de software que facilita la colaboración y la gestión del ciclo de vida completo del desarrollo de aplicaciones, en este caso, se ha utilizado GitLab como alojamiento del repositorio Git, lo que se refiere al servicio o plataforma que proporciona GitLab para almacenar y gestionar repositorios Git de manera remota. Este servicio es fundamental para el desarrollo de software colaborativo, ya que permite a los desarrolladores trabajar en equipo en un mismo proyecto, compartir código y colaborar de manera efectiva. A continuación, se exponen algunas características clave del alojamiento para repositorios Git:

- Los servicios de alojamiento para repositorios Git permiten a los desarrolladores acceder a sus repositorios desde cualquier lugar a través de Internet. Esto facilita la colaboración entre equipos distribuidos geográficamente y permite a los desarrolladores trabajar desde diferentes ubicaciones.
- Estos servicios suelen proporcionar herramientas para gestionar el acceso al repositorio y controlar quién puede ver, editar o contribuir al código. Esto incluye funciones como permisos de lectura/escritura, ramas protegidas y revisión de código mediante solicitudes de incorporación.
- Los servicios de alojamiento para repositorios Git suelen incluir características de seguridad integradas, como autenticación de dos factores, encriptación de datos y registros de auditoría. Además, garantizan la integridad del código fuente y proporcionan un historial detallado de todos los cambios realizados en el repositorio.

⁴¹ En Git, un repositorio es un lugar donde se almacena el código fuente de un proyecto junto con su historial de cambios. Los desarrolladores pueden clonar un repositorio existente para obtener una copia del código en su propio equipo, realizar cambios locales y luego enviar esos cambios de vuelta al repositorio remoto.

⁴² Git es un sistema de control de versiones distribuido, desarrollado por Linus Torvalds en 2005 para gestionar el desarrollo del kernel de Linux. Permite a los desarrolladores realizar un seguimiento de los cambios en su código fuente a lo largo del tiempo. Esto significa que pueden guardar diferentes versiones de un proyecto y revertir a versiones anteriores si es necesario. Además, es un sistema distribuido, lo que significa que cada desarrollador tiene una copia completa del historial de cambios del proyecto en su propio equipo.

- Estos servicios suelen integrarse con otras herramientas de desarrollo de software, como sistemas de seguimiento de problemas, sistemas de integración continua (CI) y servicios de despliegue automatizado. Esto facilita la automatización de los flujos de trabajo de desarrollo y mejora la productividad del equipo.
- Los servicios de alojamiento para repositorios Git suelen estar diseñados para ser altamente escalables y disponibles, lo que significa que pueden manejar grandes volúmenes de tráfico y garantizar un tiempo de actividad constante. Esto es fundamental para equipos de desarrollo que trabajan en proyectos críticos para el negocio.

4.1.1.2. Gestión de problemas en GitLab.

GitLab proporciona una robusta herramienta de gestión de problemas que permite seguir y resolver problemas, errores y tareas relacionadas con un proyecto de desarrollo de software. Permite a los usuarios crear problemas o tiques detallados, asignarlos a miembros específicos del equipo y establecer prioridades. Los comentarios facilitan la discusión y actualización del estado de los problemas, mientras que las etiquetas y los filtros ayudan a organizar y clasificar los problemas de manera eficiente. GitLab también ofrece notificaciones y menciones para mantener a todos los miembros del equipo informados sobre las discusiones y actualizaciones relacionadas con los problemas. Además, la automatización y los flujos de trabajo personalizados permiten configurar reglas para acciones específicas en los problemas, lo que agiliza el proceso de gestión y resolución de problemas. En resumen, la gestión de problemas en GitLab proporciona una herramienta integral para colaborar y abordar eficientemente los desafíos en el desarrollo de software.

4.1.1.3. CI/CD (Integración y Entrega Continua):

GitLab incluye un poderoso sistema de CI/CD que permite a los equipos automatizar el proceso de construcción, prueba y despliegue de sus aplicaciones. Los pipelines de GitLab permiten definir flujos de trabajo complejos que se activan automáticamente con cada cambio en el repositorio.

CI/CD (Integración Continua/Entrega Continua) es una metodología y práctica en el desarrollo de software que tiene como objetivo automatizar todo el ciclo de vida del desarrollo de software, desde la integración de cambios hasta la entrega de software funcionando, esto ayuda a reducir los tiempos de entrega, minimizar los errores y mejorar la calidad del software, lo que resulta en una mejor experiencia para los usuarios finales y un desarrollo más eficiente, ágil y de calidad.

La Integración Continua (CI) implica la automatización de la construcción y las pruebas de código cada vez que se realizan cambios en el repositorio de código. Los desarrolladores fusionan su código con el repositorio compartido con frecuencia, y luego se ejecutan pruebas automáticas para verificar la integridad y calidad del código. Si las pruebas pasan con éxito, se considera que el código está integrado y listo para ser implementado.

La Entrega Continua (CD) lleva la Integración Continua un paso más allá al automatizar también el proceso de entrega del software. Después de que el código pasa las pruebas en el entorno de integración continua, el proceso de entrega continua se encarga de construir, probar y preparar el software para su implementación en un entorno de producción o de prueba. Esto puede incluir la generación de artefactos de implementación, la ejecución de pruebas adicionales y la creación de entornos de prueba replicables.

4.1.1.4. Wiki y documentación:

La función de Wiki y documentación en GitLab ofrece a los equipos un espacio para documentar y compartir información importante sobre sus proyectos de desarrollo de software. Los usuarios pueden crear y editar páginas Wiki directamente desde GitLab, lo que facilita la colaboración en la creación y actualización de documentación.

La Wiki puede contener información sobre la arquitectura del proyecto, las mejores prácticas de desarrollo, instrucciones de instalación, guías de usuario y cualquier otra información relevante para el proyecto. Además, GitLab proporciona características de seguimiento de cambios que permiten ver el historial de revisiones de una página Wiki y revertir a versiones anteriores si es necesario. Esto ayuda a mantener un registro claro de los cambios en la documentación y garantiza su precisión y relevancia a lo largo del tiempo.

En resumen, la función de Wiki de GitLab proporciona una herramienta útil para documentar y compartir conocimientos dentro de los equipos de desarrollo.

4.1.1.5. Gestión de código de estilo:

La función de Gestión de Código de Estilo en GitLab es una herramienta que ayuda a mantener la consistencia y la calidad del código en un proyecto de desarrollo de software. Permite a los equipos definir reglas de estilo de codificación y aplicarlas automáticamente a los cambios de código realizados en el repositorio. Esto es especialmente útil en equipos grandes donde múltiples desarrolladores contribuyen al mismo código, ya que garantiza que todo el código cumpla con las mismas convenciones y estándares de codificación. Además, la Gestión de Código de Estilo puede configurarse para realizar comprobaciones automáticas de estilo de codificación cada vez que se realiza un commit⁴³ o una solicitud de incorporación de cambios, lo que ayuda a identificar y corregir problemas de estilo de manera temprana en el proceso de desarrollo. En resumen, la Gestión de Código de Estilo en GitLab es una herramienta que ayuda a mantener la calidad y consistencia del código en un proyecto de desarrollo de software, mejorando así su legibilidad, mantenibilidad y colaboración entre los miembros del equipo.

4.1.1.6. Control de acceso y seguridad:

El Control de Acceso y Seguridad en GitLab es una característica crítica para garantizar la protección de los proyectos y la integridad de los datos. Permite a los administradores y propietarios de proyectos controlar quién puede acceder y realizar cambios en los repositorios y otros recursos del proyecto. Esto se logra mediante la asignación de permisos específicos a los usuarios y grupos, lo que determina sus capacidades dentro del proyecto.

GitLab ofrece una variedad de controles de acceso y medidas de seguridad, que incluyen la gestión de usuarios y grupos, así como permisos granulares que permiten definir quién puede ver, editar, eliminar y administrar diferentes recursos del proyecto. Además, proporciona métodos de autenticación como autenticación basada en tokens, LDAP, OAuth y SAML para garantizar la identidad de los usuarios y controlar su acceso al proyecto.

La plataforma también integra herramientas de seguridad, como análisis estático de código, escaneo de dependencias, detección de vulnerabilidades y escaneo de contenedores, que ayudan a identificar y mitigar posibles riesgos de seguridad en el código y las dependencias del proyecto. Además, registra todas las acciones realizadas por los usuarios en el proyecto, lo que permite realizar un seguimiento detallado de quién hizo qué y cuándo, manteniendo así la responsabilidad y la transparencia en el desarrollo del proyecto.

⁴³ En Git, un commit es un registro de un cambio realizado en el repositorio. Cada commit tiene un mensaje asociado que describe el cambio realizado. Los commits permiten a los desarrolladores mantener un registro detallado de los cambios realizados en el proyecto a lo largo del tiempo.

En resumen, el Control de Acceso y Seguridad en GitLab es una característica fundamental que garantiza la protección de los proyectos y la integridad de los datos al proporcionar controles de acceso granulares, medidas de autenticación y autorización, herramientas integradas de seguridad y un registro de auditoría detallado. Esto permite a los equipos de desarrollo trabajar de manera segura y colaborativa en sus proyectos.

4.1.1.6. Integraciones y extensiones:

Las Integraciones y Extensiones en GitLab amplían la funcionalidad de la plataforma y la integran con otras herramientas y servicios populares, lo que enriquece la experiencia de desarrollo y colaboración. Aquí te explico cómo funcionan:

GitLab ofrece una amplia gama de integraciones con herramientas y servicios populares externos que amplían la funcionalidad de la plataforma, como Slack, JIRA, Jenkins y Kubernetes, entre otras. Estas integraciones permiten a los equipos conectar GitLab con sus herramientas favoritas y aprovechar sus capacidades sin tener que cambiar de contexto. Por ejemplo, pueden recibir notificaciones de actividad en GitLab directamente en Slack, vincular problemas de GitLab con tickets en JIRA, o automatizar flujos de trabajo de integración continua con Jenkins.

Además de las integraciones predefinidas, GitLab permite a los usuarios crear extensiones personalizadas mediante el uso de APIs y complementos. Esto significa que pueden desarrollar sus propias integraciones para satisfacer necesidades específicas de su proyecto o equipo. Por ejemplo, pueden crear un complemento para integrar GitLab con una herramienta interna de seguimiento de errores, o desarrollar una integración personalizada con un sistema de gestión de proyectos específico.

En resumen, las Integraciones y Extensiones en GitLab amplían su funcionalidad y aumentan su interoperabilidad con otras herramientas y servicios. Esto permite a los equipos personalizar su entorno de desarrollo y colaboración según sus necesidades específicas, mejorando así la eficiencia y la productividad en el desarrollo de software.

4.1.2. Docker

Docker es una plataforma de código abierto que se utiliza para desarrollar, enviar y ejecutar aplicaciones en contenedores. Ahora, ¿qué son los contenedores? Son entornos ligeros y portátiles que contienen todo lo necesario para ejecutar una aplicación, incluyendo el código, las bibliotecas y las dependencias. Docker utiliza la tecnología de contenedores para permitir a los desarrolladores empacar una aplicación con todas sus partes y luego desplegarla de manera consistente en cualquier entorno, ya sea un portátil, un servidor de desarrollo, un servidor de pruebas o en la nube.



Figura 22: Logo de Docker. (Fuente: [37])

A continuación, se muestran algunos conceptos clave en Docker:

- **Imagen de Docker**: Una imagen de Docker es un paquete autónomo y ligero que contiene todo lo necesario para ejecutar una aplicación, incluyendo el código, las bibliotecas, las dependencias, las variables de entorno y la configuración del entorno de ejecución. Las imágenes de Docker se utilizan como base para crear contenedores, que son instancias en ejecución de esas imágenes.

Las imágenes de Docker se construyen a partir de un archivo llamado Dockerfile, que contiene una serie de instrucciones que Docker utiliza para crear la imagen. Estas instrucciones describen los pasos necesarios para configurar el entorno de ejecución de la aplicación dentro del contenedor.

Una vez que se ha construido una imagen de Docker, puede ser compartida y reutilizada en diferentes entornos y por diferentes equipos de desarrollo. Esto hace que las imágenes de Docker sean una forma eficiente y consistente de distribuir y ejecutar aplicaciones en diferentes plataformas y entornos de implementación.

- **Contenedor Docker**: Un contenedor Docker es una instancia en tiempo de ejecución de una imagen de Docker. Es decir, un contenedor es una entidad aislada que contiene todo lo necesario para ejecutar una aplicación de forma eficiente y segura.

Los contenedores Docker son ligeros, portátiles y eficientes en cuanto a recursos, ya que comparten el núcleo del sistema operativo y utilizan tecnologías como cgroups y namespaces⁴⁴ para proporcionar un entorno aislado y seguro para la ejecución de aplicaciones. Esto hace que los contenedores Docker sean ideales para desarrollar, distribuir y ejecutar aplicaciones de manera consistente en diferentes entornos, desde el portátil del desarrollador hasta el servidor de producción en la nube.

Cuando se inicia un contenedor Docker, se toma una imagen de Docker como base y se crea una instancia de esa imagen. El contenedor se ejecuta de manera aislada en el sistema operativo host, compartiendo el kernel⁴⁵ del sistema operativo con otros contenedores y con acceso a los recursos del sistema asignados por Docker.

- **Dockerfile**: Es un archivo de texto plano que contiene una serie de instrucciones que Docker utiliza para construir una imagen de Docker. Estas instrucciones describen los pasos necesarios para configurar el entorno de ejecución de una aplicación dentro de un contenedor, instalar dependencias, copiar archivos y configurar variables de entorno, entre otras cosas.

⁴⁴ En Docker, los **namespaces** son una característica clave del kernel de Linux que permite la separación y el aislamiento de procesos. Proporcionan un entorno independiente para cada contenedor, lo que permite que múltiples contenedores se ejecuten simultáneamente sin interferir entre sí.

⁴⁵ El kernel es el núcleo o la parte central del sistema operativo de una computadora. Es el componente que se encarga de gestionar los recursos del hardware y proporcionar servicios básicos para que otros programas puedan ejecutarse. Algunas de las responsabilidades del kernel incluyen la gestión de la memoria, la administración de los procesos, la gestión de archivos y el control de los dispositivos de entrada y salida.

Algunas de las instrucciones más comunes que se pueden utilizar para personalizar y configurar una imagen Docker con sus necesidades específicas son las siguientes:

- **FROM:** Especifica la imagen base que se utilizará para construir la nueva imagen.
- **LABEL:** Agrega metadatos a la imagen, como el nombre del autor o información de contacto
- **RUN:** Ejecuta comandos en la imagen durante el proceso de construcción.
- **COPY:** Copia archivos locales al sistema de archivos del contenedor.
- **WORKDIR:** Establece el directorio de trabajo dentro del contenedor.
- **ENV:** Establece variables de entorno dentro del contenedor.
- **EXPOSE:** Indica los puertos en los que la aplicación escuchará las conexiones.
- **VOLUME:** Crea un punto de montaje para almacenar datos persistentes fuera del contenedor.
- **CMD:** Especifica el comando que se ejecutará cuando se inicie un contenedor basado en esta imagen.
- **ENTRYPOINT:** Similar a CMD, pero especifica un comando que siempre se ejecutará cuando se inicie el contenedor, incluso si se proporcionan argumentos al contenedor en el momento de la ejecución.

Estas son solo algunas de las instrucciones más comunes, pero hay más disponibles que se pueden encontrar en una lista completa de las instrucciones admitidas en la documentación oficial de Docker. Cada instrucción tiene su propia función y se puede utilizar para personalizar y configurar la imagen de Docker de acuerdo con las necesidades específicas.

A continuación, en el Fragmento de código 1, se muestra un ejemplo básico de un Dockerfile en el que se crea una imagen Docker que tendrá una aplicación programada en Python:

```
1 # Uso de una imagen base
2 FROM ubuntu:latest
3
4 # Etiqueta para el autor
5 LABEL maintainer="tudireccion@email.com"
6
7 # Actualización del sistema e instalación de dependencias
8 RUN apt-get update && apt-get install -y \
9     python3 \
10    python3-pip \
11    && rm -rf /var/lib/apt/lists/*
12
13 # Establecimiento del directorio de trabajo
14 WORKDIR /app
15
16 # Copia de archivos locales al contenedor
17 COPY . /app
18
19 # Instalación de dependencias de Python
20 RUN pip3 install -r requirements.txt
21
22 # Exposición del puerto que utiliza la aplicación
23 EXPOSE 8080
24
25 # Comando para ejecutar la aplicación
26 CMD ["python3", "app.py"]
```

Fragmento de código 1: Ejemplo de Dockerfile para una aplicación de Python. (Fuente: Elaboración propia)

Con el Dockerfile anterior se está construyendo una imagen basada en la última versión de Ubuntu⁴⁶, donde se instala Python 3 y pip. Después se copia el directorio actual, donde se supone que hay un código desarrollado en Python al directorio `'/app'` dentro del contenedor y posteriormente ejecuta dicha aplicación `'app.py'`

- **Docker Compose:** Es una herramienta que permite definir y ejecutar aplicaciones Docker con múltiples contenedores. Con Docker Compose, se puede definir la configuración de una aplicación en un archivo YAML⁴⁷, que incluye la configuración de los servicios, las redes y los volúmenes, y luego ejecutar tu aplicación con un solo comando.

⁴⁶ Ubuntu es un sistema operativo de código abierto basado en Linux, desarrollado y mantenido por Canonical Ltd. Es una de las distribuciones de Linux más populares y ampliamente utilizadas, especialmente en el ámbito de los servidores y la informática de escritorio.

⁴⁷ Un **fichero YAML** es un archivo de texto que utiliza el formato YAML para la configuración de datos, permitiendo una estructura jerárquica y legible por humanos, comúnmente usado en aplicaciones y despliegues de software.

En un archivo YAML llamado `'docker-compose.yml'`, se definen los servicios que componen la aplicación. Cada servicio puede estar compuesto por una o más imágenes de Docker, y puede incluir configuraciones adicionales como variables de entorno, volúmenes, puertos expuestos y dependencias.

Además de gestionar los contenedores, Docker Compose también puede crear y gestionar automáticamente volúmenes y redes para los servicios. Esto simplifica la configuración de almacenamiento persistente y la comunicación entre contenedores.

Docker Compose permite utilizar variables de entorno en el archivo `'docker-compose.yml'`, lo que permite parametrizar la configuración de los servicios. También es posible utilizar la sustitución de variables para referenciar valores definidos en el archivo `'env'`.

Una vez que se ha definido el archivo `'docker-compose.yml'`, se puede utilizar el comando `'docker-compose up'` para iniciar todos los contenedores definidos en el archivo. Docker Compose se encargará de crear, iniciar y conectar los contenedores según la configuración especificada en el archivo YAML.

Por último, Docker Compose proporciona una serie de comandos para administrar la aplicación, como `'docker-compose ps'` para ver el estado de los servicios, `'docker-compose stop'` para detener los contenedores, y `'docker-compose restart'` para reiniciar los contenedores. También se pueden escalar servicios utilizando el comando `'docker-compose scale'`.

A continuación, en el Fragmento de código 2, se muestra un ejemplo básico de un `'docker-compose.yml'`, donde se definen dos servicios que componen la aplicación (`'web'` y `'db'`):

- **web**: Este servicio utiliza la imagen de Node.js. Expone el puerto 3000 del contenedor al puerto 3000 del host. Monta el directorio local `'/app'` en el directorio `'/app'` del contenedor para permitir el desarrollo en tiempo real. También establece la variable de entorno `'NODE_ENV'` en "production" y especifica que depende del servicio `'db'`.
- **db**: Este servicio utiliza la imagen de MongoDB más reciente. Expone el puerto 27017 del contenedor al puerto 27017 del host. También crea un volumen llamado `'mongo_data'` para almacenar los datos de MongoDB⁴⁸.

⁴⁸ MongoDB es un sistema de base de datos NoSQL (o "Not Only SQL") orientado a documentos. En lugar de almacenar los datos en tablas como lo hace una base de datos relacional, MongoDB almacena los datos en documentos JSON (BSON, en realidad) dentro de colecciones. Esto permite una flexibilidad y escalabilidad significativas, ya que los documentos en una colección no necesitan tener la misma estructura, a diferencia de las filas en una tabla de base de datos relacional.

```

1  version: '3.8'
2
3  services:
4    web:
5      image: node:14
6      ports:
7        - "3000:3000"
8      volumes:
9        - ./app:/app
10     environment:
11       - NODE_ENV=production
12     depends_on:
13       - db
14
15    db:
16      image: mongo:latest
17      ports:
18        - "27017:27017"
19      volumes:
20        - mongo_data:/data/db
21
22  volumes:
23    mongo_data:

```

Fragmento de código 2: Ejemplo de docker-compose.yml. (Fuente: Elaboración propia)

- **Docker Swarm y Kubernetes**: Son herramientas de orquestación de contenedores que permiten gestionar, escalar y desplegar aplicaciones Docker en un clúster de servidores de manera eficiente. Estas herramientas te permiten automatizar la implementación, la escalabilidad, y el manejo de la disponibilidad de tus aplicaciones en contenedores.
 - Docker Swarm: es una herramienta de orquestación de contenedores integrada en Docker Engine. Permite crear y gestionar clústeres de contenedores Docker utilizando una API simple y familiar.
 - Kubernetes: es una plataforma de código abierto desarrollada por Google para la gestión automatizada de aplicaciones en contenedores. Ofrece características avanzadas de orquestación y escalabilidad, diseñadas para gestionar aplicaciones en entornos de producción a gran escala.

En resumen, Docker simplifica el proceso de desarrollo, implementación y administración de aplicaciones al proporcionar un entorno consistente y portátil en todo el ciclo de vida de la aplicación.

4.1.3. Vue.js

Vue es un framework⁴⁹ de JavaScript ⁵⁰que se utiliza para construir interfaces de usuario dinámicas y eficientes. Se basa en tecnologías web estándar como HTML⁵¹, CSS ⁵²y JavaScript, pero agrega un modelo de programación declarativo y basado en componentes que facilita el desarrollo de interfaces de usuario, ya sean simples o complejas.



Figura 23: Logo del Framework Vue.js. (Fuente: [34])

A continuación, se muestra un pequeño ejemplo en el que se demuestran las dos características principales de Vue.js, Rendering Declarativo y Reactividad.

```
1 import { createApp, ref } from 'vue'
2
3 createApp({
4   setup() {
5     return {
6       count: ref(0)
7     }
8   }
9 }).mount('#app')
```

Fragmento de código 3: Ejemplo de código Vue.js (archivo.vue). (Fuente: [34])

```
1 <div id="app">
2   <button @click="count++">
3     Count is: {{ count }}
4   </button>
5 </div>
```

Fragmento de código 4: Ejemplo de código Vue.js (archivo.html). (Fuente: [34])

⁴⁹ Un framework es una estructura o conjunto de herramientas y convenciones predefinidas que proporcionan un marco de trabajo para desarrollar aplicaciones de software. Estas herramientas y convenciones suelen incluir bibliotecas, APIs, patrones de diseño, estándares de codificación y una arquitectura base que facilita y agiliza el proceso de desarrollo.

⁵⁰ JavaScript es un lenguaje de programación utilizado para agregar interactividad y dinamismo a las páginas web, así como en otros entornos de desarrollo de aplicaciones.

⁵¹ HTML (HyperText Markup Language) es el lenguaje estándar utilizado para crear y diseñar páginas web. Se compone de una serie de elementos o etiquetas que definen la estructura y el contenido de una página web. Estas etiquetas permiten agregar texto, imágenes, enlaces, formularios, videos y otros tipos de contenido multimedia, así como también dar formato y estilo a la página mediante el uso de atributos y hojas de estilo CSS. HTML es fundamental en el desarrollo web y es interpretado por los navegadores web para mostrar el contenido de las páginas en línea.

⁵² CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para controlar el aspecto y el formato de los documentos HTML. Permite definir la presentación visual de los elementos HTML, como el color, la tipografía, el espaciado, el tamaño y la posición en la página. CSS funciona mediante la selección de elementos HTML y la aplicación de reglas de estilo a esos elementos, ya sea mediante atributos en línea, hojas de estilo internas o hojas de estilo externas.

Una de las características principales de Vue es su capacidad para el Rendering Declarativo, lo que significa que extiende el HTML estándar con una sintaxis de plantilla que permite describir de manera declarativa la salida HTML basada en el estado de JavaScript. Esto facilita la creación de interfaces de usuario de manera más intuitiva y comprensible.

Vue es tanto un framework como un ecosistema que abarca una amplia gama de características esenciales para el desarrollo frontend. Sin embargo, la web es extremadamente diversa y los proyectos web pueden variar significativamente en su forma y escala. Con esto en mente, Vue ha sido diseñado para ser flexible y adaptable de manera incremental. Según su propósito, Vue puede emplearse de diversas formas:

- Mejora del HTML estático sin un paso de compilación
- Como componentes web en cualquier página
- Solicitud de una sola página (SPA)
- Fullstack / Representación del lado del servidor (SSR)
- Jamstack / Generación de sitios estáticos (SSG)
- Dirigido a computadoras de escritorio, dispositivos móviles, WebGL e incluso terminales

Algunas características de Vue.js son:

- **Vue.js se basa en un enfoque de desarrollo de componentes.** Los componentes son bloques reutilizables de código que encapsulan la funcionalidad y la apariencia de una parte específica de la interfaz de usuario. Esto promueve la modularidad y facilita la construcción y mantenimiento de aplicaciones complejas.
- **Flujo unidireccional de datos:** En Vue.js, el flujo de datos sigue un patrón unidireccional, lo que significa que los datos fluyen desde el componente principal hacia los componentes secundarios a través de props, y los cambios en los datos se comunican hacia arriba a través de eventos.
- **Reactividad:** Una de las características principales de Vue.js es su sistema de reactividad, ya que rastrea los cambios en el estado de JavaScript y actualiza eficientemente el DOM⁵³ cuando ocurren cambios. Esto significa que los cambios realizados en los datos se reflejan automáticamente en la interfaz de usuario y viceversa.

No es necesario actualizar manualmente la interfaz de usuario cuando los datos cambian, lo que hace que el desarrollo sea más eficiente y menos propenso a errores. Vue utiliza directivas para enlazar datos de la aplicación con el DOM. Algunas directivas comunes incluyen ``v-bind``, ``v-if``, ``v-for`` y ``v-on``.

- **Estado de la aplicación:** Vue.js proporciona una forma conveniente de gestionar el estado de la aplicación a través de objetos conocidos como instancias Vue. Estas instancias contienen los datos de la aplicación y los métodos para manipular esos datos.

⁵³ El DOM (Document Object Model) es una interfaz de programación que representa la estructura de un documento HTML o XML como un árbol de nodos. Cada nodo en el árbol representa parte del documento, como elementos HTML, atributos, texto y comentarios. El DOM permite a los programas o scripts acceder y manipular dinámicamente el contenido, la estructura y el estilo de un documento web. Esto significa que puedes escribir scripts (como JavaScript) para cambiar el contenido de una página web, modificar su estructura, agregar o eliminar elementos, cambiar estilos y responder a eventos del usuario, todo en tiempo real, sin necesidad de recargar la página. Esto es lo que hace posible la interactividad y la dinámica en las páginas web modernas.

- **Rutas y enrutamiento:** Para construir aplicaciones de una sola página (SPA), Vue.js ofrece un enrutador oficial llamado Vue Router. Con Vue Router, los desarrolladores pueden definir rutas y navegar entre diferentes componentes de manera programática o mediante enlaces.

Otro detalle interesante de Vue es su enfoque incremental y progresivo. A diferencia de otros frameworks que pueden imponer una estructura rígida y compleja desde el principio, Vue permite que los desarrolladores lo adopten de manera gradual en sus proyectos existentes o lo utilicen para construir nuevas funcionalidades en aplicaciones más grandes.

Esto se debe a que Vue se puede integrar fácilmente en proyectos existentes, lo que significa que puedes agregar Vue a una página web existente y comenzar a beneficiarte de sus características, como la reactividad y la composición de componentes, sin tener que reescribir todo el código.

Además, Vue ofrece una curva de aprendizaje suave y una sintaxis clara y concisa que facilita la comprensión y el uso del framework, especialmente para desarrolladores que son nuevos en el desarrollo web o en el uso de frameworks de JavaScript. La documentación de Vue es detallada y accesible, proporcionando ejemplos claros y recursos útiles que ayudan a los nuevos usuarios a familiarizarse rápidamente con sus conceptos fundamentales.

Esta flexibilidad y facilidad de uso hacen que Vue sea una opción atractiva para una amplia gama de proyectos, desde pequeñas aplicaciones hasta aplicaciones complejas. Vue es capaz de manejar desde simples sistemas hasta sistemas completos de frontend, lo que lo convierte en una opción versátil para empresas que buscan escalar sus aplicaciones sin complicaciones.

En resumen, Vue proporciona una manera poderosa y flexible de construir interfaces de usuario, permitiendo a los desarrolladores crear aplicaciones web modernas eficientemente.

4.1.4. Entorno Linux simulado con el subsistema de Windows para Linux (WSL)

El Subsistema de Windows para Linux (WSL) es una herramienta poderosa desarrollada por Microsoft que permite ejecutar un entorno GNU/Linux directamente en Windows, sin necesidad de una máquina virtual o arranque dual. WSL ofrece a los usuarios la capacidad de ejecutar binarios de Linux nativos en Windows a través de una capa de compatibilidad, proporcionando una solución eficiente para los desarrolladores que desean trabajar en un entorno Linux sin abandonar el ecosistema Windows.

WSL presenta una compatibilidad robusta, permitiendo la ejecución de binarios ELF64 de Linux⁵⁴ directamente en Windows. Los usuarios pueden instalar y ejecutar múltiples distribuciones de Linux, como Ubuntu, Debian y Fedora, disponibles en la Microsoft Store. Esta característica es particularmente útil para desarrolladores que necesitan trabajar con diferentes entornos Linux para diversas tareas y proyectos.

Una de las principales ventajas de WSL es la integración del sistema de archivos. Los usuarios pueden acceder a los sistemas de archivos de Windows desde el entorno Linux y viceversa, lo que facilita la gestión y manipulación de archivos entre ambos sistemas. En WSL 2, el rendimiento del sistema de archivos se ha mejorado significativamente, ofreciendo una experiencia más fluida y rápida en comparación con WSL 1.

⁵⁴ Los binarios ELF64 (Executable and Linkable Format 64-bit) son el formato estándar para ejecutables, bibliotecas y objetos en sistemas Linux de 64 bits. Estos archivos contienen secciones que incluyen código ejecutable, datos, encabezados de sección y símbolos necesarios para la ejecución y enlace de programas. ELF64 facilita la compatibilidad entre diferentes plataformas y permite la carga dinámica de bibliotecas, optimizando la gestión de memoria y el rendimiento del sistema operativo.

En términos de redes y comunicación, las herramientas de Linux pueden acceder a la red de manera nativa, como si estuvieran ejecutándose en un entorno Linux estándar. Además, WSL permite la interoperabilidad entre herramientas de Windows y Linux, permitiendo que los comandos de Windows se ejecuten desde la línea de comandos de Linux y viceversa. Esta capacidad mejora la flexibilidad y la eficiencia, ya que los usuarios pueden utilizar las mejores herramientas disponibles en ambos sistemas operativos.

WSL 2 introduce un kernel⁵⁵ de Linux completo que se ejecuta en una máquina virtual ligera, lo que mejora la compatibilidad y el rendimiento en comparación con WSL 1. Esta versión proporciona un rendimiento de sistema de archivos superior y operaciones de red más eficientes, además de ser totalmente compatible con aplicaciones y herramientas de Linux. El uso de un kernel real de Linux asegura que las aplicaciones que dependen de características específicas del kernel funcionen correctamente, eliminando muchas de las limitaciones presentes en WSL 1.

Para instalar y configurar WSL, los usuarios deben habilitar la característica desde PowerShell con privilegios de administrador. Con el comando `wsl --install`, se instala WSL 2 junto con la distribución predeterminada de Ubuntu. Los usuarios pueden instalar otras distribuciones desde la Microsoft Store y configurar WSL 2 como la versión predeterminada utilizando el comando `wsl --set-default-version 2`. Además, se pueden convertir distribuciones a WSL 2 y listar las distribuciones instaladas para una gestión más eficiente del entorno.

WSL es particularmente útil para el desarrollo de software y administración de sistemas. Permite a los desarrolladores trabajar en un entorno Linux mientras utilizan aplicaciones y herramientas de Windows, facilitando el desarrollo multiplataforma. Los administradores de sistemas también se benefician de la capacidad de ejecutar scripts y utilizar herramientas de Linux directamente en Windows, mejorando la eficiencia y flexibilidad en la gestión de sistemas.

La integración de WSL con Docker y otras tecnologías de contenedores también facilita la portabilidad y consistencia de los entornos de desarrollo y despliegue. Los contenedores Docker pueden ejecutarse en WSL 2, permitiendo a los desarrolladores crear y probar aplicaciones en un entorno que refleja con precisión su entorno de producción, lo que es de especial utilidad en el desarrollo de este proyecto.

En conclusión, WSL es una herramienta revolucionaria que permite a los usuarios de Windows ejecutar un entorno Linux completo sin la necesidad de una máquina virtual o arranque dual. Al combinar lo mejor de ambos mundos, WSL mejora la productividad y facilita el desarrollo multiplataforma, proporcionando una solución versátil y eficiente para desarrolladores y administradores de sistemas.

⁵⁵ Un kernel es el núcleo central de un sistema operativo, responsable de gestionar los recursos del hardware y proporcionar servicios esenciales a las aplicaciones. Actúa como un puente entre el software y el hardware, manejando tareas como la administración de la memoria, la gestión de procesos, la comunicación entre dispositivos y la interfaz con el sistema de archivos. El kernel asegura que diferentes programas y usuarios puedan operar de manera segura y eficiente en un entorno informático.

4.2. Elección del Framework

La elección del Framework a utilizar para desarrollar la aplicación se basará en la comparación entre Vue.js, Angular y React.js basada en varios aspectos clave:

- Curva de Aprendizaje:
 - Vue.js tiende a tener una curva de aprendizaje más suave en comparación con Angular y React.js. Su sintaxis es intuitiva y fácil de entender para aquellos que están familiarizados con HTML, CSS y JavaScript.
 - Angular puede resultar más complejo debido a su arquitectura basada en inyección de dependencias y conceptos como módulos, servicios y directivas.
 - React.js puede ser desafiante para aquellos nuevos en el paradigma de programación declarativa y el uso de JSX (JavaScript XML).
- Rendimiento (Como puede observarse en la Figura 24 y en la Figura 25):
 - Los tres frameworks tienen un rendimiento excelente, sin embargo, Angular y Vue.js presentan un enfoque más orientado hacia la optimización, ofreciendo herramientas integradas para mejorar el rendimiento de la aplicación.
 - React.js, siendo solo la capa de vista, puede ser más flexible en términos de optimización, pero requiere que el desarrollador tome decisiones conscientes sobre cómo optimizar la aplicación.
- Tamaño de la Comunidad y Ecosistema:
 - React.js tiene una comunidad masiva respaldada por Facebook y una gran cantidad de bibliotecas y herramientas de terceros disponibles.
 - Vue.js ha experimentado un crecimiento significativo en su comunidad en los últimos años y tiene una serie de bibliotecas y herramientas que lo respaldan.
 - Angular, siendo respaldado por Google, tiene una comunidad sólida y un ecosistema maduro, aunque es posible que no sea tan grande como las comunidades de React.js y Vue.js.
- Flexibilidad y Escalabilidad:
 - Vue.js es conocido por su enfoque incremental, lo que significa que es fácil integrarlo en proyectos existentes y escalar gradualmente a medida que la aplicación crece.
 - Angular tiene una estructura más definida y es más adecuado para proyectos grandes y complejos que necesitan una arquitectura sólida y un conjunto completo de características listas para usar.
 - React.js es muy flexible y se puede utilizar en una amplia variedad de proyectos, desde pequeñas aplicaciones hasta aplicaciones empresariales a gran escala, pero requiere más decisiones de arquitectura por parte del desarrollador.

| Name Duration for... | vue- v3.2.47 | angular- v15.0.1 | react- v18.2.0 |
|---|-------------------------------|-------------------------------|-------------------------------|
| Implementation notes | | | |
| Implementation link | code | code | code |
| create rows creating 1,000 rows (5 warmup runs). | 42.0 ± 0.7 (1.18) | 45.7 ± 0.5 (1.29) | 47.5 ± 0.6 (1.34) |
| replace all rows updating all 1,000 rows (5 warmup runs). | 44.8 ± 0.5 (1.16) | 48.2 ± 0.6 (1.26) | 52.8 ± 0.3 (1.37) |
| partial update updating every 10th row for 1,000 rows (3 warmup runs). 16 x CPU slowdown. | 103.5 ± 3.4 (1.18) | 96.7 ± 2.5 (1.10) | 128.1 ± 3.0 (1.46) |
| select row highlighting a selected row. (5 warmup runs). 16 x CPU slowdown. | 22.1 ± 1.3 (2.31) | 15.6 ± 1.0 (1.64) | 39.3 ± 0.6 (4.12) |
| swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown. | 29.0 ± 0.8 (1.17) | 166.0 ± 1.0 (6.71) | 163.5 ± 0.7 (6.61) |
| remove row removing one row. (5 warmup runs). 4 x CPU slowdown. | 45.8 ± 1.1 (1.20) | 42.3 ± 1.2 (1.10) | 47.7 ± 1.3 (1.25) |
| create many rows creating 10,000 rows. (5 warmup runs with 1k rows). | 475.3 ± 1.5 (1.19) | 474.2 ± 1.9 (1.18) | 661.5 ± 2.4 (1.65) |
| append rows to large table appending 1,000 to a table of 10,000 rows. 2 x CPU slowdown. | 94.7 ± 0.8 (1.14) | 101.7 ± 0.8 (1.23) | 115.0 ± 0.7 (1.39) |
| clear rows clearing a table with 1,000 rows. 8 x CPU slowdown. (5 warmup runs). | 34.7 ± 1.2 (1.21) | 61.1 ± 1.7 (2.13) | 37.9 ± 1.0 (1.32) |
| geometric mean of all factors in the table | 1.27 | 1.59 | 1.87 |
| compare: Green means significantly faster, red significantly slower | com- pare | com- pare | com- pare |

Figura 24: Comparativa de tiempos entre React, Angular y Vue. (Fuente:[35])

Comparar Vue.js, Angular y React en términos de rendimiento y uso de recursos proporciona una visión clara de cómo se comportan estos frameworks en diferentes escenarios de desarrollo de aplicaciones web. Vue.js se destaca por su eficiencia en el uso de memoria y CPU, gracias a su enfoque ligero del virtual DOM y su modularidad. Esto lo hace ideal para aplicaciones de tamaño mediano y pequeño donde se requiere una respuesta rápida y una experiencia de usuario fluida. Su capacidad para minimizar las operaciones de actualización y gestionar eficazmente los componentes contribuye a un rendimiento optimizado.

Angular, por otro lado, es conocido por su robustez y estructura, pero esto también se refleja en un uso potencialmente mayor de memoria y CPU en comparación con Vue.js y React. Aunque las últimas versiones de Angular han mejorado significativamente en términos de rendimiento, especialmente para aplicaciones empresariales complejas, su arquitectura más compleja puede requerir ajustes adicionales para optimizar el uso de recursos.

React ofrece un equilibrio único entre rendimiento y flexibilidad. Con su enfoque en el virtual DOM y actualizaciones eficientes de componentes, React generalmente muestra un consumo moderado de recursos. Es adecuado para una amplia gama de aplicaciones, desde proyectos pequeños hasta grandes plataformas dinámicas, gracias a su capacidad para manejar estados complejos y su gran comunidad de soporte que impulsa constantes mejoras de rendimiento.

En términos de gestión de estado, Vue.js proporciona Vuex para una administración centralizada, Angular utiliza servicios y observables, mientras que React emplea Context API y bibliotecas como Redux. Esta diversidad permite a los desarrolladores seleccionar la mejor opción según las necesidades específicas del proyecto y las habilidades del equipo. En resumen, la elección entre Vue.js, Angular y React depende de las prioridades de rendimiento, la complejidad del proyecto y las preferencias del equipo de desarrollo.

| Name | vue-v3.2.47 | angular-v15.0.1 | react-v18.2.0 |
|---|----------------|-----------------|----------------|
| ready memory Memory usage after page load. | 0.9 (1.41) | 1.6 (2.60) | 1.1 (1.88) |
| run memory Memory usage after adding 1,000 rows. | 3.7 (2.09) | 4.7 (2.62) | 5.0 (2.78) |
| update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times | 3.8 (1.95) | 4.7 (2.45) | 5.5 (2.83) |
| creating/clearing 1k rows (5 cycles) Memory usage after creating and cleaning 1000 rows 5 times | 1.2 (1.73) | 2.3 (3.35) | 2.0 (2.83) |
| run memory 10k Memory usage after adding 10,000 rows. | 27.7 (2.42) | 29.8 (2.61) | 36.2 (3.17) |
| geometric mean of all factors in the table | 1.89 | 2.71 | 2.66 |

Figura 25: Comparativa de rendimiento entre React, Angular y Vue. (Fuente: [35])

En conclusión, para una aplicación con un número tan reducido de usuarios (1-5 usuarios), la prioridad principal no debería ser el rendimiento en términos de uso de memoria y CPU, ya que la carga sobre el servidor será mínima. En este caso, sería más relevante considerar otros aspectos como la facilidad de desarrollo, la curva de aprendizaje, la productividad del equipo y el ecosistema de soporte.

Dado este contexto, Vue.js podría ser una elección óptima. Vue.js es conocido por su simplicidad y facilidad de integración en proyectos existentes. Es ligero y tiene un tamaño pequeño, lo que facilita su carga rápida en el navegador del usuario final. Además, Vue.js proporciona un enfoque gradual para adoptar características avanzadas, lo cual es beneficioso cuando se trabaja en un proyecto pequeño con un equipo pequeño.

4.3. Entorno de Desarrollo

El primer paso para preparar el entorno de desarrollo es clonar el repositorio, para ello hay que obtener acceso al repositorio de GitLab donde se encuentra almacenado el código. Una vez se tiene acceso al repositorio hay que decidir si se trabajará con SSH o HTTPS, para lo que hay que configurar la autenticación, es decir, para SSH se debe configurar una SSH Key y para el caso de HTTPS se debe configurar un Access Token. Como en este caso se utiliza HTTPS, a continuación, se explica cómo configurar un Personal Access Token:

1. El primer paso es ir a **Preferences** en el perfil de GitLab y después ir **Access Tokens**. Una vez en esa sección, se debe ir a la sección **Personal Access Tokens** y ahí seleccionar la opción **Add new token**.

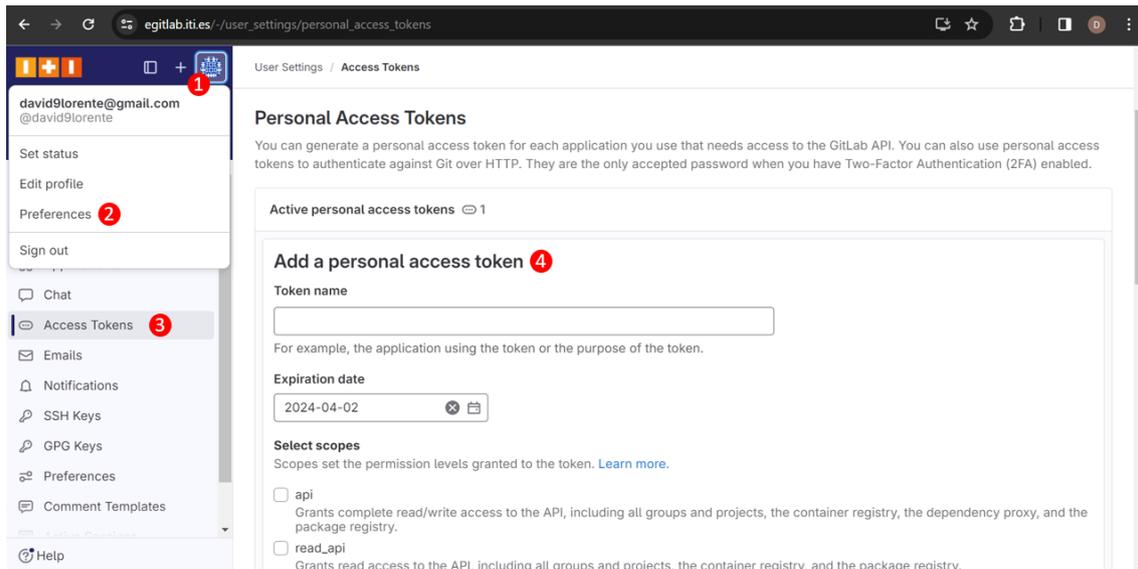


Figura 26: Pasos a seguir para crear un Personal Access Token en GitLab.
(Fuente: Elaboración propia a partir de la interfaz de usuario de GitLab)

2. Una vez configurado el Access Token, el siguiente paso es crear una carpeta donde se clonará el repositorio y, posteriormente, clonar el repositorio, para ello, después de crear la carpeta, se abrirá una terminal en esa carpeta y se ejecutarán los siguientes comandos:

```
>> git clone https://<usuario>:<AccessToken>@egitlab.iti.es/qdsn/frontend_src.git  
>> git clone https://<usuario>:<AccessToken>@egitlab.iti.es/qdsn/backend_src.git
```

En segundo lugar, tras clonar el repositorio, hay que construir las imágenes de Docker que se van a utilizar para el backend, ya que se necesitan tres imágenes para ejecutar la aplicación en el entorno de desarrollo: una para el backend, una para la base de datos de mongo y otra para simular los datos de los nodos.

Para utilizar Docker se hará uso de un sistema Linux, por lo que, si no se dispone de un sistema con distribución Linux, se podrá hacer uso de una máquina virtual o, como en este caso, se puede hacer uso de WSL, anteriormente explicado. Para ello se ejecutan los siguientes comandos:

3. En primer lugar, se debe instalar Ubuntu en WSL. WSL permite ejecutar un entorno Linux en Windows sin la necesidad de una máquina virtual completa. El parámetro `-d` especifica la distribución, en este caso, Ubuntu.

```
>> wsl --install -d Ubuntu
```

- Una vez instalado Ubuntu, se recomienda cambiar la versión de WSL para la instalación de Ubuntu a WSL 2, que ofrece un kernel Linux completo y mejor rendimiento en comparación con WSL 1. Este paso podría no ser necesario si se tiene WSL 2 (*wsl -l -v*).

```
>> wsl --set-version ubuntu 2
```

Una vez instalado Ubuntu se puede lanzar y configurar con los siguientes comandos:

- El siguiente comando permite al usuario especificado (*username*, que debe ser reemplazado por el nombre de usuario elegido) ejecutar comandos con *sudo* sin necesidad de ingresar una contraseña. Para encontrar el nombre de usuario actual, se puede usar el comando *echo \$USER*.

```
>> sudo sh -c "echo 'username ALL=NOPASSWD: ALL' >> /etc/sudoers" (replace username with your user: echo $USER)
```

- Actualiza la lista de paquetes disponibles en los repositorios de Ubuntu, lo cual es importante para asegurarse de que se descarguen las versiones más recientes.

```
>> sudo apt-get update
```

- Descarga y añade la clave GPG de Docker para verificar la autenticidad de los paquetes descargados desde el repositorio de Docker.

```
>> curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

- Añade el repositorio oficial de Docker a la lista de fuentes de paquetes. *\$(lsb_release -cs)* se reemplaza automáticamente por el nombre de la distribución de Ubuntu que estás usando (como focal para Ubuntu 20.04).

```
>> sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

- Instala Docker Compose, una herramienta que permite definir y ejecutar aplicaciones Docker de múltiples contenedores.

```
>> sudo apt-get install docker-compose
```

- Instala Docker Community Edition (CE), la versión gratuita y de código abierto de Docker, que incluye Docker Engine, CLI y el demonio de Docker.

```
>> sudo apt-get install docker-ce
```

- Añade el usuario al grupo *docker*, permitiendo ejecutar comandos de Docker sin *sudo*. Esto es útil para evitar la necesidad de usar privilegios elevados para cada comando Docker.

```
>> sudo gpasswd -a $USER docker
```

- Aplica los cambios de grupo inmediatamente sin necesidad de cerrar la sesión y volver a entrar. Esto activa la membresía del grupo *docker* para la sesión actual.

```
>> newgrp docker
```

- Inicia el servicio Docker, necesario para ejecutar y gestionar contenedores Docker.

```
>> sudo service docker start
```

Es importante usar estas configuraciones con precaución, especialmente los cambios en el archivo *sudoers*, ya que pueden afectar la seguridad del sistema.

Una vez configurado el entorno Ubuntu se puede empezar a construir y correr las imágenes de Docker para montar el entorno, en este caso concreto, se han clonado los repositorios en la carpeta TFG TELECO, por lo que para compilar las imágenes Docker se debe abrir una terminal en dicho directorio y ejecutar los siguientes comandos, los cuales ejecutarán los siguientes Dockerfile:

14. Para las imágenes de Mongo y Eclipse-Mosquitto, se utilizarán las imágenes oficiales de Docker por lo que hay que descargarlas:

```
>> docker pull mongo
>> docker pull eclipse-mosquitto
```

15. Para compilar la imagen backend-demo se ejecutará el siguiente comando para ejecutar el Dockerfile (Fragmento de código 5), el cual creará una imagen basada en Python (versión 3.8), instalará los requisitos detallados en el fichero **requirements.txt** y ejecutará el script **testManager.py**.

```
>> docker build -t backend-demo ./backend_src/dummy/
```

```
1 FROM python:3.8
2
3 WORKDIR /backend
4
5 COPY . /backend
6
7 RUN pip install --no-cache-dir -r requirements.txt
8
9 CMD ["python", "./testManager.py"]
10
```

Fragmento de código 5: Dockerfile de la imagen de Docker "backend-demo". (Fuente: Elaboración propia)

16. Para compilar la imagen node-simulator se ejecutará el siguiente comando para ejecutar el Dockerfile (Fragmento de código 6), el cual creará una imagen basada en Python (versión 3.8), instalará los requisitos detallados en el fichero **requirements.txt** y ejecutará el script **node-simulator.py**.

```
>> docker build -t node-simulator ./backend_src/simulator/
```

```
1 FROM python:3.8
2
3 WORKDIR /backend
4
5 COPY . /backend
6
7 RUN pip install --no-cache-dir -r requirements.txt
8
9 CMD ["python", "./node-simulator.py"]
10
```

Fragmento de código 6: Dockerfile de la imagen de Docker "node-simulator". (Fuente: Elaboración propia)

17. Para compilar la imagen controller-sim se ejecutará el siguiente comando para ejecutar el Dockerfile (Fragmento de código 7), el cual creará una imagen basada en Python (versión 3.8), instalará los requisitos detallados en el fichero *requirements.txt* y ejecutará el script *controllerSimulator.py*.

```
>> docker build -t controller-sim ./backend_src/controller/
```

```
1 FROM python:3.8
2
3 WORKDIR /backend
4
5 COPY . /backend
6
7 RUN pip install --no-cache-dir -r requirements.txt
8
9 CMD ["python", "./controllerSimulator.py"]
10
```

Fragmento de código 7: Dockerfile de la imagen de Docker “controller-sim”. (Fuente: Elaboración propia)

Una vez se tienen construidas todas las imágenes Docker necesarias, se puede lanzar dichas imágenes, para lo que se tienen tres opciones:

- Lanzar una a una las imágenes con los siguientes comandos:

```
>> docker run --rm --network host --name mongo-database -d mongo:latest
>> docker run --rm -p 127.0.0.1:1883:1883/tcp -p 9001:9001 --name broker -d eclipse-mosquitto
>> docker run --rm --network host --name sdn-controller -d controller-sim
>> docker run --network host -d backend-demo
>> docker run --network host -d node-simulator
```

- Lanzar un script creado para construir y correr todas las imágenes necesarias (Fragmento de código 8). Este fichero está en el repositorio con el nombre “*deploy.sh*”.

```
1 docker pull mongo
2 docker pull eclipse-mosquitto
3 docker build -t node-sim simulator/.
4 docker build -t controller-sim controller/.
5 docker build -t backend-demo dummy/.
6
7 docker run --rm --network host --name mongo-database -d mongo:latest
8 docker run --rm -p 127.0.0.1:1883:1883/tcp -p 9001:9001 --name broker
  -d eclipse-mosquitto
9 docker run --rm --network host --name sdn-controller -d controller-sim
10 docker run --rm --network host --name node -d node-sim
11 docker run --rm --network host --name backend backend-demo
```

Fragmento de código 8: Script para correr el backend de la aplicación. (Fuente: Elaboración propia)

- Lanzar el Docker Compose, para lo que se puede ejecutar el *docker-compose.yml* que se puede encontrar en el repositorio *frontend_src*, el cual ha sido desarrollado en este trabajo y se muestra a continuación. Para ejecutarlo se debe abrir una terminal en *fronten_src* y ejecutar los siguientes comandos:

```
>> docker-compose up -d
```

```
1  version: "3.7"
2  services:
3    backend:
4      image: backend-demo
5      container_name: backend-demo
6      network_mode: "host"
7    simulator:
8      image: node-simulator
9      container_name: node-simulator
10     hostname: node-simulator
11     network_mode: "host"
12    mongo:
13     image: mongo:latest
14     container_name: mongo
15     network_mode: "host"
```

Fragmento de código 9: Fichero docker-compose.yml para correr la aplicación. (Fuente: Elaboración propia)

Por último, una vez se tiene el backend corriendo, se puede ejecutar el código del frontend, para ello se deben seguir los siguientes pasos para preparar el entorno de desarrollo:

1. Instalar NVM:

```
>> wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
```
2. Instalar Node con NVM:

```
>> nvm install 12.22.12
```
3. Instalar Vue Cli:

```
>> npm install -g @vue/cli
>> npm install --save-dev @vue/cli-service
```
4. Iniciar el servidor, para lo que hay que abrir una terminal en el repositorio *frontend_src* y ejecutar:

```
>> npm run serve
```

Otra opción diferente a todos estos pasos para ejecutar el frontend sería construir una imagen Docker y correrla sobre el puerto 8080. Para compilar la imagen del front-end se ejecutará el siguiente comando para ejecutar el Dockerfile (Fragmento de código 10), el cual creará una imagen basada en node (versión 11.15-stretch), instalará los requisitos detallados en el fichero *package.json* y ejecutará la aplicación.

```
>> docker build -t qdsn_front .
>> docker run --rm -it -p 8080:8080 --name qdsn_front qdsn_front
```

```

1 # Base image
2 FROM node:11.15-stretch
3
4 # Make app as actual work directory
5 WORKDIR /app
6
7 # Copy 'package.json' y 'package-lock.json' (if available)
8 COPY package*.json ./
9
10 # Install dependencies
11 RUN npm install
12
13 # Copy files and folders of the project to workdir
14 COPY . .
15
16 # start app
17 CMD ["npm", "run", "serve"]

```

Fragmento de código 10: Dockerfile del front-end de la aplicación. (Fuente: Elaboración propia)

Finalmente, después de seguir todos estos pasos, se podrá acceder a la aplicación desde cualquier navegador yendo a <http://localhost:8080>.

4.4. Arquitectura de la aplicación.

La arquitectura de la aplicación (ver Figura 27) se organiza en tres secciones principales: Frontend, Backend y Gateway Services, cada una con componentes y tecnologías específicas que contribuyen a la funcionalidad general de la aplicación.

- El Frontend está desarrollado en Vue.js, un framework JavaScript que facilita la creación de interfaces de usuario dinámicas y reactivas. Dentro de esta sección, se destacan componentes como Web y Pages, que representan la interfaz de usuario accesible desde navegadores web. El corazón del frontend es el Frontend SC, que actúa como controlador, gestionando la lógica de presentación y la interacción del usuario con la aplicación. Esta capa se comunica con el backend para recuperar datos y realizar operaciones necesarias, asegurando una experiencia de usuario fluida y eficiente.
- El Backend de la aplicación está implementado en Python, un lenguaje de programación conocido por su simplicidad y poder, especialmente en el desarrollo de aplicaciones web y procesamiento de datos. El componente central es el Backend SC, que maneja la lógica de negocio y sirve como punto de integración de diversos servicios. Este backend interactúa con los usuarios y otros sistemas mediante REST API para solicitudes basadas en HTTP y socket.io para comunicaciones en tiempo real. Además, el sistema utiliza una base de datos MongoDB, que almacena los datos de la aplicación de manera estructurada y accesible. Para manejar el flujo de datos en tiempo real y la comunicación entre diferentes servicios, se implementan tecnologías adicionales como Kafka para la gestión de flujos de datos y MQTT para la comunicación en redes IoT. También se utiliza OPC UA, un protocolo de comunicación estándar en automatización industrial, asegurando la interoperabilidad entre diversos sistemas y dispositivos.

- Los Gateway Services soportan la conectividad y comunicación de la aplicación con redes y dispositivos externos. Incluyen un SDN Controller que gestiona la configuración dinámica de la red, permitiendo la optimización del tráfico de datos. Avahi mDNS proporciona descubrimiento de red, facilitando que los dispositivos se encuentren y comuniquen entre sí en una red local. La interfaz 6LoWPAN es utilizada para la comunicación en redes de baja potencia, típicas de dispositivos IoT, mientras que el WiFi AP actúa como un punto de acceso, proporcionando conectividad a la red.
- La aplicación está diseñada para ejecutarse en contenedores Docker, lo que facilita la gestión y el despliegue en diversos entornos. Docker permite empaquetar la aplicación junto con todas sus dependencias, asegurando que funcione de manera consistente en cualquier lugar.

En conjunto, esta arquitectura modular y bien estructurada permite una integración eficiente entre el frontend y el backend, soportada por servicios robustos que garantizan la conectividad y la comunicación continua, todo mientras se mantiene la escalabilidad y facilidad de mantenimiento gracias a la utilización de Docker y tecnologías modernas.

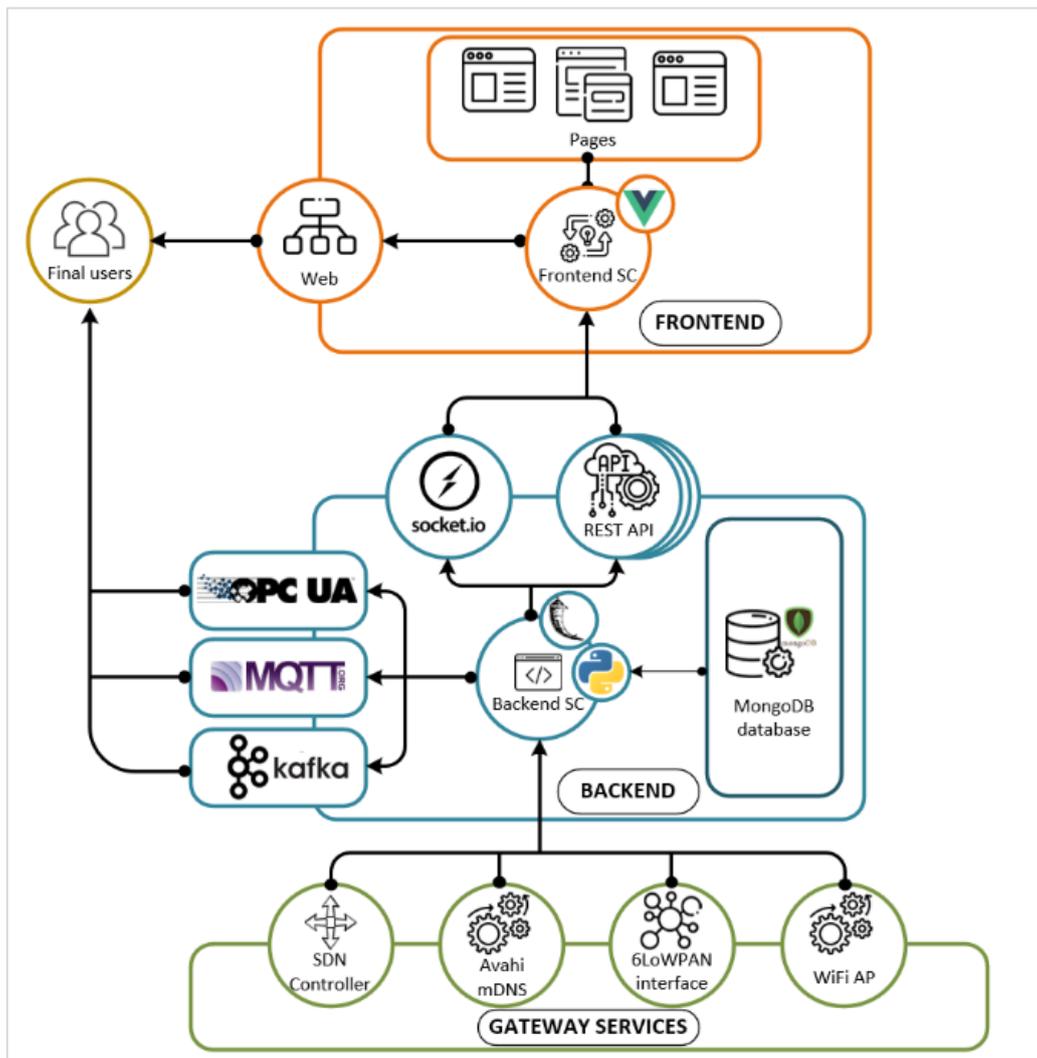


Figura 27: Diagrama de la arquitectura de la aplicación. (Fuente: ITI)

5. Requisitos de la plataforma a desarrollar para el despliegue y monitorización de redes de sensores inalámbricas.

Mediante el sistema de despliegue QDSN, un operario, sin necesidad de tener experiencia en comunicaciones, puede ser capaz de desplegar una red de sensores inalámbricos siguiendo un proceso iterativo. El operario se ayuda de un interfaz HMI⁵⁶, mediante LEDs en este caso, en el que le irá indicando la mejor posición para colocar los dispositivos. En la Figura 28 se puede ver una descripción del proceso.

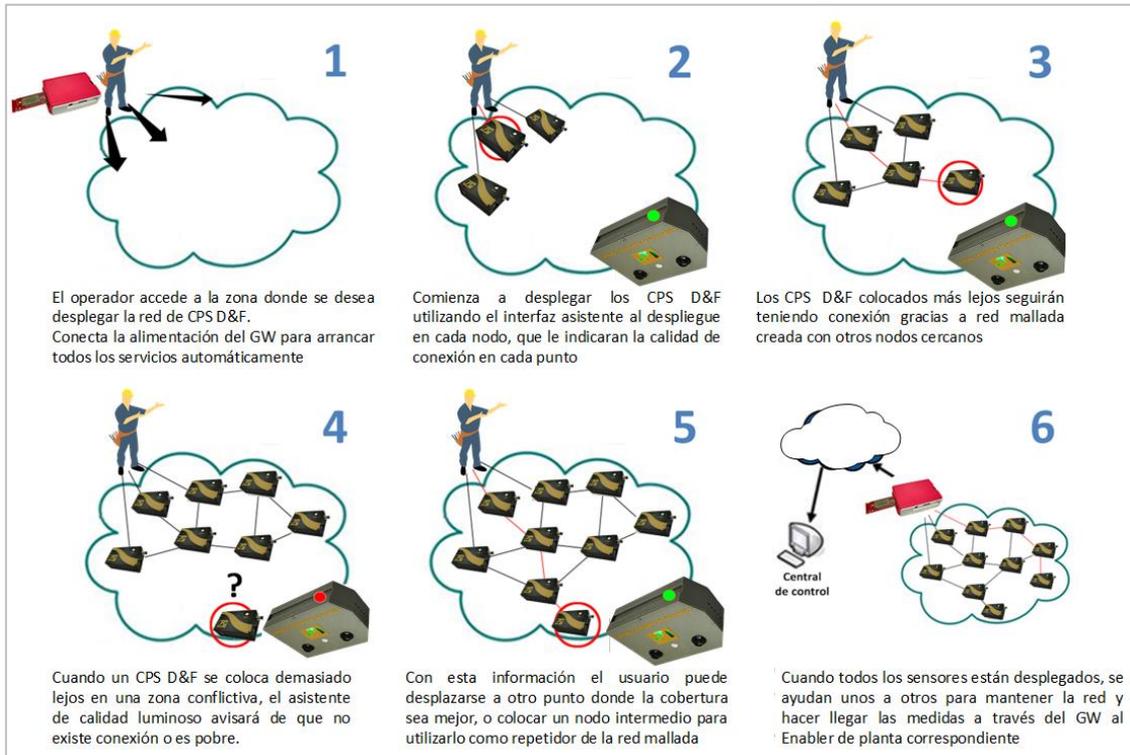


Figura 28: Flujo de trabajo para el despliegue de una WSN. (Fuente: ITI)

El objetivo del interfaz web para la plataforma QDSN consiste en disponer de un interfaz, tanto de configuración como de visualización de variables en tiempo real. El sistema está formado por una pasarela de comunicaciones, encargada de transferir la información de la WSN hacia una red ethernet, y un conjunto de nodos sensores encargados de la generación de la información. En la Figura 29 se puede observar el diseño conceptual del sistema QDSN.

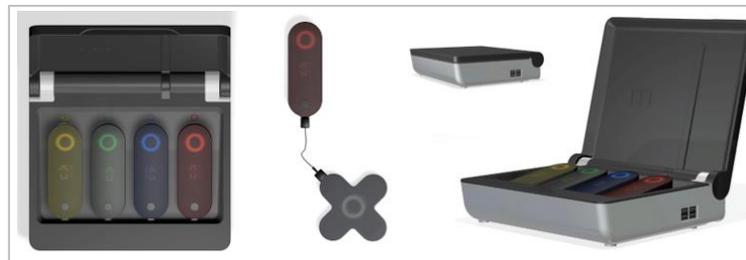


Figura 29: Nodos de sensores del sistema QDSN. (Fuente: ITI)

⁵⁶ Una interfaz HMI (Human-Machine Interface) es una plataforma que permite la interacción entre los humanos y las máquinas o sistemas. Esta interfaz facilita el control, monitoreo y gestión de procesos y dispositivos mediante una representación visual y/o táctil.

La web será el interfaz de configuración de la pasarela, como si de un router convencional se tratara, pero con funcionalidades específicas para el control, configuración y monitorización de la red en particular que se ha desplegado.

En base a todo el análisis expuesto en apartados anteriores, en los que se recoge tanto los sistemas desarrollados en proyectos previos como un estudio del estado del arte actual, se van a proponer una serie de requisitos de funcionamiento que deben de estar presentes, en mayor o menor medida, en los sistemas finalmente desarrollados. Estos requisitos estarán divididos en dos partes, por un lado, los requisitos necesarios para la parte del establecimiento de las comunicaciones, y por otro, los requisitos de la parte hardware.

5.1. Definición de requisitos de diseño de comunicaciones

El diseño de un sistema de comunicaciones eficaz y robusto es crucial para garantizar el rendimiento óptimo y la fiabilidad de las redes de sensores distribuidas. Es necesario considerar una serie de aspectos técnicos y operativos que aborden la resistencia a interferencias, la calidad del servicio, el ahorro energético, la interoperabilidad y el soporte al despliegue. Estos requisitos forman la base para desarrollar una infraestructura de comunicaciones que cumpla con los estándares más altos y satisfaga las demandas específicas del sistema. A continuación, se exponen los requisitos:

- Robustez frente a interferencias en la red:
 - Soporte multicanal y comunicaciones determinista para mitigar las interferencias con el resto de los nodos de la misma red.
 - Mitigar interferencias multi camino y de tecnologías en la misma banda mediante técnicas de salto en frecuencia.
 - Evitar la desconexión debido a las atenuaciones de la señal mediante enrutamiento dinámico.
- Calidad de servicio:
 - Los equipos deben de tener una serie de recursos radio base garantizados, obtenidos mediante una planificación sobre TSCH⁵⁷.
 - La tasa de transferencia de cada nodo debe estar acotada y analizada por el sistema, de tal forma que en función del estado del nodo (número de hijos, rol de transmisión, ...) se pueda dar un valor a una herramienta de planificación.
 - La latencia de los equipos deberá de estar acotada, en función de su ubicación lógica en la topología de red.
 - Los equipos deben determinar la calidad del enlace, además de ser capaces de alternar a un camino con mayor calidad si estuviera disponible.

⁵⁷ TSCH (Time-Slotted Channel Hopping) es un protocolo de comunicación utilizado en redes de sensores inalámbricos. Combina el acceso basado en ranuras de tiempo (TDMA) con el salto de canal (FHSS) para mejorar la fiabilidad y eficiencia de la red. Cada nodo se sincroniza en el tiempo y cambia de canal en intervalos regulares, lo que reduce las interferencias y aumenta la robustez de la comunicación. TSCH es una parte clave del estándar IEEE 802.15.4e y se utiliza en aplicaciones industriales y de IoT que requieren comunicaciones deterministas y de baja potencia.

- Los nodos WSN deberán de recoger diferentes tipos de métricas de calidad como: nivel de señal RSSI⁵⁸, identificador de la probabilidad de éxito de las comunicaciones (ETX⁵⁹), congestión de las colas de mensajes o el nivel de las baterías.
- Ahorro energético: Los equipos deben ser capaces de registrar el consumo de batería, además del nivel actual de las mismas.
- Interoperabilidad IP: Los nodos WSN tendrán la posibilidad de interoperar con otros equipos IPv6.
- Soporte al despliegue
 - Las medidas de calidad de cada nodo deben de proporcionarse a un sistema central que pueda realizar una evaluación de estas.
 - Los nodos inalámbricos dispondrán de un interfaz HMI en el que poder representar el nivel de calidad en la ubicación actual.
 - El sistema de representación de calidad deberá ser suficiente para conocer la mejor ubicación de un equipo, pudiendo utilizar la herramienta de manera optativa.
- Herramienta de planificación
 - El Gateway⁶⁰ tendrá disponible una herramienta que permita visualizar diferentes factores de rendimiento de la red de sensores.
 - La herramienta dispondrá de información sobre el rendimiento de los nodos de manera individual.
 - Mediante la herramienta será posible visualizar métricas como el nivel de batería, throughput individual y acumulado, latencia, recursos asignados, calidad del enlace.
 - Mediante la herramienta será posible realizar simulaciones que muestren como se modifica el rendimiento de la red al añadir nodos nuevos.

⁵⁸ RSSI (Received Signal Strength Indication) es una medida utilizada en redes de comunicación inalámbrica para indicar la intensidad de la señal recibida por un receptor desde un transmisor. Es un valor numérico que representa la potencia relativa de la señal radioeléctrica en decibelios (dBm) o en una escala relativa. El RSSI se utiliza para determinar la calidad de la conexión inalámbrica y puede influir en decisiones como el cambio de canal o la retransmisión de datos para optimizar el rendimiento y la fiabilidad de la comunicación.

⁵⁹ ETX (Expected Transmission Count) es una métrica utilizada en redes de comunicación para estimar la calidad de un enlace entre dos nodos. Representa el número esperado de transmisiones necesarias para que un paquete de datos se entregue correctamente al destino, considerando pérdidas de paquetes y retransmisiones. Un valor bajo de ETX indica un enlace de alta calidad y confiabilidad, mientras que un valor alto puede indicar problemas de interferencia, atenuación de señal o congestión en la red.

⁶⁰ Un Gateway es un dispositivo de red que actúa como un punto de entrada y salida entre redes diferentes, facilitando la comunicación y el intercambio de datos entre ellas. Suele realizar funciones de traducción de protocolos, enrutamiento de tráfico y gestión de la conectividad, permitiendo que dispositivos de diferentes tipos y tecnologías puedan comunicarse entre sí de manera efectiva. En contextos específicos, como en redes de sensores o IoT, un Gateway puede ser el punto central para la recolección, procesamiento y envío de datos hacia sistemas externos o la nube.

Para cumplir con algunos de los requisitos anteriormente expuestos, se ha propuesto la siguiente selección de protocolos y tecnologías para llevar a cabo la implementación del sistema de comunicaciones.

A nivel físico y MAC, los dispositivos utilizarán el estándar IEEE 802.15.4 por lo que podrán trabajar en la banda de 2,4GHz empleando hasta 16 canales de los disponibles. Por otra parte, para el mecanismo de acceso al medio, el método TSCH incluye diferentes técnicas que lo hacen más robusto frente a interferencias y desvanecimientos de la señal, lo que lo hace un mecanismo idóneo para trabajar en escenarios industriales.

Para la parte de integración con redes IPv6, se ha incluido el protocolo de adaptación 6LoWPAN, que permite comprimir los datagramas IPv6 para que puedan ser utilizados en el estándar IEEE 802.15.4, cuyo tamaño de paquete es muy inferior a lo permitido en IPv6.

Para la parte de encaminamiento, el protocolo RPL permite crear una topología en árbol de manera sencilla y eficiente, lo que permite que equipos con limitaciones de consumo energético puedan construir sus tablas de encaminamiento sin necesidad de utilizar un excesivo intercambio de mensajes ni de consumo energético.

Finalmente, los mensajes de datos que transmitan los nodos WSN utilizarán el protocolo UDP, ya que, al no requerir el establecimiento de una sesión, requiere menor cantidad de tráfico de señalización para el envío de información.

La pasarela de comunicaciones deberá de disponer de dos interfaces de comunicaciones, por un lado, el interfaz que le permita conectar con los nodos que utilicen IEEE 802.15.4, y por otro, un interfaz de red que le permita tener acceso a internet o a una red local mediante Ethernet, o de manera adicional Wi-Fi.

Además, la pasarela de comunicaciones debe ser capaz de interactuar con servicios de terceros a través de los protocolos de comunicación más extendidos en la industria, como son MQTT y OPC-UA. Sería interesante que esta pasarela también fuese capaz de interactuar con algunas de las plataformas IoT más extendidas como Azure o AWS. Por último, otro de los servicios que deberán de estar disponibles en la pasarela es un servidor web mediante el que podamos acceder a un interfaz de configuración del sistema, herramientas de planificación y de visualización.

En la Tabla 10 se puede observar un resumen de los requisitos de comunicaciones para el sistema.

| Requisitos de diseño de comunicaciones | |
|--|-----------|
| Requisito | Prioridad |
| Robustez frente a interferencias en la red | Alta |
| Calidad de servicio | Alta |
| Ahorro energético | Alta |
| Interoperabilidad IP | Alta |
| SopORTE al despliegue | Alta |
| Herramienta de planificación | Alta |

Tabla 10: Requisitos de diseño de comunicaciones. (Fuente: ITI)

Finalmente, la herramienta de planificación que se propone deberá de ejecutarse en la pasarela, y el usuario podrá acceder a ella para obtener estadísticas de rendimiento del sistema, así como configurar los dispositivos o planificar/simular un nuevo despliegue. En la Figura 30 se muestra el diseño conceptual de la herramienta.

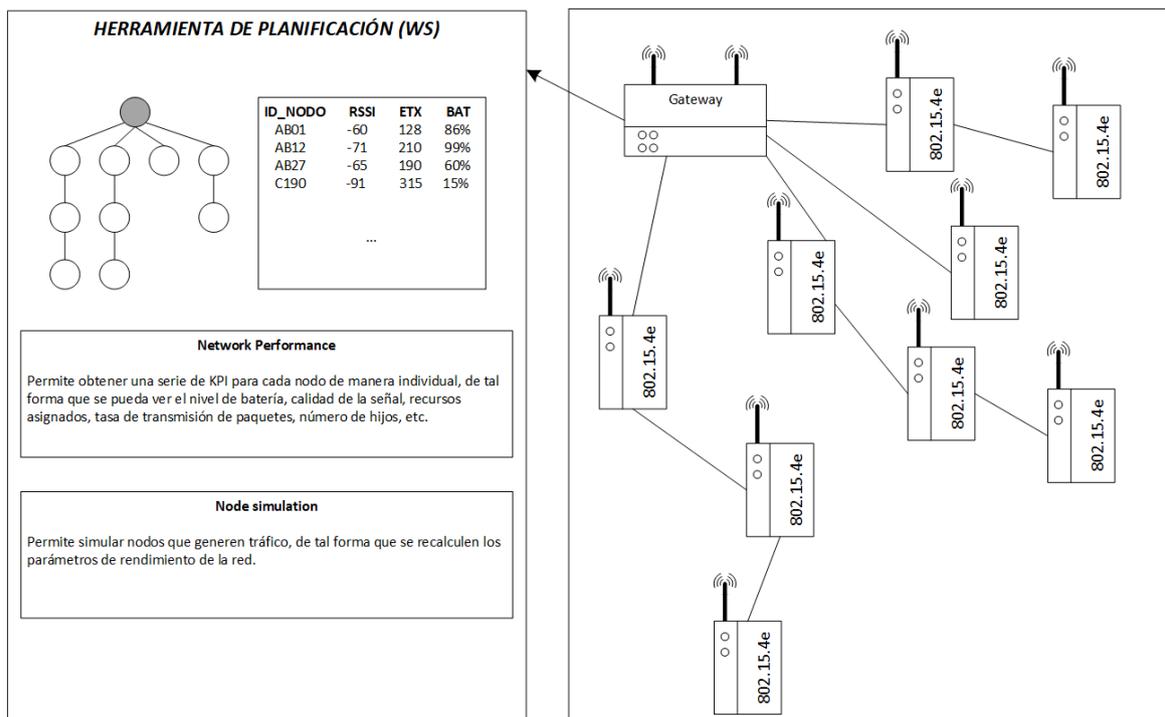


Figura 30: Diseño conceptual de la herramienta de planificación. (Fuente: ITI)

5.2. Requisitos generales de la interfaz de usuario a desarrollar

En esta sección se detallan los requisitos esenciales que debe cumplir la interfaz de usuario de la plataforma web a desarrollar. Estos requisitos han sido diseñados para asegurar que la plataforma no solo sea funcional y eficiente, sino también accesible y escalable. A continuación, se describen los aspectos clave que se considerarán en el desarrollo de la interfaz de usuario:

- **Frontend desplegado sobre RaspberryPi⁶¹ o similar:** Inicialmente, la plataforma web estará desplegada en un equipo hardware con distribución Linux, como Raspberry Pi u otro dispositivo similar. Este dispositivo actuará como la pasarela principal de comunicaciones para la aplicación web. Se ha elegido esta configuración inicialmente debido a su bajo costo y su versatilidad para ejecutar aplicaciones web en un entorno controlado. Esta flexibilidad permitirá adaptar la infraestructura de despliegue según las necesidades, así como para garantizar la escalabilidad y disponibilidad futuras de la aplicación.
- **Idioma de la plataforma:** La plataforma web se diseñará principalmente en inglés para alcanzar una mayor accesibilidad y alcance global. No obstante, se considerará como requisito secundario la capacidad de alternar entre los idiomas inglés y español. Esta funcionalidad adicional permitirá adaptar la plataforma a las preferencias de los usuarios, garantizando una experiencia más personalizada y satisfactoria. La implementación de esta alternancia de idiomas se realizará de manera que los usuarios puedan seleccionar su idioma preferido desde la interfaz de usuario, asegurando así una navegación intuitiva y amigable independientemente del idioma elegido.

⁶¹ Raspberry Pi es una serie de computadoras de placa única (SBC, por sus siglas en inglés) desarrollada por la Raspberry Pi Foundation. Estas pequeñas computadoras están diseñadas para ser económicas, versátiles y de bajo consumo de energía. Se utilizan en una amplia variedad de proyectos educativos, de bricolaje y comerciales.

- **El sistema está pensado para que accedan entre 1-5 usuarios de forma simultánea:** Esta limitación ha sido establecida considerando las necesidades, garantizando un rendimiento óptimo y una experiencia fluida para los usuarios dentro de este rango. Esta capacidad proporciona un equilibrio entre eficiencia y escalabilidad, permitiendo una gestión eficaz de los recursos del sistema.
- **Arquitectura basada en API REST⁶²:** En esta sección se contempla la implementación de un API REST que permitirá la consulta y configuración de diversos campos del equipo a través de peticiones HTTP. Este API REST será diseñado siguiendo los principios de REST para garantizar la interoperabilidad, la escalabilidad y la simplicidad en la comunicación entre el frontend y el backend de la aplicación. Además, se considera altamente beneficioso el enfoque de tener el servidor web de configuración dockerizado, lo que proporcionará un entorno de desarrollo y despliegue más consistente y portátil. La utilización de contenedores Docker facilitará la gestión de la infraestructura y asegurará la consistencia del entorno de ejecución, permitiendo una configuración y despliegue más ágil y controlado de la aplicación web.
- **Plataforma web responsiva:** Se valorará altamente la capacidad de la plataforma web para adaptarse de manera óptima a diferentes dispositivos, incluyendo exploradores de escritorio, smartphones y tabletas. Dado que es común que el personal utilice dispositivos móviles durante las fases de despliegue de la plataforma, se considera importante que la aplicación pueda ser accesible y funcional en estas circunstancias. No obstante, es crucial destacar que el principal foco de atención recae en garantizar que la versión de escritorio de la plataforma sea atractiva y funcional. La experiencia del usuario en la plataforma web de escritorio debe ser prioritaria, asegurando un diseño intuitivo y atractivo que facilite la interacción y maximice la eficiencia del usuario durante su uso.
- **Selección de lenguaje o framework:** Dado que gran parte de las funcionalidades de la plataforma web requieren una consulta constante de información de la pasarela, así como la visualización en tiempo real de datos procedentes de los nodos inalámbricos, es fundamental seleccionar un lenguaje de programación o framework que facilite esta interacción y representación de datos. Se dará prioridad a aquellos lenguajes o frameworks que cuenten con bibliotecas o herramientas que simplifiquen la comunicación con la pasarela y permitan la integración de flujos de datos en tiempo real de manera eficiente y segura. Angular, React y Vue.js son opciones sólidas que ofrecen capacidades para manejar flujos de datos en tiempo real y realizar consultas constantes a la pasarela. Cada uno de estos frameworks tiene sus propias fortalezas y características únicas, por lo que la elección final dependerá de la experiencia y preferencias del equipo de desarrollo, esta decisión se explicará en el punto dedicado a ello.
- **Comunicación “inter-process” (IPC) entre pasarela y frontend:** Como parte del desarrollo de la plataforma web, es fundamental definir la tecnología que facilitará el intercambio de información en tiempo real entre la pasarela y el frontend. Se considerarán diversas opciones tecnológicas, como consultas API REST, pipes, sockets, AMQP (Advanced Message Queuing Protocol), Kafka, MQTT (Message Queuing Telemetry Transport), entre otras. La elección de la tecnología adecuada dependerá de los requisitos

⁶² API REST, que significa Interfaz de Programación de Aplicaciones Representacional del Estado Transferido, es un estilo arquitectónico para el diseño de servicios web que se basa en el protocolo HTTP. Permite a las aplicaciones comunicarse entre sí de manera eficiente y escalable, utilizando URIs para identificar recursos y métodos HTTP (GET, POST, PUT, DELETE) para operar sobre ellos.

específicos del proyecto, incluyendo la velocidad, la fiabilidad, la escalabilidad y la complejidad de la comunicación requerida. Se evaluarán cuidadosamente las características y capacidades de cada opción para garantizar una comunicación eficiente y segura entre la pasarela y el frontend, asegurando así un funcionamiento óptimo de la plataforma web en tiempo real.

Estos requisitos generales establecerán las bases para el desarrollo de una plataforma web robusta, versátil y preparada para futuras expansiones, garantizando una experiencia de usuario satisfactoria y un funcionamiento óptimo.

5.4. Diseño de la interfaz de usuario a desarrollar

En esta sección se detallan las distintas ventanas de visualización y configuración que formarán parte de la plataforma web. Estas ventanas permitirán a los usuarios interactuar con la aplicación, visualizar datos en tiempo real, realizar configuraciones y ajustes según sea necesario. La disposición y funcionalidades de estas ventanas se diseñarán teniendo en cuenta los requisitos específicos del proyecto y las necesidades de los usuarios finales. La interfaz web debe incluir cuatro funcionalidades principales:

1. **Despliegue de nodos:** La funcionalidad de Despliegue de nodos se presenta como un componente fundamental en la configuración inicial de dispositivos y el soporte al despliegue de la red de sensores. Se ha diseñado como un proceso guiado, presentado en forma de un breve Wizard con pasos concisos, con el objetivo de facilitar al operario el proceso de instalación y configuración de los sensores de manera eficiente y sin complicaciones.

La herramienta ofrece un proceso paso a paso, desde la configuración inicial hasta la activación de los dispositivos. Se proporcionan opciones para la configuración específica de cada dispositivo, facilitando la personalización de acuerdo con las necesidades del entorno de despliegue.

Durante el proceso, se realizan validaciones en tiempo real para garantizar la integridad de los datos ingresados. Asimismo, se ofrece asistencia contextual y ayuda en línea para aclarar cualquier duda o proporcionar información adicional sobre las opciones de configuración disponibles. Además, se facilitará una herramienta de simulación que advierta al usuario de configuraciones problemáticas o que no sean posibles.

El objetivo es simplificar y agilizar el proceso de instalación y configuración inicial de los dispositivos, garantizando una experiencia fluida y sin contratiempos para el operario.

2. **Estadísticas de red:** La funcionalidad de Estadísticas de red ofrece una visión general de la red de dispositivos desplegados, proporcionando estadísticas generales, una representación de la topología y un listado de nodos con información adicional.

Esta herramienta permite a los usuarios comprender el estado y la salud de la red en su conjunto. Proporciona métricas clave, como la disponibilidad de los nodos, el estado de la conexión y el rendimiento general de la red.

La representación de la topología ofrece una visualización gráfica de la estructura de la red, mostrando las relaciones entre los nodos y resaltando cualquier anomalía o congestión que pueda surgir.

Además, se proporciona un listado detallado de todos los nodos desplegados, junto con información adicional relevante, como el estado de la batería, la intensidad de la señal y cualquier alerta o evento reciente asociado con cada nodo.

3. **Datos en tiempo real:** La funcionalidad de Datos en tiempo real permite a los usuarios visualizar la información transmitida por todos los dispositivos de la red en tiempo real. Los usuarios pueden seleccionar diferentes sensores de cada dispositivo y obtener la representación en directo de los mensajes recibidos.

Esta herramienta ofrece una vista dinámica y actualizada de los datos, lo que facilita la detección temprana de anomalías y la toma de decisiones informadas. Los usuarios pueden monitorear de manera efectiva los cambios y eventos en la red a medida que ocurren, lo que les permite reaccionar rápidamente ante cualquier situación inesperada.

La capacidad de seleccionar diferentes sensores y personalizar la visualización de los datos proporciona una experiencia de usuario flexible y adaptable. Los usuarios pueden centrarse en los aspectos específicos de la información que son más relevantes para sus necesidades, lo que les permite obtener información útil y relevante de manera eficiente.

4. **Configuración de la pasarela:** La funcionalidad de Configuración de la pasarela permite a los usuarios modificar diferentes aspectos de gestión de la pasarela de comunicación. Por ejemplo, pueden cambiar direcciones IP, configurar el cliente MQTT y cambiar el punto de acceso Wi-Fi.

Esta herramienta proporciona a los usuarios el control completo sobre la configuración y el funcionamiento de la pasarela, lo que les permite adaptarla según sus necesidades específicas y requisitos del entorno.

La capacidad de cambiar la dirección IP facilita la integración de la pasarela en diferentes redes y entornos de trabajo. La configuración del cliente MQTT permite a los usuarios establecer parámetros de conexión para garantizar una comunicación segura y eficiente con otros dispositivos.

Además, el cambio del punto de acceso Wi-Fi ofrece flexibilidad en cuanto a la conectividad de la pasarela, permitiendo a los usuarios ajustar la configuración según los requisitos de la red local.

Cada una de estas funcionalidades tendrá su propia ventana específica dentro de la interfaz de usuario, diseñada para permitir la configuración o visualización de una información determinada. Para mejorar la usabilidad y la experiencia del usuario, se ha concebido la disposición de estas ventanas en una barra lateral, siguiendo el orden establecido anteriormente. Esto permitirá a los usuarios acceder fácilmente a las funcionalidades según el flujo de trabajo deseado.

La disposición en una barra lateral proporcionará una estructura clara y coherente, lo que facilitará la navegación y la interacción con la plataforma. Los usuarios podrán alternar entre las diferentes ventanas de manera rápida y sencilla, siguiendo el flujo lógico de trabajo establecido para cada una de las funcionalidades.

5.4.1. Flujo de trabajo de un operario desde que arranca el sistema

Cuando los nodos se encienden, entran en un estado de reposo o modo despliegue, sin enviar información de ningún sensor, pero facilitando la conexión de nodos nuevos. Al iniciar la plataforma, el operario accede al menú "Despliegue de nodos/Node Deployment", donde se encuentra con un proceso guiado que le permite configurar los nodos de manera sencilla y eficiente. En esta sección, se proporcionan instrucciones detalladas sobre cómo seleccionar la ubicación adecuada para cada nodo, cómo conectar los sensores correspondientes y cómo configurar qué sensores enviarán datos desde ese nodo específico.

Además, se ofrecen opciones para configurar los parámetros de los sensores, como intervalos de muestreo y umbrales de activación. Una vez que se completa la configuración de los nodos, el operario tiene la opción de ejecutar una simulación para verificar el comportamiento de los nodos antes de cargar la configuración en los nodos finales, es decir, la aplicación debe incluir una función de "Simulación" que permita al operario visualizar cómo la configuración almacenada en la pasarela afecta a los nodos, en comparación con la configuración actual cargada en los nodos. Esto proporcionará una forma segura de probar cambios en la configuración antes de implementarlos en los nodos finales. El operario podría decidir si cargar la configuración directamente a los nodos o dejar la configuración en la pasarela después de evaluar la comparación de la simulación.

Una vez que se completa la configuración de los nodos, el operario puede acceder al menú "Network/Network Statistics", que proporciona una visión general del funcionamiento de la red. Aquí, se muestran estadísticas importantes, como la disponibilidad de los nodos, el estado de la conexión y el rendimiento general de la red. Esta información permite al operario monitorear el estado de la red y tomar medidas proactivas para garantizar su buen funcionamiento.

Por otro lado, el operario puede monitorear en tiempo real la información recibida de los diferentes sensores en la ventana "Livestream". Esta función proporciona una vista dinámica y actualizada de los datos, lo que permite al operario detectar y responder rápidamente a cualquier anomalía o evento inesperado en la red.

Finalmente, el menú "Configuración" permite al operario ajustar parámetros de red y otros servicios de la pasarela de comunicaciones según sea necesario. Aquí, el operario puede realizar cambios en la configuración de la pasarela, como la configuración del cliente MQTT y el cambio del punto de acceso Wi-Fi, para garantizar un funcionamiento óptimo de la red en todo momento.

A continuación, se muestran una serie de bocetos que ejemplifican cada una de las funcionalidades descritas.

5.4.1. Inicio de sesión y perfil de usuario

El acceso a la plataforma web debe estar protegido mediante un sistema de autenticación que requiere un nombre de usuario y una contraseña. Por defecto, la plataforma debe contar con un usuario administrador, cuya contraseña inicial sea modificable desde la misma plataforma web una vez iniciada la sesión.

A continuación, se muestra un boceto preliminar que proporciona una visión general de la disposición de los elementos en la pantalla de inicio y perfil de usuario. Es importante tener en cuenta que este diseño es solo un prototipo inicial y podría cambiar en el diseño final. Se podría considerar la inclusión de un logo del proyecto o del sistema que pueda complementar la estética de la pantalla de inicio y perfil de usuario.

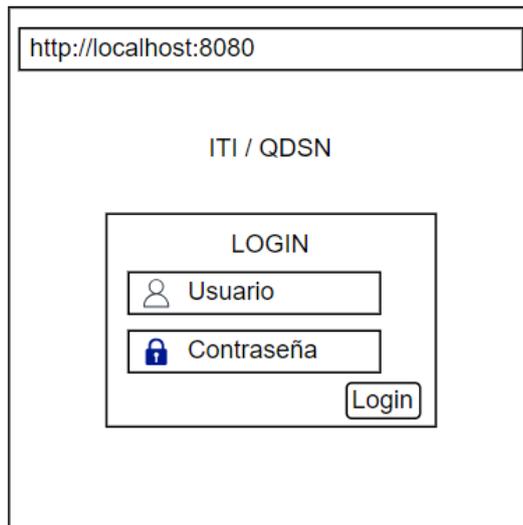


Figura 31: Boceto de la página de inicio de sesión. (Fuente: Elaboración propia)

Una vez dentro de la interfaz, los usuarios deben tener la opción de hacer clic en un botón dedicado para acceder a la gestión de su perfil. Desde esta sección, podrían realizar acciones como cerrar sesión o modificar la información de su cuenta, incluida la contraseña, tal y como se muestra en la Figura 32.

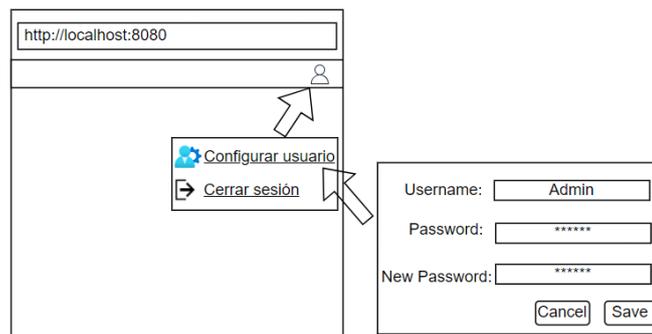


Figura 32: Boceto del botón para información del perfil. (Fuente: Elaboración propia)

5.4.2. Menú o navegador entre diferentes ventanas de visualización

La navegación entre las diferentes ventanas de visualización en la plataforma estará controlada por un menú vertical que proporcionará una navegación clara y organizada a través de las diferentes secciones de la plataforma. Cada opción está diseñada para ser fácilmente accesible y reconocible para los usuarios.

Además, en la parte superior de la pantalla, se encontrará un menú horizontal o menú de herramientas que albergará cinco botones, tal como se muestra en la Figura 33.

- El primer botón, representado con una casa, permitirá a los usuarios navegar rápidamente hasta la ventana de inicio o "home".
- El segundo botón, representado con tres rayas horizontales, desplegará un menú vertical que facilitará la navegación a través de las diferentes ventanas de la aplicación.
- El tercer botón, representado por una flecha hacia abajo, ofrecerá la funcionalidad de exportar datos, como se explicará más adelante en este documento.

- El cuarto botón, identificado con la letra "i" dentro de un círculo, desplegará una ventana con información relevante sobre el estado de la red.
- Por último, el botón representado por un ícono de usuario permitirá configurar el perfil del usuario o cerrar sesión, como se explicó en el apartado anterior.

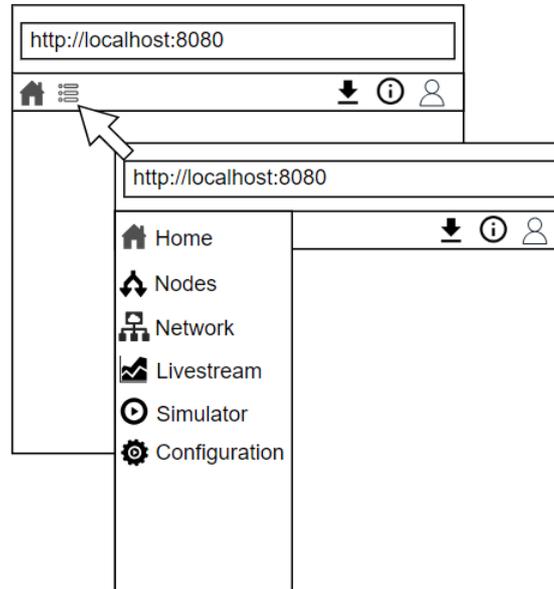


Figura 33: Boceto del menú principal de la aplicación. (Fuente: Elaboración propia)

Este diseño preliminar proporciona una visión general de la disposición de los elementos en el menú principal de la plataforma web, ofreciendo una base para futuras iteraciones y refinamientos del diseño final.

5.4.3. Página principal o página de inicio

La página principal de la aplicación o pantalla de inicio (Home) deberá ofrecer a los usuarios una visión panorámica del estado general de la red WSN una vez que han iniciado sesión. Este panel deberá centralizar información crucial sobre la conectividad de red y el estado de los dispositivos desplegados, permitiendo a los usuarios comprender rápidamente la salud y operatividad de su sistema en un solo vistazo.

En primer lugar, debe aparecer una sección dedicada al Gateway, lo que proporcionará esenciales sobre la conectividad de red del sistema. Aquí, los usuarios podrán verificar el estado de las interfaces Ethernet y WiFi, determinando si están conectadas o desconectadas. Además, se debe mostrar el nombre del SSID WiFi para facilitar el acceso local a la red.

Por otro lado, se requiere de una sección dedicada a la Red QDSN, la cual ofrecerá una visión detallada de los nodos desplegados en el sistema. Se presentará aquí el número de nodos configurados, indicando aquellos que han sido debidamente ajustados y están listos para operar. Del mismo modo, se mostrará el número de nodos sin configurar, lo que permite a los usuarios identificar rápidamente aquellos que aún requieren ajustes. Además, se proporcionará información sobre la cantidad total de sensores disponibles, lo que ayudará a los usuarios a comprender el alcance y la capacidad de su sistema en términos de recopilación de datos.

El diseño de esta página principal se enfoca en la claridad y la accesibilidad de la información. Se ha optado por una disposición intuitiva que permite una rápida comprensión del estado del sistema. El encabezado presenta el nombre de la aplicación y opciones de perfil para una navegación sencilla, como se ha visto en el apartado anterior. Las secciones de Gateway y Red

QDSN están claramente definidas, con indicadores visuales y números destacados para una fácil identificación, pero no necesariamente tienen que estar organizado de esta manera.

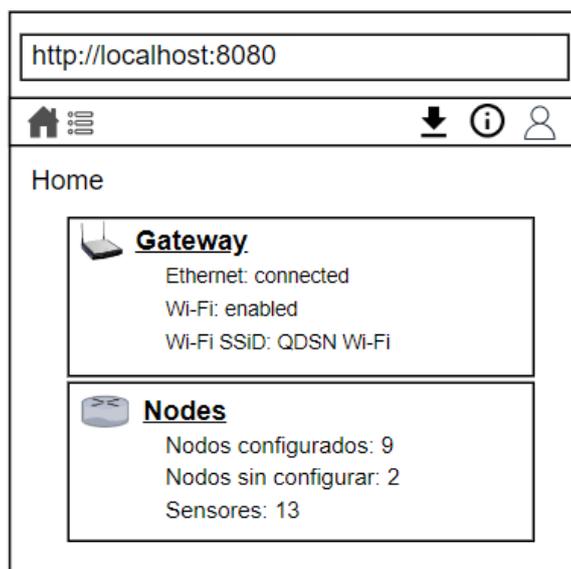


Figura 34: Boceto de la página de inicio. (Fuente: Elaboración propia)

5.4.4. Despliegue de nodos

La ventana de despliegue de nodos tiene como propósito principal la configuración de los sensores disponibles en cada nodo, así como facilitar el proceso de despliegue de estos nodos dentro de la red. Los nodos, al ser encendidos inicialmente, carecerán de cualquier sensor disponible para evitar el envío de información innecesaria en caso de no haber sensores conectados.

Al encender los nodos por primera vez, automáticamente se encuentran en un estado "Sin Configurar" o similar, indicando la necesidad de realizar una configuración básica antes de su implementación.

La ventana de despliegue debe presentar un listado de los nodos desplegados en la red, proporcionando la siguiente información:

- **Node ID:** Identificador único asignado a cada nodo para su identificación dentro del sistema.
- **Nombre:** Por defecto los nodos no tienen un nombre configurado, pero se podría ofrecer la posibilidad de añadir un nombre descriptivo al nodo, permitiendo a los usuarios identificar fácilmente su ubicación o propósito.
- **Dirección IPv6:** Dirección IP única asignada a cada nodo dentro de la red.
- **Estado:** Indica el estado actual del nodo, que puede ser:
 - Idle: Indica que el nodo está inactivo y no está desplegado ni generando datos.
 - Deployed: Señala que el nodo ha sido desplegado en la red, pero aún no está activo.
 - Active: Significa que el nodo está activo y generando datos.

Por otro lado, la ventana también debe ofrecer un botón que permite cambiar a otra ventana e iniciar un asistente o Wizard para guiar a los usuarios a través del proceso de instalación y configuración de los nodos. Este asistente facilitará el proceso al proporcionar pasos claros y secuenciados para la configuración adecuada de los sensores para el funcionamiento óptimo del nodo en la red.

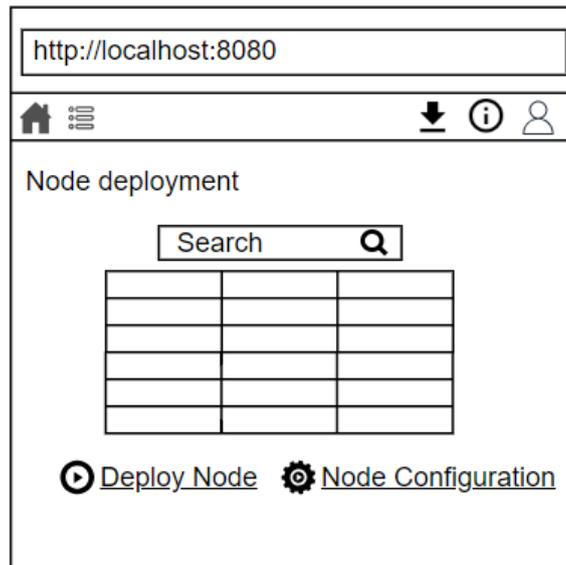


Figura 35: Boceto de la página de despliegue de nodos. (Fuente: Elaboración propia)

Además, fuera de la tabla de nodos, debe aparecer un botón o elemento similar que permitirá realizar una simulación comparando los valores configurados (sin cargar en los nodos) con la configuración actual de los nodos. Al hacer clic en este botón, los usuarios serán redirigidos a una ventana dedicada para visualizar los resultados y comparaciones de manera clara y concisa. Este simulador será explicado en este documento.

Por otro lado, deberá aparecer también un botón de configuración del nodo, el cual estará presente junto a cada nodo en la tabla y permitirá a los usuarios acceder a un asistente para desplegar y configurar el nodo de manera rápida y sencilla. El asistente se diseñará con énfasis en la facilidad de uso, con pasos claros y visuales para guiar al operario a través del proceso de configuración. Este asistente de despliegue será explicado en el siguiente punto de este documento.

5.4.5. Asistente de despliegue de nodos

El primer paso del asistente: a la hora de desplegar el nodo, estará conectado a la red, pero no transmitirá información de los sensores. Al conectarse a la red dispondrá de un nivel de calidad, que en el nodo vienen representados por diferentes colores (rojo, azul, verde).

Aquí se indica al operario que debe colocar el nodo en la ubicación deseada y comprobar la calidad de la conexión. En este punto se considera mostrar una imagen, o una animación que muestre el color del led parpadeando, que indique al operario que debe revisar la calidad del nodo. También se pueden obtener medidas en tiempo real de la calidad para que el asistente muestre en tiempo real la calidad que tiene el nodo.

Este asistente es único para cada nodo, por lo que se puede mostrar el ID único en algún punto de esta pantalla. Además, los nodos dispondrán de un sensor NFC, que al leerlo deberán de acceder directamente a esta pantalla de configuración.

Al pulsar en siguiente se accede al segundo paso del asistente. En este se indica al operario que debe conectar el HUB al nodo, y los sensores al HUB, este paso es únicamente descriptivo y no requiere de feedback por la plataforma. En este apartado habría que hacer una aclaración sobre como identificar los diferentes puertos de los sensores, para dar detalle de cómo hay que conectarlo dependiendo de la tecnología a utilizar. Por ejemplo, los puertos pueden ser de diferentes colores, dependiendo de si son para I2C, Entradas digitales, etc.

El tercer paso consiste en configurar los sensores de este nodo en concreto. Al llegar a este punto del asistente, si el nodo es nuevo, no debe mostrar ningún sensor, y se deben configurar uno por uno añadiéndolos a una lista o similar. Se considera tener un botón + en el que pulsar para añadir un nuevo sensor, que abra una ventana pop-up o similar para completar la información del sensor. Con forme se vayan añadiendo, se va completando la tabla de sensores. Cada vez que se añada un sensor nuevo, se enviará esta información a la pasarela para que pueda comunicar con el nodo en concreto y activar este sensor.

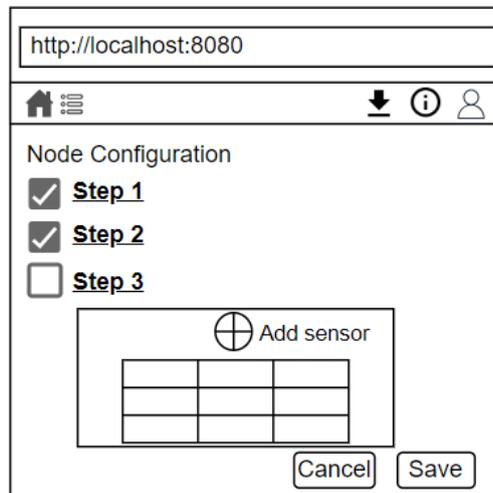


Figura 36: Boceto de la página de asistente de despliegue. (Fuente: Elaboración propia)

Una vez pulsado en OK o similar, el sensor se añade a la ventana anterior, pudiendo añadir un sensor nuevo. Cada vez que se termine con la configuración del sensor, se comunica a la pasarela. De esta forma se evita tener la información almacenada en la web. Se puede realizar una consulta API/Rest a la pasarela para que indique que sensores tiene configurados un sensor, de tal forma que, si se vuelve a acceder al asistente, se muestren los sensores ya configurados.

Si se quiere modificar un sensor ya añadido a la tabla, para modificar el periodo de muestreo, por ejemplo, se debe poder abrir la ventana de pop up para cambiar la configuración de un sensor que ya esté en la tabla.

Además, se añade la posibilidad de eliminar un sensor de la lista, avisando nuevamente a la pasarela para que deshabilite la lectura de ese sensor. Este intercambio de mensajes con la pasarela se realizaría mediante API/Rest.

5.4.6. Simulador

La funcionalidad del simulador permite visualizar la topología y las variables actuales de los nodos desplegados en la red. Esta ventana de simulación se accede desde el listado de nodos configurados y facilita la comparación entre la configuración realizada en la pasarela (configuración virtual) y la configuración real de los nodos.

El listado de parámetros a mostrar es:

- ID del nodo: Identificador único asignado a cada nodo.
- Nodo padre: Identificador del nodo padre en la topología de la red.
- Período de muestreo de sensores: Permite configurar sensores simulados con un determinado periodo de muestreo mediante una ventana emergente accesible a través de un botón.
- Nivel de batería: Indica el nivel actual de batería de cada nodo.
- Tiempo de vida del nodo: Indica el tiempo estimado que la batería del nodo durará hasta que se apague.

El simulador mostrará la diferencia entre los parámetros actuales de la red y los que se han modificado para la simulación. Esto se realiza identificando los parámetros reales configurados con color negro y los parámetros modificados para la simulación con color rojo. Se puede proporcionar una tabla que muestre los valores "Real / Simulado" para una fácil comparación (por ejemplo, Periodo: 800 / 1600).

Al hacer clic en el botón de simular, se obtienen los valores de tiempo de vida "simulados" mediante el intercambio de información con la pasarela a través del API/REST. La solicitud incluye la tabla de parámetros y la respuesta contiene los resultados de la simulación. Con esta respuesta se realiza la representación tanto de la topología como de los parámetros reales y simulados.

Al acceder a la ventana del simulador, se mostrará la red actual desplegada, proporcionando a los usuarios una herramienta poderosa para visualizar y comparar la configuración real y simulada de los nodos en la red. Además, se podrá observar el diagrama de topología de la red.

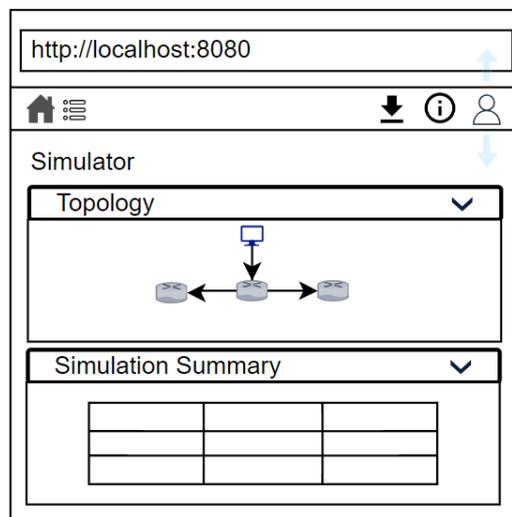


Figura 37: Boceto de la página del simulador. (Fuente: Elaboración propia)

5.4.7. Estadísticas de red

La funcionalidad de "Estadísticas de Red" permite visualizar información detallada sobre el funcionamiento de la red de sensores inalámbricos. Esta ventana se puede dividir en tres secciones principales: estadísticas de red, topología de la red y tabla con información detallada de cada nodo.

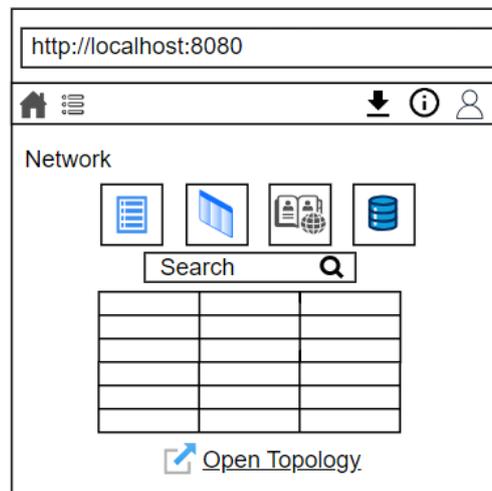


Figura 38: Boceto de la página de estadísticas de la red. (Fuente: Elaboración propia)

- Estadísticas de la Red:
 - Packet Delivery Rate (PDR): Porcentaje de éxito en la recepción de mensajes desde la red de sensores.
 - Paquetes Totales Recibidos: Número total de mensajes recibidos.
 - Número Total de Nodos Desplegados: Cantidad total de nodos presentes en la red.
 - Latencia Media de la Red: Retardo desde que se envía un mensaje hasta que es recibido en el Gateway.

Esta información se irá modificando cada cierto tiempo, ya que estas estadísticas cambian cada vez que se recibe un mensaje. No necesariamente debe modificarse cada vez que llegue un mensaje, pero si tener un refresco constante para ver cómo evoluciona la instalación.

- Topología de la Red:

La función de la Topología de la Red permite visualizar la estructura de la red de sensores, destacando las conexiones entre los nodos y el Gateway. Se presenta la topología en forma de árbol, lo que permite identificar claramente la conexión entre cada nodo y su nodo padre o siguiente salto en la red. Este enfoque visual facilita la comprensión de la jerarquía y la disposición de los nodos en la red.

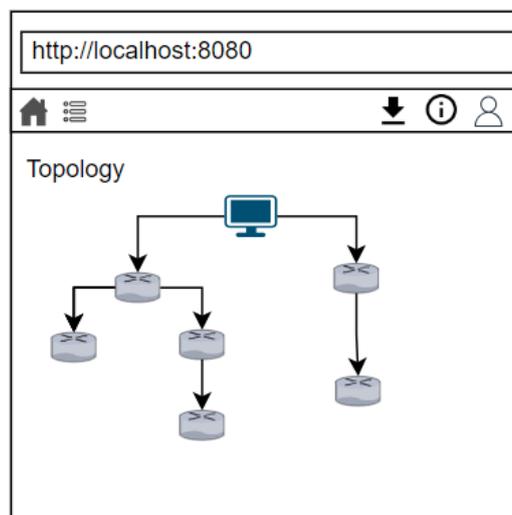


Figura 39: Boceto de la página de la topología de red. (Fuente: Elaboración propia)

La representación de la topología se actualiza dinámicamente a medida que se añaden o modifican nodos en la red. Esto garantiza que la visualización refleje con precisión la configuración actual de la red de sensores, proporcionando a los usuarios información actualizada y precisa sobre la estructura de su red en todo momento.

- Lista de nodos desplegados:
 - ID del nodo: Identificador único del nodo.
 - Estado: Indica si el nodo está operativo o aún no está configurado.
 - Batería: Porcentaje de batería restante en el nodo.
 - Calidad del Enlace: Nivel de calidad de la conexión del nodo.
 - Tiempo del Último Mensaje: Tiempo transcurrido desde la recepción del último mensaje.
 - Localizar: Botón que permite enviar un mensaje al nodo para encender un LED u otra acción similar.

Dado que esta ventana tiene mucha información, puede ser interesante contar dos vistas diferenciadas “Simple/Avanzada”. En la vista Simple, los usuarios podrán acceder a parámetros generales, como el nivel de calidad del enlace. Esta vista proporciona una visión general y simplificada de la información relevante de la red. Por otro lado, en la vista Avanzada se presentan valores más detallados y concretos de cada nodo, permitiendo a los usuarios profundizar en aspectos específicos de la red. Esta vista ofrece una perspectiva más detallada y exhaustiva de los datos.

El cambio entre estas dos vistas se debe diseñar para ser sencillo y rápido, lo que permite a los usuarios seleccionar fácilmente el nivel de detalle que desean visualizar en cada momento.

La implementación de estas dos vistas tiene como objetivo brindar flexibilidad a los usuarios, permitiéndoles acceder rápidamente a información general o detallada según sus necesidades específicas. Ambas vistas están claramente diferenciadas para asegurar que cada una tenga un propósito claro y distintivo.

5.4.8. Datos en tiempo real

La pantalla de visualización de datos en tiempo real permite seleccionar el nodo y el sensor específico que se desea visualizar. Cada nodo puede tener un número determinado de sensores configurados, y se plantea que mediante un menú desplegable se muestren estos sensores para su selección. Una vez seleccionado el nodo y el sensor, se inicia la representación en tiempo real conforme se reciben los datos.

La configuración de las gráficas que se plantea es la siguiente:

- **Ajuste de Límites de los Ejes Verticales:** Se plantea la posibilidad de ajustar los límites de los ejes verticales de manera sencilla. Los ejes se autoajustan en función de los datos recibidos.
- **Eje Horizontal:** Muestra las últimas 100 muestras y se autoajusta para mostrar todas las muestras recibidas mientras no se cambie de ventana.
- **Herramientas Avanzadas:** Se busca una herramienta completa para la creación de gráficas, que permita pausar la gráfica y desplazarse por los últimos valores representados.
- **Representación de Múltiples Flujos de Datos:** Es posible representar dos flujos de datos de diferentes nodos en la misma gráfica, aunque puede haber conflictos si los sensores son de diferente tipo.

Por otro lado, la pantalla mostrará un listado de todos los nodos desplegados, y permitiendo seleccionar el sensor a representar. Se plantea la posibilidad de mostrar todos los sensores disponibles, indicando de qué nodo son, aunque esto podría resultar en un listado muy largo.

Inicialmente se plantea tener tres tipos de gráficas diferenciadas, dependiendo del tipo de sensor que se vaya a visualizar:

- **Valores Discretos:** Utiliza un gráfico de barras u otro tipo similar, donde los valores anteriores en el tiempo no tienen relevancia.
- **Valores Continuos:** Se emplea un gráfico de puntos para visualizar los valores anteriores en el eje de tiempo.

- **Valores Acumulados:** Se utiliza un gráfico escalonado para mostrar valores anteriores cercanos en el tiempo, permitiendo observar la diferencia entre el totalizador en instantes anteriores y el totalizador actual.

Por último, se plante mostrar la información general del nodo seleccionado, con diferentes datos como los sensores configurados, el ID del nodo, el nombre, la IPv6, etc. Esta información ayudará a entender mejor los datos en vivo que se están analizando.

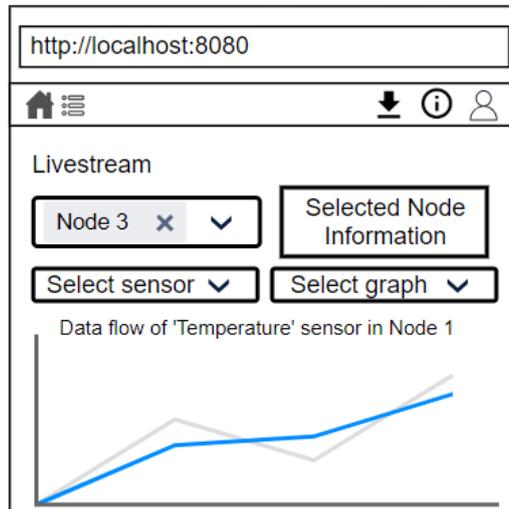


Figura 40: Boceto de la página de datos en vivo. (Fuente: Elaboración propia)

La pantalla de visualización de datos en tiempo real se ha diseñado para proporcionar una experiencia completa y adaptable a las necesidades específicas de los usuarios, ofreciendo una amplia variedad de opciones de visualización y configuración.

5.4.9. Configuración

La funcionalidad de configuración permite ajustar los parámetros de red y servicios de la pasarela de manera sencilla y directa. La pantalla de configuración es simple y se utiliza para actualizar cambios y enviarlos a la pasarela mediante peticiones API/REST.

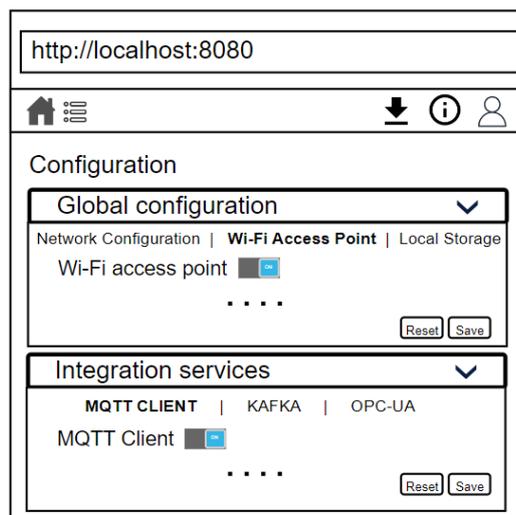


Figura 41: Boceto de la página de configuración de red. (Fuente: Elaboración propia)

- Configuración de red:
 - DHCP: Permite activar o desactivar el servicio DHCP. Cuando DHCP está activado, los campos de IP, máscara y puerta de enlace se pueden deshabilitar.
 - Dirección IP: Se puede introducir manualmente la dirección IP, verificando si es correcta.
 - Máscara de Red y Puerta de Enlace Predeterminada: Se introducen de manera similar a la dirección IP.
 - mDNS Hostname: Campo para indicar el nombre de host para el servicio mDNS.

Se debe plantear la gestión de cambios en la dirección IP, cuando se modifican estos parámetros, la dirección IP de la pasarela se actualizará, lo que reiniciará el acceso a la web. La gestión de esto se debe estudiar para ser realizada mediante la mejor forma posible.

- Configuración del Punto de Acceso WiFi:
 - SSID WiFi: Nombre de la red WiFi para el acceso local.
 - Contraseña WiFi: Contraseña para conectarse al punto de acceso WiFi de la pasarela.
- Integración con otras plataformas:
 - Configuración MQTT:
 - Dirección del Broker MQTT.
 - Puerto MQTT.
 - Client ID, Username y Password.
 - Formato del Topic para Publicaciones: %red%nodo%sensor.
 - Configuración OPC-UA
 - Configuración KAFKA

Cada una de estas secciones se puede configurar de manera independiente, lo que permite actualizar la información para MQTT, OPC-UA o Kafka por separado. Se dispone de una función REST para cada uno de estos bloques, y se han diferenciado en tres secciones distintas para una gestión más clara y organizada.

La pantalla de configuración estará diseñada para brindar una experiencia de usuario fluida y eficiente, permitiendo ajustes precisos y configuraciones personalizadas según las necesidades específicas de cada usuario.

6. Resultado y aplicación web final.

Esta sección ofrece una visión detallada de los componentes fundamentales y las decisiones arquitectónicas que han dado forma a la plataforma. Este análisis exhaustivo examina la estructura subyacente de la aplicación, desde la capa de presentación hasta la capa de datos, destacando las tecnologías empleadas, los patrones de diseño implementados y los principios de usabilidad adoptados.

A lo largo de esta sección, se exploran las estrategias utilizadas para garantizar la eficacia, el rendimiento y la escalabilidad de la aplicación web, así como para proporcionar una experiencia de usuario intuitiva y satisfactoria. Se analizan en detalle las interacciones entre los diferentes componentes del sistema, destacando su cohesión y su contribución a la funcionalidad global de la aplicación.

Al comprender el diseño y la arquitectura subyacentes de la aplicación web, se sientan las bases para una evaluación exhaustiva del proyecto, la identificación de áreas de mejora y la planificación de futuras expansiones y actualizaciones. Esta sección desempeña un papel crucial en la comprensión integral de la aplicación y en la optimización de su rendimiento y usabilidad a largo plazo.

6.1. Pantalla de inicio de sesión (Login)

La página de inicio de sesión (Login) es el punto de entrada principal para los usuarios de la aplicación. Está diseñada para autenticar a los usuarios y permitirles acceder a las funcionalidades protegidas de la aplicación. Esta página presenta un diseño limpio y profesional, con un formulario de login centrado verticalmente en la pantalla y con un fondo de imagen que proporciona una atmósfera atractiva. La barra de herramientas en la parte superior muestra el título "ITI - QDSN LOGIN", proporcionando orientación al usuario sobre el propósito de la página.

El formulario de inicio de sesión consta de dos campos de entrada: uno para el nombre de usuario y otro para la contraseña. Estos campos están diseñados con la capacidad de previsualizar la contraseña mediante un icono oculto/visible.

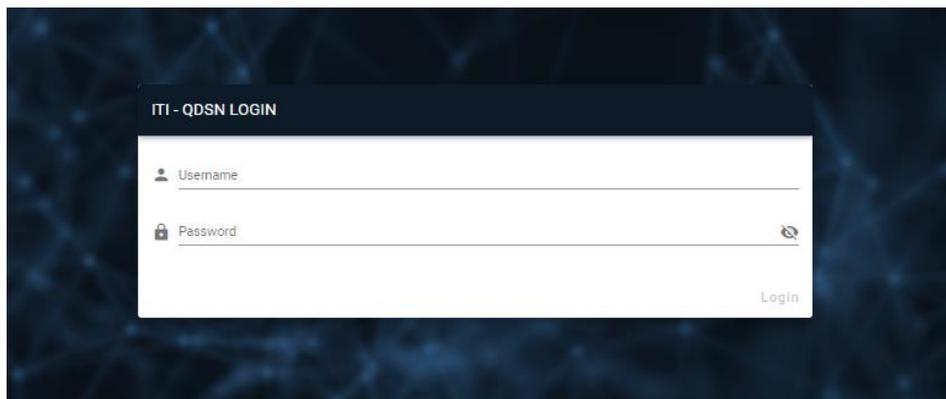


Figura 42: Pantalla de inicio de sesión. (Fuente: Elaboración propia)

La función '*signIn()*' se activa cuando se hace clic en el botón "**Login**". Esta función recopila los datos del formulario (nombre de usuario y contraseña) y los envía como parámetro al método '*signIn*' del componente '**CommonModule**', la cual se muestra a continuación:

```

1  signIn: ({ commit }, credentials) => {
2    console.log("sign in");
3    var userPass = JSON.stringify(credentials);
4    axios.post(process.env.VUE_APP_API_BACK + "login", userPass,
5      {
6        headers: {
7          "Content-Type": "application/json"
8        },
9      })
10   .then((response) => {
11     // Save the token, refreshToken and username on local info
12     tokenService.storeToken(response.data.access_token);
13     tokenService.storeRefreshToken(response.data.refresh_token);
14     usernameService.storeUsername(credentials.username);
15
16     commit('setLoginValue', true);
17     // Connect websocket service
18     store.dispatch("RealTimeModule/connect");
19
20     // Redirects to main view
21     router.push({ path: '/' });
22   })
23   .catch((error) => {
24     store.dispatch('CommonModule/openAndFillSnackbar',
25       {
26         showSnackbar: true,
27         snackbarText: 'Login error',
28         snackbarType: 'error'
29       });
30
31     tokenService.clearToken();
32     usernameService.clearUsername();
33   })
34 },

```

Fragmento de código 11: Función "signIn". (Fuente: Elaboración propia)

La función **'signIn'** forma parte del módulo **'CommonModule'** y se encarga de gestionar el proceso de inicio de sesión de los usuarios en la aplicación web. Su objetivo es enviar las credenciales del usuario al servidor, procesar la respuesta y realizar las acciones correspondientes según el resultado.

La función comienza imprimiendo en la consola el mensaje "sign in", proporcionando una indicación útil para el seguimiento del flujo de ejecución durante el proceso de inicio de sesión.

Posteriormente, las credenciales del usuario se convierten en una cadena JSON utilizando el método **'JSON.stringify()'**. Esta cadena JSON se enviará al servidor como parte de la solicitud HTTP.

Utilizando la biblioteca Axios⁶³, se realiza una solicitud POST al endpoint de inicio de sesión del API back-end, el cual se explica en el punto 5.4.1. de este documento. Como parámetro de entrada del endpoint, se incluye la cadena JSON de las credenciales en el cuerpo de la solicitud y se establece el encabezado de la solicitud para indicar que se está enviando un objeto JSON.

La función `‘.then()’` maneja la respuesta exitosa del servidor. El token de acceso y el token de actualización devueltos por el servidor se almacenan utilizando los servicios `‘tokenService’` y `‘usernameService’`, respectivamente. Estos tokens son cruciales para autenticar las solicitudes futuras del usuario y para identificar al usuario en la aplicación.

El objeto `tokenService` es una instancia de la clase `‘TokenService’`, diseñada para manejar tokens de autenticación en una aplicación web. Proporciona métodos para verificar la autenticación del usuario, almacenar y recuperar tokens de acceso y de actualización, y limpiar los tokens del almacenamiento local del navegador cuando sea necesario. Esta instancia se utiliza globalmente en la aplicación para gestionar la autenticación del usuario de manera eficiente.

Por otro lado, el objeto `usernameService` es una instancia de la clase `UsernameService`, diseñada para manejar el nombre de usuario en una aplicación web. Proporciona métodos para almacenar, recuperar y limpiar el nombre de usuario en el almacenamiento local del navegador. Esta instancia se utiliza globalmente en la aplicación para administrar el nombre de usuario de manera eficiente.

Además, se realiza una mutación en el store de Vuex para actualizar el estado de `loginValue`, indicando que el usuario ha iniciado sesión correctamente. Luego, se llama a la función `‘connect’` del módulo `‘RealTimeModule’`, la cual tiene como objetivo establecer una conexión WebSocket⁶⁴ con el servidor para recibir y procesar datos en tiempo real sobre la información de los sensores y la información de la red WSN (Wireless Sensor Network).

```
1 connect: ({ commit }) => {
2   instance = websocketsInstance();
3   instance.connect();
4   instance.onMessageArrivedSensorsInfo((message: any) => {
5     commit('setSensorsInfo', message)
6   })
7   instance.onMessageArrivedWSNInfo((message: any) => {
8     commit('setWsnInfo', message)
9   })
10 },
```

Fragmento de código 12: Función “connect” del módulo “RealTimeModule”. (Fuente: Elaboración propia)

Finalmente, el usuario es redirigido a la página principal de la aplicación utilizando el enrutador `Vue Router`.

⁶³ Axios es una biblioteca JavaScript que se utiliza para realizar solicitudes HTTP desde el navegador o desde Node.js. Proporciona una interfaz simple y fácil de usar para realizar solicitudes HTTP como GET, POST, PUT y DELETE, y admite características como la gestión de datos JSON, la autenticación de tokens y la gestión de errores de manera eficiente.

⁶⁴ Una conexión WebSocket es una tecnología de comunicación bidireccional y en tiempo real que se establece entre un cliente y un servidor a través de un protocolo WebSocket. A diferencia de HTTP, que es un protocolo de comunicación unidireccional y basado en solicitudes, WebSocket permite una comunicación full-duplex, lo que significa que tanto el cliente como el servidor pueden enviar y recibir datos simultáneamente.

En caso de que se produzca un error durante el proceso de inicio de sesión, la función `‘catch()’` maneja la excepción. Se despacha una acción para abrir un Snackbar⁶⁵ con un mensaje de error, proporcionando retroalimentación al usuario sobre el problema. Además, se eliminan cualquier token de acceso previamente almacenado y se limpia el nombre de usuario del servicio correspondiente, asegurando una gestión adecuada de los datos de autenticación en caso de fallo en el inicio de sesión.

En resumen, la función `‘signIn’` gestiona de manera integral el proceso de inicio de sesión del usuario, garantizando una experiencia fluida y segura en la aplicación web.

6.2. Pantalla principal de la aplicación (MainLayout)

En el cuerpo principal de la aplicación web se encuentran los componentes esenciales que estarán presentes durante toda la navegación del usuario. Estos componentes incluyen el AppSnackbar, que se utiliza para mostrar notificaciones al usuario. Estas notificaciones pueden abarcar una variedad de tipos, tales como mensajes de éxito, advertencias o errores, y aparecen temporalmente en la pantalla para mantener al usuario informado sobre el estado de diversas acciones realizadas en la aplicación.

Otro componente crucial es el AppToolbar, que es la barra de navegación superior. Esta barra proporciona una interfaz consistente y accesible para la navegación dentro de la aplicación. En el extremo derecho de esta barra se encuentra un botón que muestra el nombre del usuario actual. Al hacer clic en este botón, se despliega un menú con tres opciones principales: cambiar el idioma, cambiar la contraseña y cerrar sesión. Este diseño facilita al usuario el acceso rápido a funciones importantes sin tener que abandonar la página actual.

El AppNavigationDrawer es otro componente fundamental que se encuentra en el cuerpo principal de la aplicación. Este menú lateral está ubicado en la parte izquierda de la pantalla y, al ingresar a la aplicación, se presenta colapsado para maximizar el espacio útil en la pantalla principal. Sin embargo, si el usuario desplaza el cursor hacia la izquierda y lo coloca sobre el menú, este se expande completamente, mostrando todas las opciones de navegación disponibles. Este comportamiento permite una interfaz limpia y eficiente, donde el menú no interfiere con el contenido principal a menos que sea necesario.

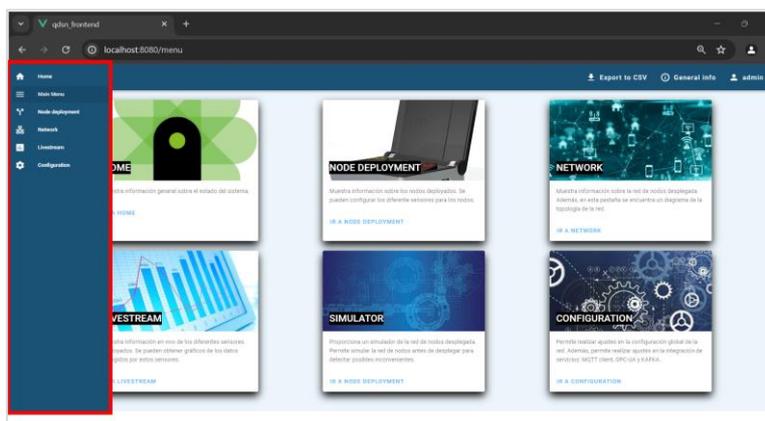


Figura 43: Menú vertical de navegación de la aplicación. (Fuente: Elaboración propia)

⁶⁵ Un Snackbar es un componente de interfaz de usuario utilizado en aplicaciones web y móviles para mostrar mensajes breves y no intrusivos al usuario. Se presenta de forma temporal en la parte inferior o superior de la pantalla, desapareciendo automáticamente después de unos segundos. Puede ser interactivo, permitiendo al usuario realizar acciones como deshacer una operación. Por lo general, tiene un diseño discreto que se integra con la interfaz de usuario y puede personalizarse para adaptarse al estilo de la aplicación. Se utiliza para proporcionar retroalimentación rápida sobre acciones realizadas, errores o eventos importantes.

Todos estos componentes están organizados en la carpeta /components en el directorio raíz del proyecto. Esta organización facilita la gestión y el mantenimiento del código, ya que agrupa funcionalidades similares en un solo lugar, permitiendo a los desarrolladores localizar y modificar componentes con mayor facilidad.

Al lado del botón de usuario en la barra de navegación superior, se encuentra un botón adicional para exportar datos a CSV. Este botón permite a los usuarios seleccionar un rango de tiempo para exportar los datos relevantes de la aplicación. Para especificar el rango de tiempo, el usuario debe elegir una fecha y hora de inicio y fin mediante un componente específico. Este componente utiliza la librería externa vuetify-datetime-picker, que proporciona una interfaz intuitiva para la selección de fechas y horas.

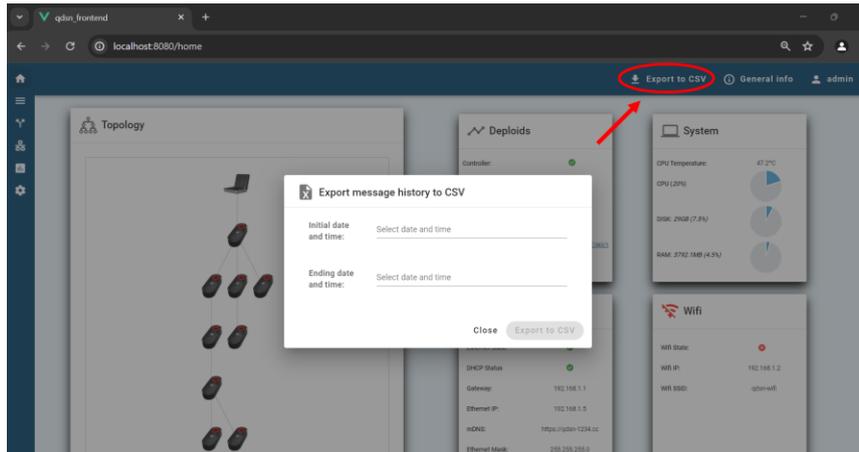


Figura 44: Funcionalidad de exportar a CSV de la aplicación. (Fuente: Elaboración propia)

Al pulsar sobre alguno de los campos de fecha, aparecerá el componente para elegir la fecha. Una vez seleccionada la fecha, el componente permite elegir la hora correspondiente. Este proceso se repite para tanto la fecha y hora de inicio como la de fin. Una vez que el usuario ha seleccionado ambos rangos de tiempo, puede pulsar el botón de exportación para generar el archivo CSV.

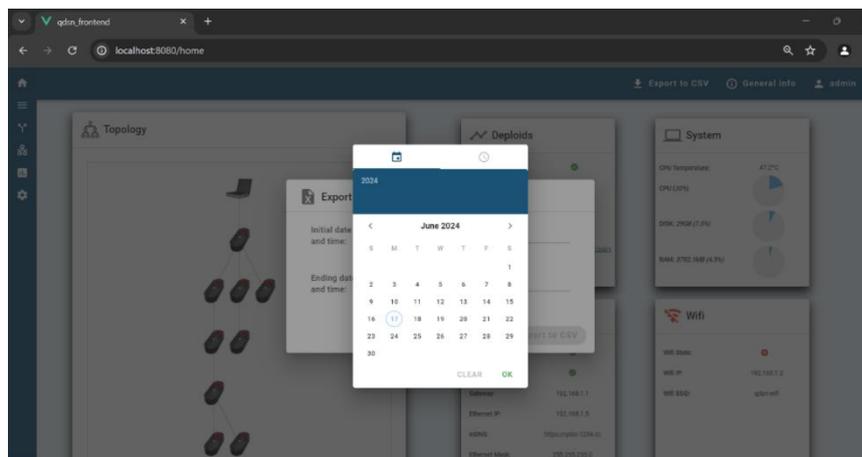


Figura 45: Elección de fecha en la funcionalidad de exportar a CSV. (Fuente: Elaboración propia)

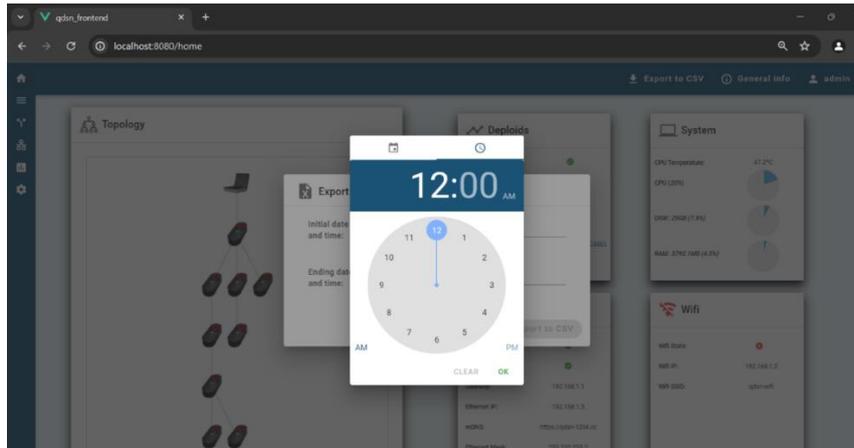


Figura 46: Elección de horario en la funcionalidad de exportar a CSV. (Fuente: Elaboración propia)

La funcionalidad que recupera la información necesaria para el fichero CSV y lo crea se encuentra en el módulo **'CommonModule'**. Este módulo maneja las llamadas a los servicios necesarios para obtener los datos dentro del rango de tiempo especificado, formatear estos datos adecuadamente y generar el archivo CSV para su descarga.



Figura 47: Vista del desplegable para exportar a CSV tras elegir fechas. (Fuente: Elaboración propia)

Dentro de la barra superior, al hacer clic en el botón que muestra el nombre del usuario, se despliega un menú con varias opciones. La opción de cambiar el idioma despliega una lista en la parte izquierda de la pantalla mostrando los idiomas disponibles para la aplicación. Este componente, denominado **'AppLanguageButton'**, permite al usuario seleccionar entre inglés y español. La lista se despliega de manera intuitiva, permitiendo al usuario cambiar el idioma de la interfaz rápidamente y sin complicaciones.

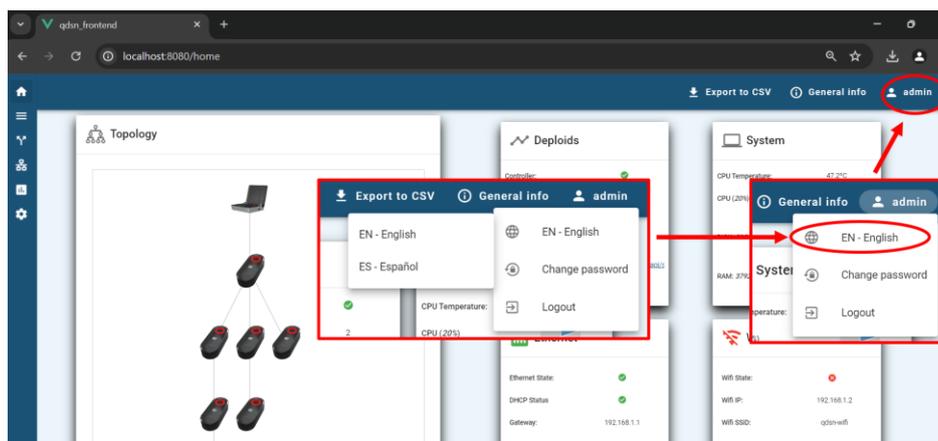


Figura 48: Funcionalidad de elección de idioma en la aplicación. (Fuente: Elaboración propia)

Si el usuario selecciona la opción de cambiar la contraseña, se abre una ventana emergente específica para esta tarea. Este componente, llamado **'ChangePasswordDialog'**, ha sido desarrollado y ubicado en un subdirectorio dentro del directorio **'MainLayout'**. Esta estructura organizativa no solo facilita la extensión y modificación del componente en el futuro, sino que también mantiene el código ordenado y coherente.

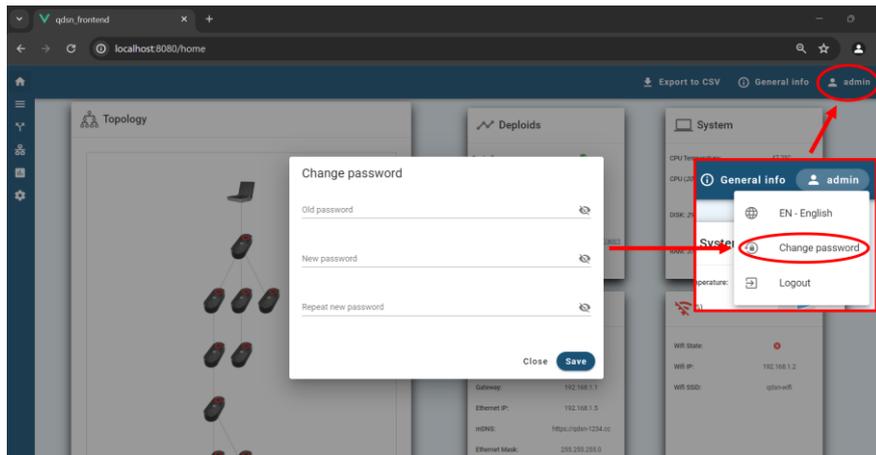


Figura 49: Funcionalidad de cambiar contraseña en la aplicación. (Fuente: Elaboración propia)

Finalmente, al seleccionar la opción de cerrar sesión, el usuario es redirigido de vuelta a la pantalla de inicio de sesión. Este proceso asegura que la sesión actual se cierre correctamente, protegiendo la seguridad de la cuenta del usuario. El diseño de esta funcionalidad está pensado para ser intuitivo y seguro, minimizando el riesgo de accesos no autorizados.

En resumen, el cuerpo principal de la aplicación web está diseñado para proporcionar una navegación intuitiva y accesible, con componentes clave organizados de manera eficiente para mejorar la experiencia del usuario y facilitar el mantenimiento del código. La adición de funcionalidades como la exportación a CSV, cambio de idioma, y gestión de contraseñas, asegura que los usuarios tengan acceso a herramientas poderosas y flexibles para interactuar con la aplicación de manera efectiva.

6.3. Pantalla de inicio (Home)

En la pantalla principal de la aplicación (Home), se presentan cinco tarjetas informativas cuidadosamente diseñadas para proporcionar una visión completa y detallada del estado de la red de sensores WSN (Wireless Sensor Network) y del sistema en general. Cada una de estas tarjetas cumple una función específica, permitiendo al usuario monitorizar y gestionar eficientemente diversos aspectos de la red y los recursos del sistema.

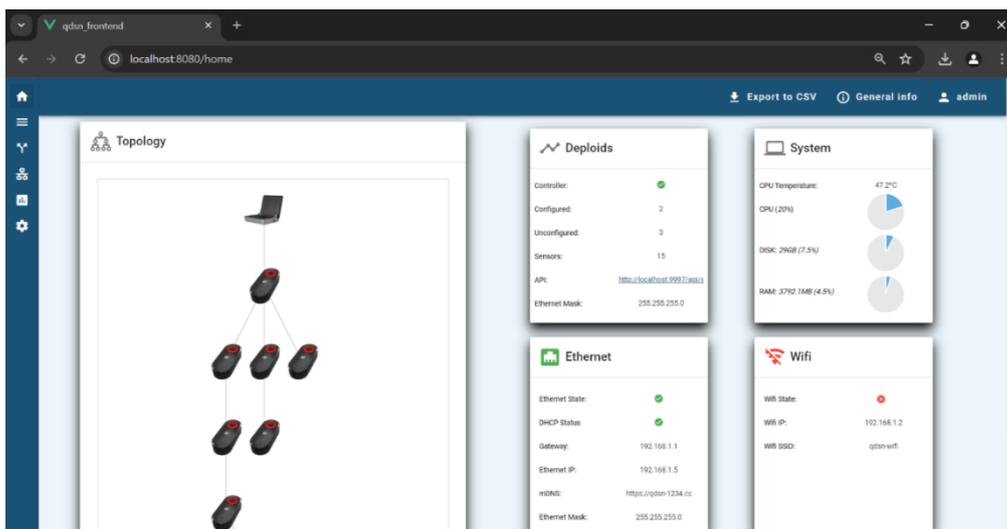


Figura 50: Página de inicio (Home) de la aplicación. (Fuente: Elaboración propia)

6.3.1. Primera Tarjeta: Gráfico de Topología de la Red

La primera tarjeta está dedicada a mostrar el gráfico de topología de la red de sensores WSN. Este gráfico ofrece una visualización detallada de cómo están conectados los nodos dentro de la red. A través de esta representación gráfica, el usuario puede entender rápidamente la estructura y las conexiones de los sensores, identificando las relaciones y dependencias entre ellos. La topología puede mostrar nodos individuales, enlaces entre nodos y posibles rutas de comunicación. Esto es particularmente útil para detectar problemas como nodos desconectados o áreas de la red que pueden necesitar optimización. La capacidad de ver la red de manera gráfica facilita la identificación de posibles problemas o áreas de interés dentro de la red.



Figura 51: Tarjeta que muestra el gráfico de topología de la red. (Fuente: Elaboración propia)

6.3.2. Segunda Tarjeta: Estado del Despliegue de la Red (Deploids)

La segunda tarjeta, denominada "Deploids", proporciona información crítica sobre el estado del despliegue de la red de nodos. En esta tarjeta se incluye:

- Estado del Controlador: Indicado por un icono que se ilumina en verde si el controlador está activo y en rojo si está inactivo. Esto permite al usuario ver de un vistazo si el controlador principal de la red está funcionando correctamente.
- Número de Nodos Configurados y No Configurados: Muestra cuántos nodos están correctamente configurados y operativos, y cuántos aún no han sido configurados, lo cual es esencial para el mantenimiento y la gestión de la red.
- Número de Sensores: Proporciona el total de sensores conectados a la red, permitiendo al usuario tener una visión clara de la escala de la red.
- URL de la API: Un enlace clicable que permite al usuario navegar directamente a la interfaz de la API, facilitando el acceso a funciones avanzadas y configuraciones detalladas.
- Máscara de Subred Ethernet: Muestra la máscara de subred utilizada, esencial para la configuración de red y resolución de problemas.

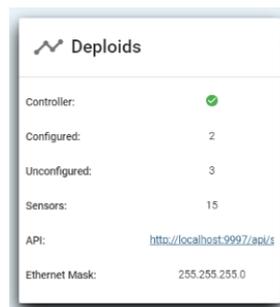


Figura 52: Tarjeta resumen del estado de despliegue de nodos. (Fuente: Elaboración propia)

6.3.3. Tercera Tarjeta: Estado General del Sistema (System)

La tercera tarjeta, denominada "System", ofrece una visión general del estado del sistema con varios gráficos de tarta que representan:

- Temperatura de la CPU: Muestra la temperatura actual de la CPU, permitiendo al usuario monitorear el enfriamiento y prevenir sobrecalentamientos que podrían dañar el hardware.
- Estado de la CPU: Representa el uso de la CPU en porcentaje, proporcionando una visión clara de la carga de trabajo actual y permitiendo la detección de procesos que puedan estar consumiendo demasiados recursos.
- Estado de la RAM: Muestra el uso de la memoria RAM, lo cual es crucial para asegurar que el sistema tenga suficiente memoria disponible para operar eficientemente.
- Estado del Disco: Indica el uso del almacenamiento en disco, permitiendo al usuario monitorear el espacio disponible y prever la necesidad de ampliación de almacenamiento.

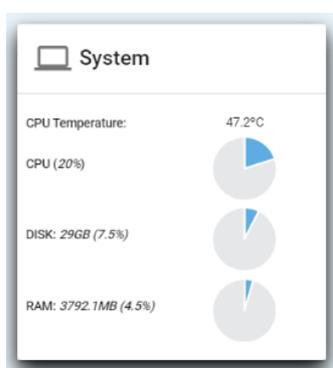


Figura 53: Tarjeta resumen del estado del sistema. (Fuente: Elaboración propia)

6.3.4. Cuarta Tarjeta: Estado de la Conexión Ethernet

La cuarta tarjeta, denominada "Ethernet", muestra el estado de la conexión Ethernet de manera detallada:

- Estado de Conectividad: Representado por un icono que se ilumina en verde si la conexión está activa y en rojo si no lo está, proporcionando una indicación rápida del estado de la red.
- Estado DHCP: También representado por un icono, indica si el DHCP está habilitado y funcionando correctamente.
- IP del Gateway: Muestra la dirección IP del gateway, esencial para la configuración de red y resolución de problemas.
- IP Ethernet: La dirección IP asignada a la interfaz Ethernet del dispositivo, fundamental para la identificación y gestión del dispositivo en la red.
- URL mDNS: Proporciona la URL mDNS para la detección y acceso fácil al dispositivo en redes locales.
- Máscara de Subred Ethernet: Similar a la segunda tarjeta, muestra la máscara de subred utilizada para la interfaz Ethernet.

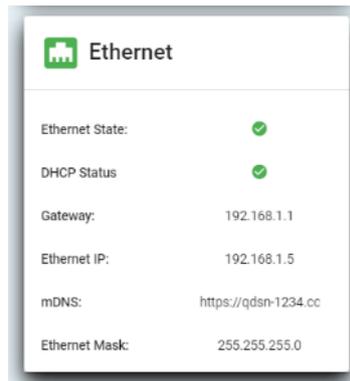


Figura 54: Tarjeta del estado Ethernet de la aplicación. (Fuente: Elaboración propia)

6.3.5. Quinta Tarjeta: Conectividad WiFi

La última tarjeta se dedica a mostrar la conectividad WiFi, proporcionando una visión similar a la tarjeta de Ethernet:

- Estado de Conectividad WiFi: Indicado por un icono que muestra si la conexión WiFi está activa (verde) o no (rojo).
- IP WiFi: La dirección IP asignada a la interfaz WiFi, crucial para la identificación y gestión del dispositivo en la red inalámbrica.
- SSID: El nombre de la red WiFi a la que está conectado el dispositivo, lo que ayuda al usuario a confirmar que está conectado a la red correcta.

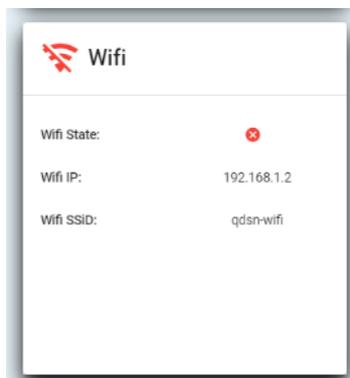


Figura 55: Tarjeta del estado Wi-Fi de la aplicación. (Fuente: Elaboración propia)

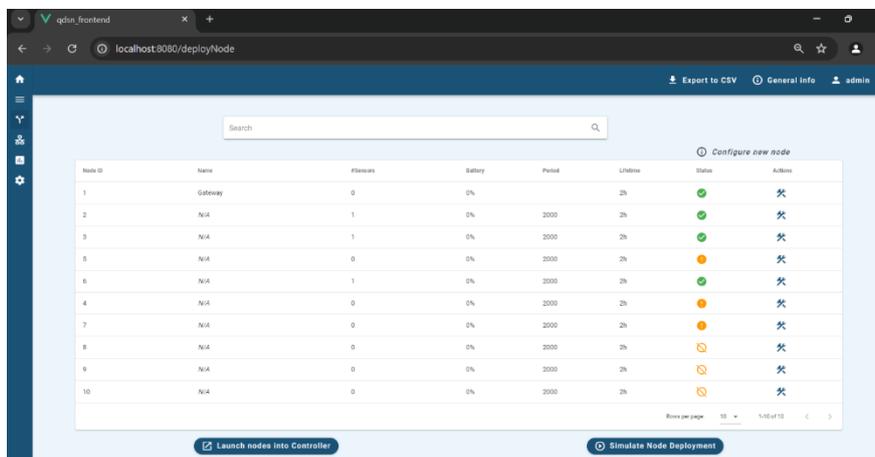
En resumen, la pantalla principal de la aplicación está diseñada para ofrecer una interfaz intuitiva y detallada que permite al usuario monitorear y gestionar eficazmente la red de sensores WSN y los recursos del sistema. Cada tarjeta proporciona información clave de manera clara y accesible, facilitando la identificación de problemas y la toma de decisiones informadas para mantener el sistema en óptimas condiciones. La estructura modular de estas tarjetas permite una gestión eficiente y una experiencia de usuario mejorada.

6.4. Pantalla de despliegue de nodos (Node Deployment)

En la pestaña denominada "Despliegue de Nodos", se presenta una interfaz diseñada para proporcionar una visión completa y detallada del estado y configuración de todos los nodos de la red de sensores WSN. Esta sección es esencial para la gestión y monitoreo eficiente de los nodos, ofreciendo varias funcionalidades clave organizadas en una tabla resumen, una barra de búsqueda, y botones de acción.

La integración de estas funcionalidades en la pestaña "Despliegue de Nodos" está diseñada para proporcionar una experiencia de usuario eficiente y efectiva. La tabla resumen ofrece una visión detallada y gestionable de los nodos, mientras que la barra de búsqueda facilita la navegación y el acceso rápido a la información. Los botones de acción proporcionan las herramientas necesarias para interactuar y gestionar la red de manera proactiva.

Cada componente de esta pestaña está diseñado para trabajar en conjunto, proporcionando al usuario un control completo sobre el despliegue y la gestión de los nodos de la red de sensores WSN. La organización lógica y la presentación clara de la información aseguran que el usuario pueda tomar decisiones informadas y ejecutar acciones de manera rápida y eficiente.



| Node ID | Name | #Sensors | Battery | Pinout | Lifetime | Status | Actions |
|---------|---------|----------|---------|--------|----------|--------|---------|
| 1 | Gateway | 0 | 0% | 2000 | 2h | Green | ✖ |
| 2 | N/A | 1 | 0% | 2000 | 2h | Green | ✖ |
| 3 | N/A | 1 | 0% | 2000 | 2h | Green | ✖ |
| 5 | N/A | 0 | 0% | 2000 | 2h | Yellow | ✖ |
| 6 | N/A | 1 | 0% | 2000 | 2h | Green | ✖ |
| 4 | N/A | 0 | 0% | 2000 | 2h | Yellow | ✖ |
| 7 | N/A | 0 | 0% | 2000 | 2h | Yellow | ✖ |
| 8 | N/A | 0 | 0% | 2000 | 2h | Yellow | ✖ |
| 9 | N/A | 0 | 0% | 2000 | 2h | Yellow | ✖ |
| 10 | N/A | 0 | 0% | 2000 | 2h | Yellow | ✖ |

Figura 56: Página de despliegue de nodos de la aplicación. (Fuente: Elaboración propia)

6.4.1. Tabla resumen de nodos

La pieza central de esta pestaña es una tabla resumen que contiene información crucial sobre todos los nodos de la red. Las columnas de esta tabla incluyen:

- **ID del Nodo:** Un identificador único para cada nodo, que permite al usuario diferenciar y gestionar cada nodo individualmente.
- **Nombre:** El nombre asignado a cada nodo, que puede ser utilizado para identificar fácilmente la función o ubicación del nodo dentro de la red. Este nombre se puede cambiar pulsando en el botón de la columna "acciones", como se verá posteriormente.
- **Número de Sensores Configurados:** Muestra la cantidad de sensores que están configurados y operativos en cada nodo, proporcionando una visión clara de la capacidad de cada nodo.
- **Batería del Nodo:** Indica el nivel de batería de cada nodo, permitiendo al usuario monitorear y planificar el reemplazo o recarga de baterías.

- **Periodo:** El intervalo de tiempo en el que el nodo recoge y envía datos, lo cual es esencial para la sincronización y eficiencia de la red.
- **Tiempo de Vida:** Muestra una estimación del tiempo de vida restante del nodo basado en su uso actual y nivel de batería.
- **Estado:** Indica el estado operativo del nodo con tres posibles estados visualizados mediante iconos específicos:
 - **Configurado:** Representado por el icono verde de la Figura 57. Este estado indica que el nodo está configurado y operativo sin cambios pendientes.
 - **Con cambios:** Representado por el segundo icono de la Figura 57. Este estado muestra que hay cambios realizados en la configuración del nodo que aún no se han desplegado al controlador.
 - **Sin cambios:** Representado por el tercer icono de la Figura 57. Indica que el nodo no ha tenido cambios en su configuración.

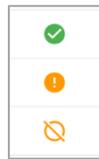


Figura 57: Diferentes estados de los nodos. (Fuente: Elaboración propia)

- **Acciones:** Proporciona botones o enlaces para realizar acciones específicas sobre cada nodo, al pulsar se abre una ventana de configuración como se aprecia en la Figura 58, la cual se explicará en el punto 6.4.2. Configuración del nodo.

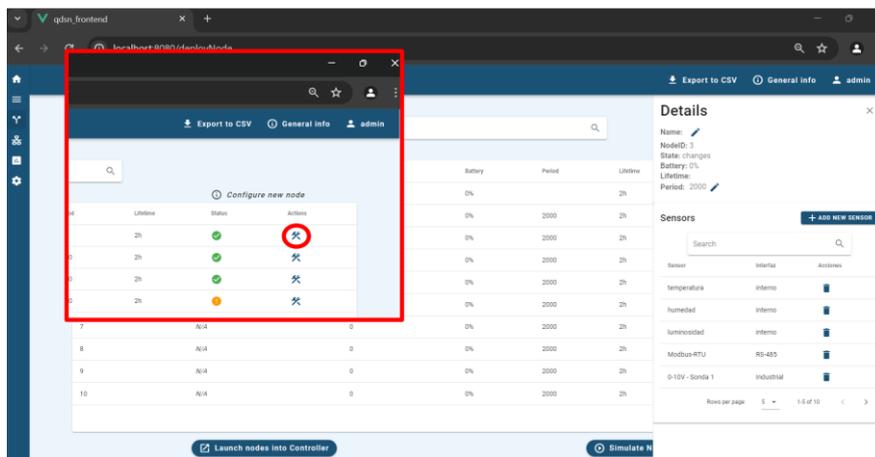


Figura 58: Botón de acción en la pantalla de despliegue de nodos. (Fuente: Elaboración propia)

6.4.2. Configuración del nodo

Como se ha visto en el apartado anterior y en la Figura 58, en la pestaña denominada "Despliegue de Nodos", se ofrece una funcionalidad avanzada que permite al usuario explorar y gestionar en detalle cada nodo de la red de sensores WSN. Al hacer clic en el botón de la columna de acciones de cualquier nodo en la tabla resumen, se despliega un menú lateral en la parte derecha de la pantalla. Este menú, llamado "**Detalles del Nodo**" o "**Details**", proporciona una visión detallada y comprensible del nodo seleccionado, facilitando la administración precisa de sus configuraciones y estados.

El menú "Detalles del Nodo" despliega información clave del nodo seleccionado, tales como su nombre, ID del nodo, estado actual, nivel de batería, tiempo de vida estimado y el periodo de transmisión de datos. Esta presentación detallada permite al usuario comprender rápidamente el estado operativo del nodo y tomar decisiones informadas sobre su gestión. Además, como se observa en la Figura 59 el usuario, en este menú, puede cambiar el nombre del nodo y el periodo de transmisión clicando en el botón representado por un lápiz junto al nombre o el periodo, respectivamente.

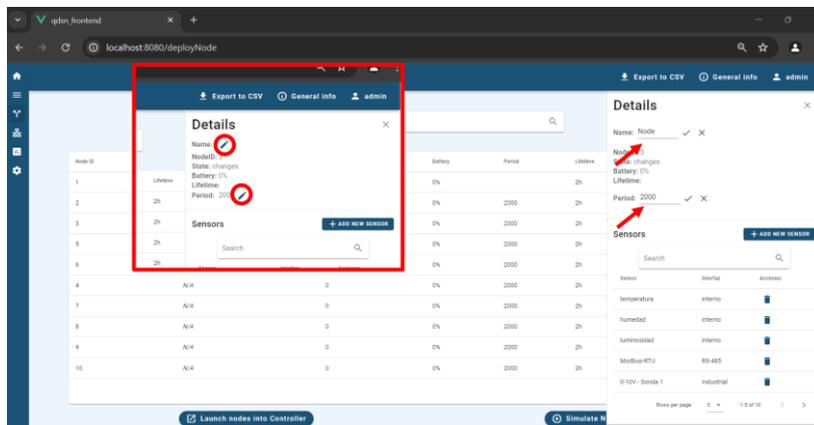


Figura 59: Funcionalidad de cambiar el nombre o periodo de los nodos. (Fuente: Elaboración propia)

Debajo de estos detalles generales, el menú incluye una sección dedicada a los sensores conectados al nodo. Esta sección muestra una tabla con tres columnas: "Sensor", que indica el tipo o nombre del sensor; "Interfaz del Sensor", que especifica cómo está conectado el sensor al nodo; y "Acciones", que contiene un botón para eliminar el sensor seleccionado del nodo. Esta tabla proporciona una visión clara y concisa de los sensores actualmente configurados en el nodo, permitiendo una gestión fácil y eficiente.

Encima de la tabla de sensores, un botón etiquetado como "**Add Sensor**" o "**Añadir Sensor**" facilita la expansión de las capacidades del nodo. Al hacer clic en este botón, se abre un pop-up que presenta una lista de sensores disponibles para ser añadidos al nodo. El usuario puede seleccionar uno o varios sensores de esta lista y, al pulsar el botón "**Save**" o "**Guardar**", los sensores seleccionados se añaden a la tabla en el menú de detalles del nodo. Este proceso es directo y eficiente, y al completarse, el estado del nodo se actualiza automáticamente a "**Changed**" o "**Con cambios**", indicando que hay modificaciones pendientes de ser aplicadas.

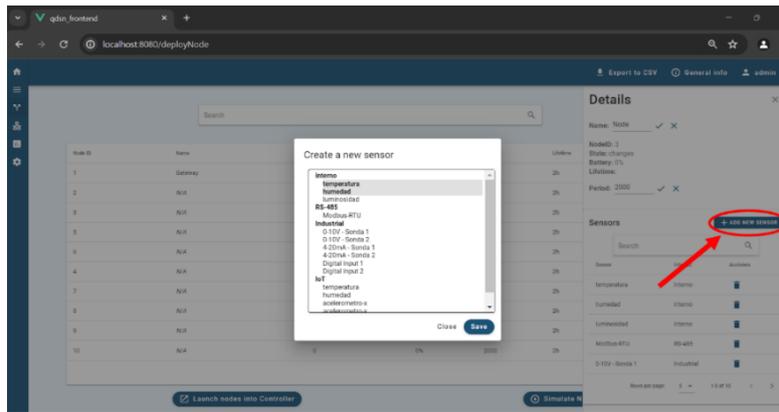


Figura 60: Funcionalidad de añadir sensores a un nodo en la aplicación. (Fuente: Elaboración propia)

La integración de estas funcionalidades en la pestaña "Despliegue de Nodos" está cuidadosamente diseñada para proporcionar una experiencia de usuario fluida y eficaz. El acceso a detalles específicos de cada nodo, combinado con la capacidad de gestionar sensores de manera dinámica, permite a los usuarios mantener un control preciso sobre la configuración y el estado de cada nodo. Esta interfaz no solo facilita la supervisión y gestión de la red de sensores, sino que también asegura que cualquier cambio necesario se pueda realizar de manera rápida y sin complicaciones.

El diseño intuitivo y la organización lógica de esta pestaña garantizan que los usuarios puedan navegar y operar sin esfuerzo. La capacidad de añadir y eliminar sensores mediante una interfaz amigable y la actualización automática del estado del nodo aseguran que la red se mantenga adaptable y eficiente. Esta atención al detalle en la funcionalidad y usabilidad del menú "Detalles del Nodo" demuestra un enfoque centrado en el usuario, proporcionando las herramientas necesarias para gestionar la red de sensores WSN de manera óptima y efectiva.

6.4.3. Barra de búsqueda

Encima de la tabla se encuentra una barra de búsqueda que permite al usuario filtrar rápidamente los nodos según diferentes criterios. Esta funcionalidad es esencial para manejar grandes cantidades de nodos, permitiendo encontrar rápidamente un nodo específico o un grupo de nodos con características similares. La barra de búsqueda admite la entrada de texto libre, lo que facilita la búsqueda por ID del nodo, nombre, estado, u otros atributos. Además, las columnas de la tabla se pueden ordenar clicando sobre la cabecera de la columna por la cual se quiere ordenar, por ejemplo, al clicar sobre "Nombre", la tabla se ordenará por nombre en orden alfabético.

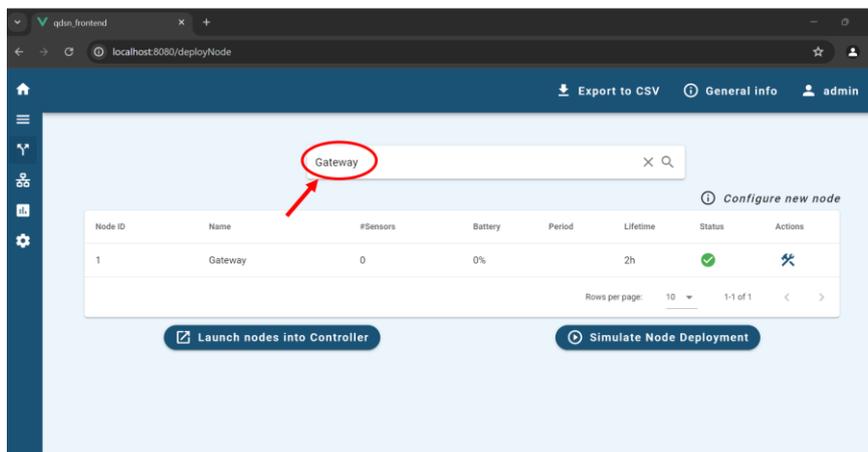


Figura 61: Barra de búsqueda en la pantalla de despliegue de nodos de la aplicación. (Fuente: Elaboración propia)

6.4.4. Botones de acción

Debajo de la tabla, se encuentran dos botones de acción que permiten al usuario interactuar con la red de manera más avanzada:

- Lanza el Simulador: Este botón permite al usuario iniciar un simulador que emula el comportamiento de la red de nodos. Esta funcionalidad es útil para probar configuraciones, identificar posibles problemas y validar el rendimiento de la red en un entorno controlado antes de implementarla en el mundo real. La vista de simulador se explicará en detalle en el siguiente punto de este documento, ”5.3.5. Página de Simulación (Simulator)”.

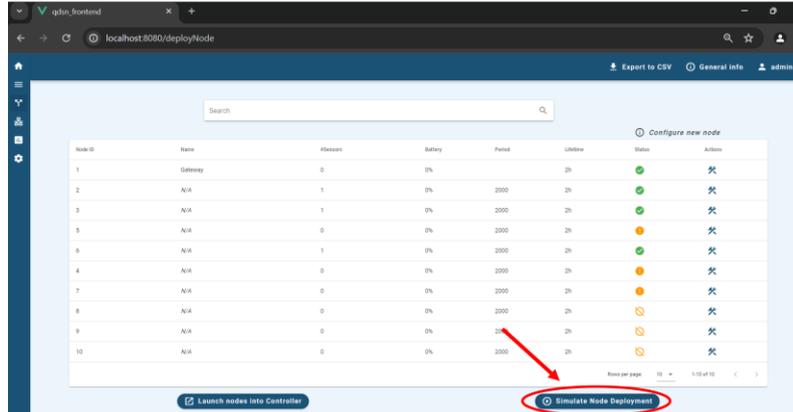


Figura 62: Botón de lanzar el simulador en la pantalla de despliegue de nodos. (Fuente: Elaboración propia)

- Lanza Despliegue de Nodos: Este botón inicia el proceso de despliegue de los nodos, configurando y activando los nodos según la configuración actual. Esta acción es esencial para poner en marcha nuevos nodos o reconfigurar nodos existentes, este botón estará habilitado solamente tras simular la configuración y comprobar que la configuración es correcta.



Figura 63: Botón de lanzar nodos al controlador habilitado. (Fuente: Elaboración propia)

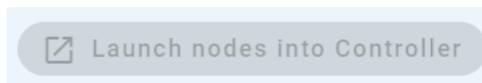


Figura 64: Botón de lanzar nodos al controlador deshabilitado. (Fuente: Elaboración propia)

Además, sobre la tabla resumen del despliegue de nodos, como puede mostrarse en la Figura 65, existe un botón informativo llamado “**Help: How to configure a node**” o “**Ayuda: Cómo configurar un nodo**” que abre una página con los pasos a seguir para configurar el nodo a modo de asistente de despliegue de nodos. Este asistente de despliegue muestra tres pasos a seguir:

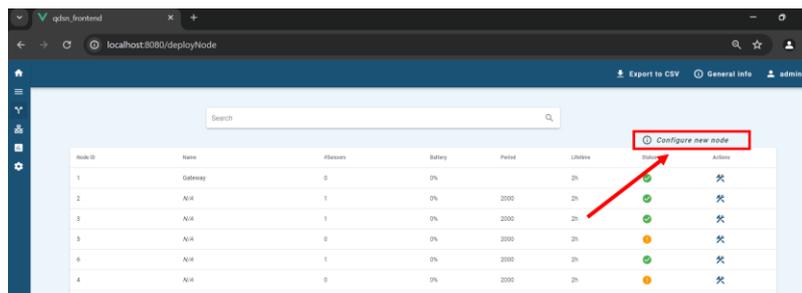


Figura 65: Botón del asistente de despliegue de nodos. (Fuente: Elaboración propia)

- **Paso 1:** Informa al usuario de que tiene que colocar el nodo en la ubicación deseada y comprobar la calidad, para eso tiene que mirar el dispositivo físico que muestra el color del led.

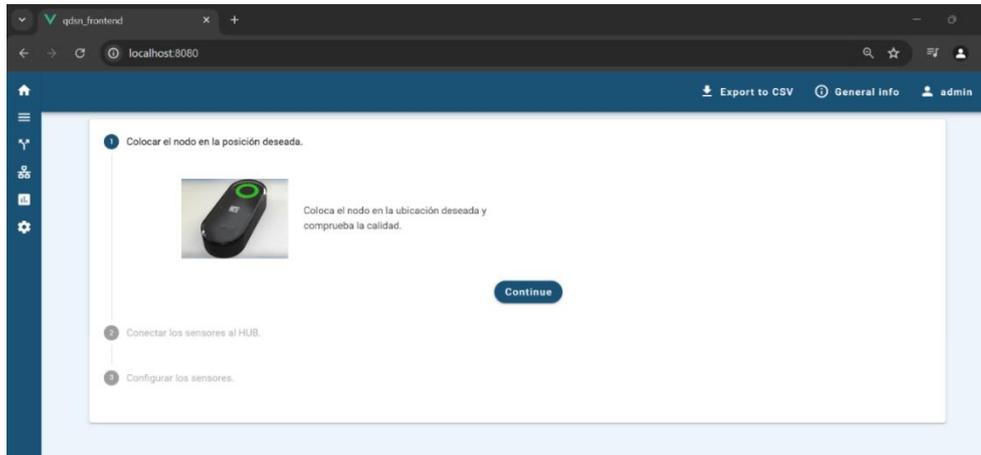


Figura 66: Paso 1 del asistente de despliegue de nodos de la aplicación. (Fuente: Elaboración propia)

- **Paso 2:** Informa al usuario para que conecte los sensores al HUB, es decir, al dispositivo que actúa como un punto central de conexión para otros nodos o sensores de la red. La función principal del HUB es recibir datos de los sensores conectados, agregarlos y transmitirlos a un nodo base o una estación central donde se procesan y analizan. Los hubs en una WSN suelen tener las siguientes características y funciones:
 - **Agregación de Datos:** Recopilan datos de múltiples sensores y pueden realizar un procesamiento preliminar para reducir la cantidad de datos que se deben enviar a la estación base. Esto ayuda a conservar energía y ancho de banda.
 - **Relé de Comunicación:** Actúan como intermediarios para retransmitir datos de sensores que están fuera del alcance directo de la estación base, extendiendo así la cobertura de la red.
 - **Gestión de Energía:** Pueden gestionar el uso de energía de la red optimizando las rutas de comunicación y reduciendo la cantidad de transmisión de datos innecesarios.
 - **Coordinación de Red:** En algunas configuraciones, los hubs pueden coordinar las actividades de los sensores, como la sincronización de tiempos para la transmisión de datos y la activación o desactivación de sensores.
 - **Escalabilidad:** Facilitan la escalabilidad de la red, permitiendo la adición de nuevos sensores sin necesidad de reconfigurar toda la red.

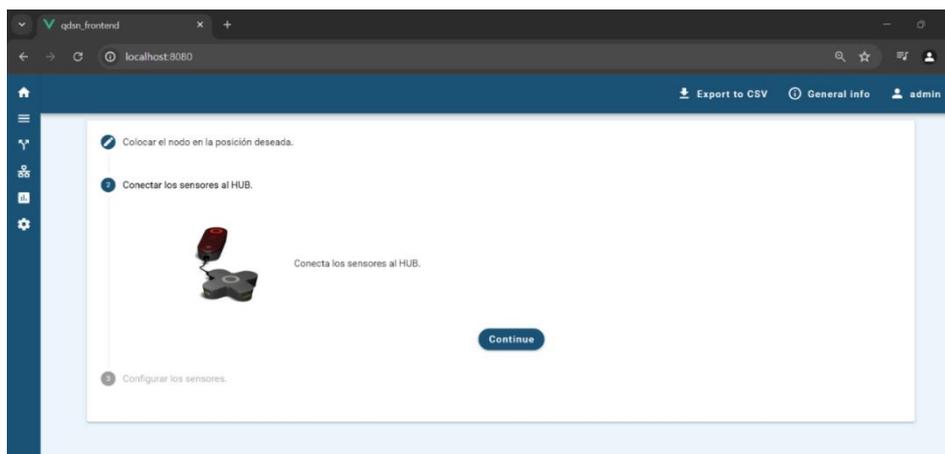


Figura 67: Paso 2 del asistente de despliegue de nodos de la aplicación. (Fuente: Elaboración propia)

- **Paso 3:** Este paso indica al usuario como configurar los sensores desde la aplicación, tal y como se ha explicado en el apartado anterior de este documento 6.4.2. Configuración del nodo.

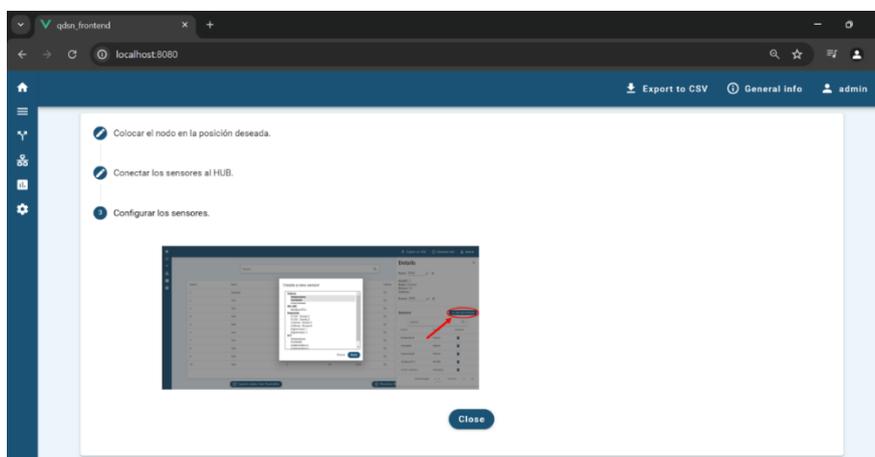


Figura 68: Paso 3 del asistente de despliegue de nodos de la aplicación. (Fuente: Elaboración propia)

6.5. Pantalla del simulador (Simulator)

La pestaña del "Simulador" actúa como una extensión avanzada de la funcionalidad ofrecida en la página de "Despliegue de Nodos", proporcionando una vista detallada y comparativa del estado actual de la red de sensores WSN y su configuración proyectada tras aplicar cambios al controlador. Al hacer clic en el botón "Lanzar el Simulador", se activa esta página.

En la tabla principal, los datos se presentan en un formato dual que facilita la comparación. Cada fila muestra la información actual del nodo en texto negro: ID del nodo, nombre, número de sensores configurados, nivel de batería, periodo de transmisión de datos, tiempo de vida estimado y estado. Separados por una barra vertical, los mismos campos aparecen en texto naranja, mostrando los valores proyectados tras aplicar la nueva configuración. Este diseño permite a los usuarios identificar rápidamente las diferencias y evaluar el impacto de los cambios, ya que parámetros como la batería, el número de sensores, el tiempo de vida y el periodo de transmisión pueden cambiar.

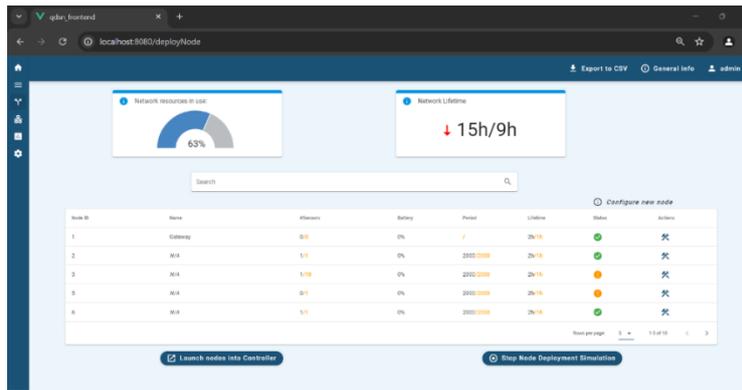


Figura 69: Pantalla de simulación de la aplicación en modo informativo. (Fuente: Elaboración propia)

En la parte superior de la página del simulador, dos tarjetas informativas adicionales ofrecen una visión global del impacto de los cambios propuestos. La tarjeta de la izquierda varía su color dependiendo del estado del sistema: azul en modo informativo, naranja en modo de advertencia y rojo en modo de error. Esta tarjeta muestra una vista general del estado proyectado del sistema tras aplicar la configuración, proporcionando una instantánea rápida de cómo quedará configurada la red de sensores WSN. La tarjeta de la derecha se centra en el tiempo de vida general del sistema, comparando el estado actual con el proyectado. Esta comparación permite a los usuarios evaluar si los ajustes mejorarán o reducirán la eficiencia y duración operativa de la red.

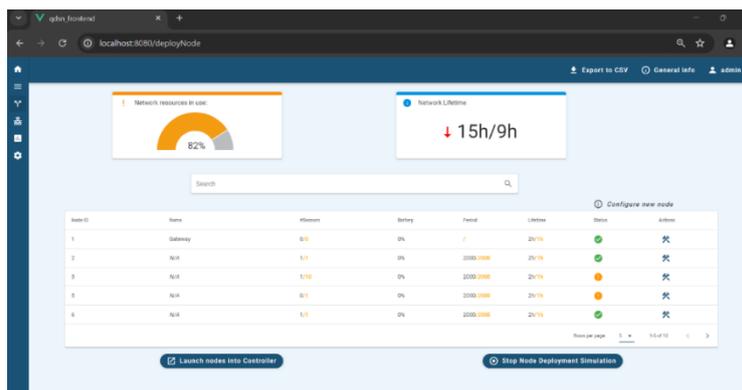


Figura 70: Pantalla de simulación de la aplicación en modo warning. (Fuente: Elaboración propia)

Una característica crítica de esta página es la interactividad y las restricciones basadas en el estado del sistema. Si el sistema no permite la configuración, el botón "Lanzar Configuración" aparecerá deshabilitado, previniendo cualquier acción hasta que se resuelvan los problemas subyacentes. Esto asegura que los usuarios no puedan aplicar cambios que podrían llevar a un mal funcionamiento o a una configuración inadecuada de la red.

La integración de estas funcionalidades en la pestaña del simulador está diseñada para ofrecer una experiencia de usuario fluida y efectiva. La visualización clara y contrastada de los datos actuales y proyectados facilita la toma de decisiones informadas y estratégicas. Además, la variabilidad del color en la tarjeta de la izquierda proporciona un indicativo visual inmediato del estado del sistema, mejorando la capacidad del usuario para reaccionar y ajustar configuraciones de manera proactiva.

Esta capacidad de simulación avanzada no solo mejora la usabilidad y la eficacia en la gestión de la red, sino que también actúa como una herramienta poderosa para la optimización continua y proactiva de la configuración de los nodos y sensores en el sistema WSN. La página del simulador asegura que cualquier cambio realizado contribuirá positivamente al rendimiento y la sostenibilidad de la red, proporcionando una plataforma robusta para la planificación y validación de configuraciones antes de su implementación real.

6.6. Pantalla de estadísticas de la red (Network)

En la página de "Network" se presenta una vista general del estado de la red mediante una serie de elementos visuales organizados para facilitar la comprensión rápida de los datos clave.

En la parte superior de la página, se presentan cuatro cuadros rectangulares con información clave sobre el estado de la red. Es decir, estos cuadros proporcionan una visión instantánea de métricas cruciales sobre el rendimiento y la salud de la red.

- **PDR (Packet Delivery Rate)**: Este cuadro muestra la tasa de entrega de tramas, que es un indicador del porcentaje de tramas o paquetes de datos que han sido correctamente recibidos y procesados en comparación con los enviados. Un valor alto indica una red eficiente y confiable.
- **Paquetes Recibidos**: Este indicador refleja el número total de paquetes de datos que han sido recibidos por el sistema desde los nodos. Este número puede ayudar a evaluar el volumen de tráfico de datos en la red y la capacidad de los nodos para comunicarse de manera efectiva.
- **Nodos Expirados**: Este cuadro indica el número de nodos que han perdido la conexión o que ya no están activos en la red. Un aumento en este número puede señalar problemas de conectividad o fallos en los nodos.
- **Historial Reciente**: Este indicador muestra el número de eventos recientes registrados en la red, como errores o alertas. Un valor en esta sección puede ayudar a identificar y diagnosticar problemas recientes o cambios en el estado de la red.

Debajo de los cuadros, se presenta una tabla titulada "Resumen de Nodos", como puede observarse en la Figura 71, esta tabla incluye las siguientes columnas para proporcionar una visión detallada de cada nodo:

- **Estado**: Indicador de estado con colores como en la pestaña de Node Deployment, representando la condición operativa de los nodos.
- **ID del nodo**: Designación o etiqueta del nodo que facilita la identificación.
- **Batería**: Porcentaje de batería restante del nodo.
- **Calidad**: Calidad del enlace del nodo.
- **Última vez visto**: Intervalo de tiempo desde la última actualización de datos del nodo.
- **Nodo Padre**: ID del nodo padre del nodo.

Esta disposición permite a los usuarios monitorear el estado y rendimiento de cada nodo en la red, facilitando la detección de problemas o áreas que requieren atención. Además, desde esta pestaña se puede acceder a la topología de la red WSN (Figura 73), clicando en el botón como se muestra en la Figura 72.

La página "Network" está diseñada para ofrecer una combinación de datos resumidos y detallados, permitiendo a los administradores y usuarios tener una visión clara y concisa del estado de la red de sensores.

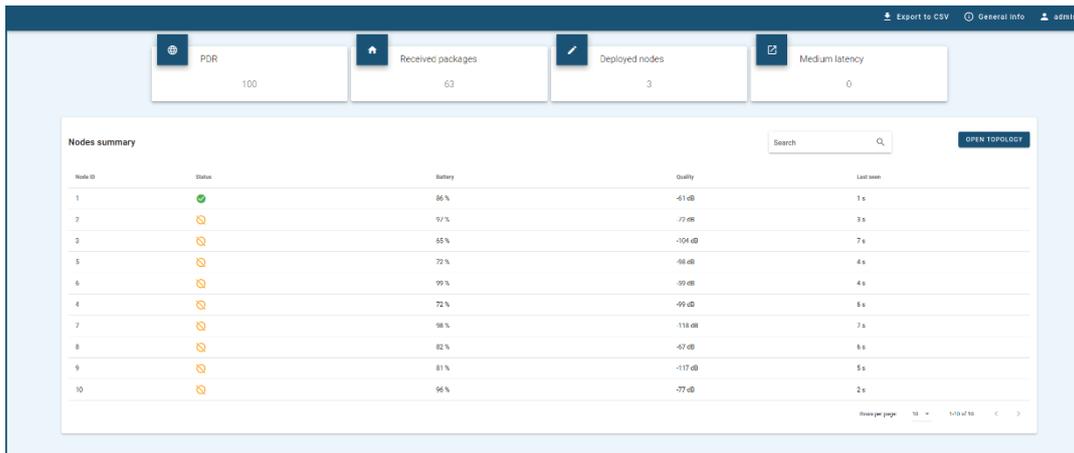


Figura 71: Pantalla de Network de la aplicación. (Fuente: Elaboración propia)

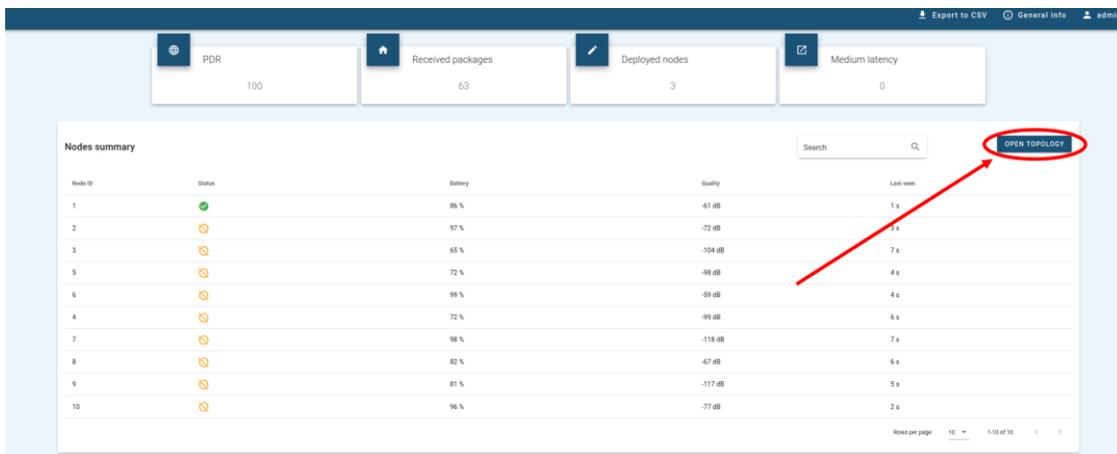


Figura 72: Botón para abrir la Topología en la pantalla de Network. (Fuente: Elaboración propia)

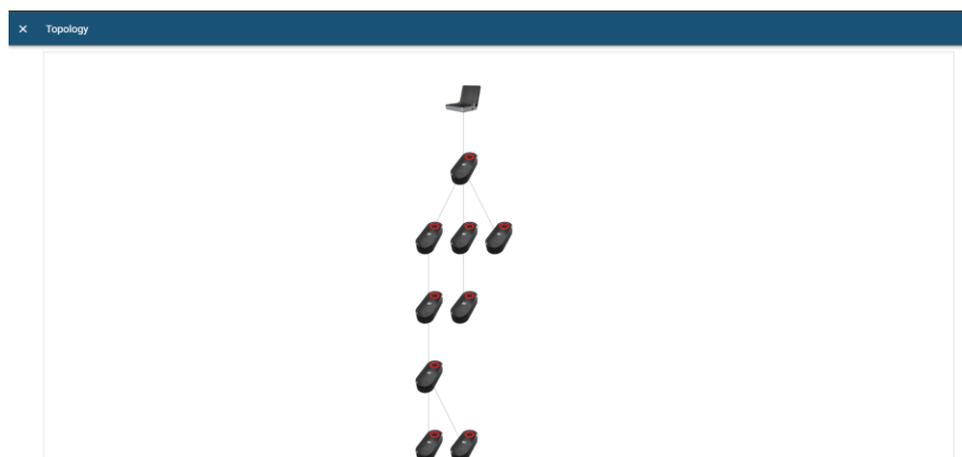


Figura 73: Gráfico de topología en la pantalla de Network. (Fuente: Elaboración propia)

6.7. Pantalla de datos en tiempo real (Livestream)

La pantalla de "Datos en Vivo" o "Livestream" está diseñada para ofrecer una experiencia integral de monitorización en tiempo real de los datos enviados por los sensores configurados en la red de nodos WSN. Esta página es altamente interactiva y permite a los usuarios personalizar la visualización de datos según sus necesidades específicas.

En la parte superior de la página, los usuarios pueden seleccionar uno o varios sensores de los nodos configurados. Esta selección se realiza mediante un selector de sensores como se muestra en la Figura 74.

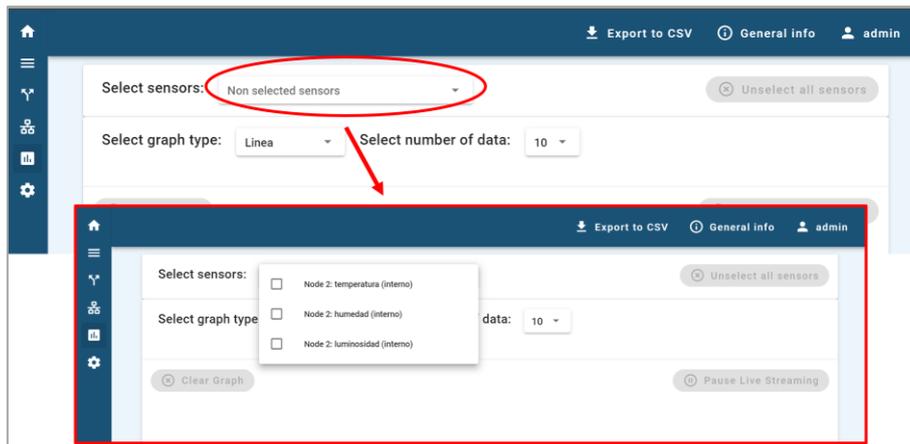


Figura 74: Selección de sensores en la página de datos en vivo. (Fuente: Elaboración propia)

Una vez que se elige un sensor, o varios, estos aparecen debajo del selector en forma de "chips" o etiquetas visuales que representan cada sensor seleccionado, como se puede observar en la Figura 75. Cada chip tiene un botón de eliminación, permitiendo al usuario remover sensores individualmente. Además, hay un botón que permite eliminar todos los sensores seleccionados de una vez, proporcionando una manera rápida de reiniciar la selección.

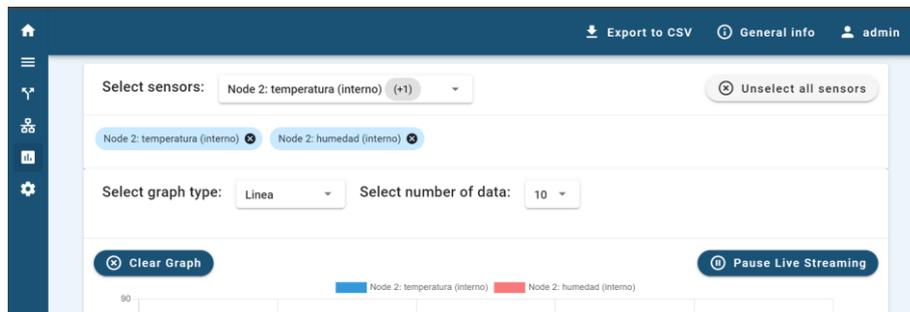


Figura 75: Sensores seleccionados en la página de datos en vivo. (Fuente: Elaboración propia)

Debajo de la lista de sensores seleccionados, los usuarios pueden elegir el tipo de gráfico en el que quieren representar los datos: línea, barras o tarta, como pueden observarse en la Figura 77, Figura 78 y Figura 79, respectivamente. También pueden especificar el número de datos que desean visualizar, ajustando así la cantidad de información mostrada en el gráfico según sus necesidades.

Una vez seleccionados los sensores y el tipo de gráfico, el gráfico correspondiente aparece en la parte inferior de la página. Este gráfico es interactivo, ofreciendo botones para pausar la representación en vivo o para limpiar el gráfico. Al comienzo, estos botones están deshabilitados, ya que no existen sensores seleccionados y, por lo tanto, no hay datos para representar. El gráfico permanecerá oculto hasta que el usuario seleccione al menos un sensor, asegurando que la interfaz no muestre datos vacíos o irrelevantes.

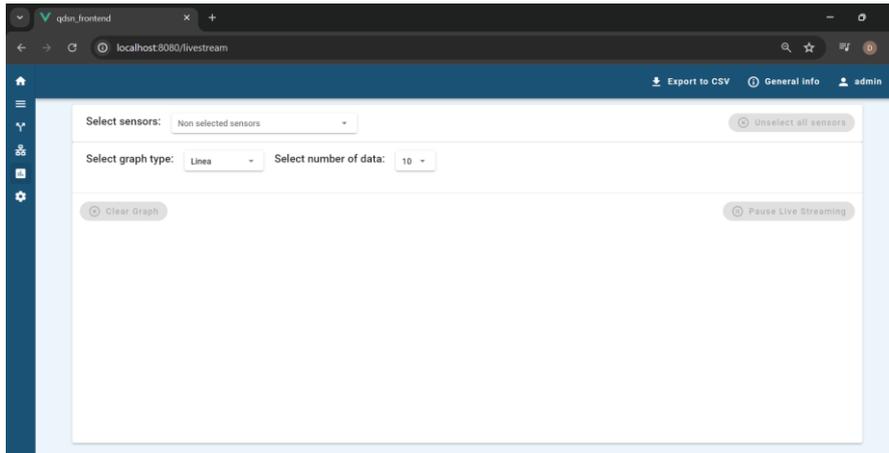


Figura 76: Página de datos en vivo sin seleccionar sensores. (Fuente: Elaboración propia)

La pantalla de "Datos en Vivo" combina estas funcionalidades de manera fluida y eficiente. La capacidad de seleccionar múltiples sensores y verlos representados en un gráfico interactivo permite a los usuarios monitorear y analizar datos en tiempo real de forma precisa. La opción de eliminar sensores seleccionados individualmente o todos a la vez proporciona flexibilidad y control sobre la visualización de datos.

Esta integración cuidadosa de funcionalidades asegura que la página de "Datos en Vivo" no solo sea fácil de usar, sino también poderosa y adaptable a las necesidades del usuario. La presentación clara y la capacidad de personalizar la visualización de datos mejoran significativamente la usabilidad y la eficacia en la gestión de la red de sensores WSN. Con esta herramienta, los usuarios pueden tomar decisiones informadas basadas en datos actualizados y precisos, optimizando así el rendimiento y la sostenibilidad de la red.

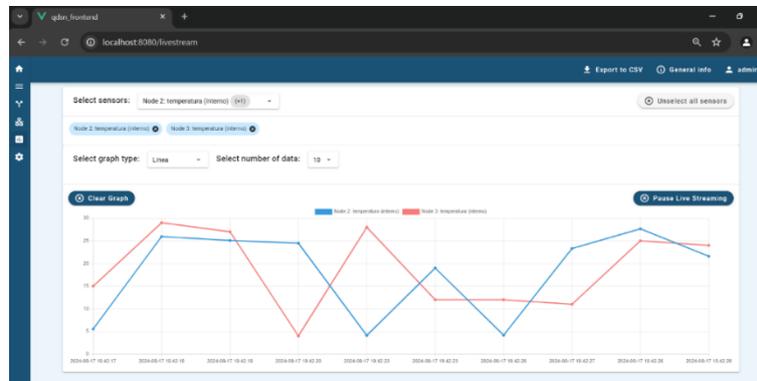


Figura 77: Gráfico de Línea en la pantalla de datos en vivo. (Fuente: Elaboración propia)

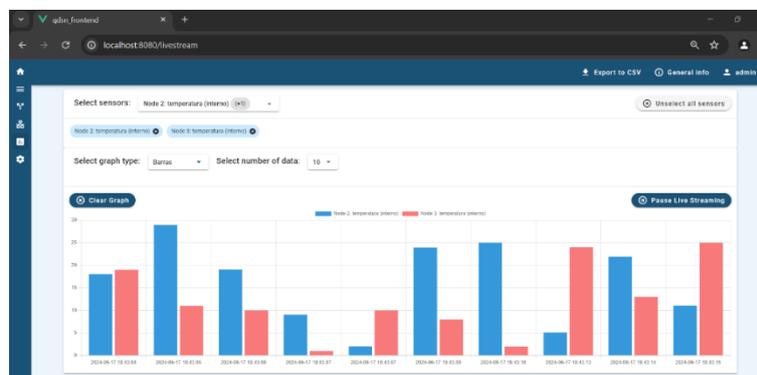


Figura 78: Gráfico de Barras en la pantalla de datos en vivo. (Fuente: Elaboración propia)

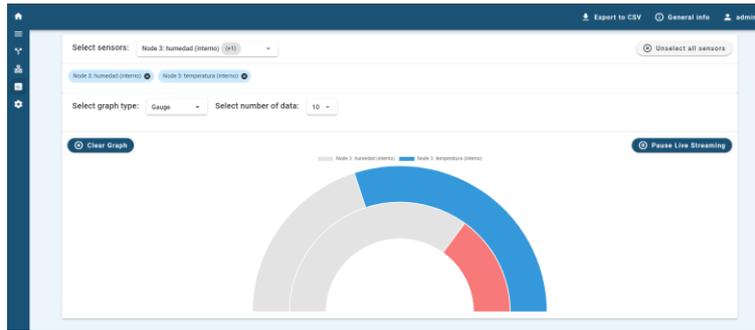


Figura 79: Gráfico de Gauge en la pantalla de datos en vivo. (Fuente: Elaboración propia)

6.8. Pantalla de configuración del sistema (Configuration)

La pantalla de "Configuración" es una sección crucial de la aplicación, diseñada para permitir a los usuarios ajustar y personalizar diversos aspectos de la red de sensores WSN y su integración con otros servicios. Esta página está organizada en dos paneles desplegados: uno dedicado a la "Configuración Global" y otro enfocado en la "Integración de Servicios".

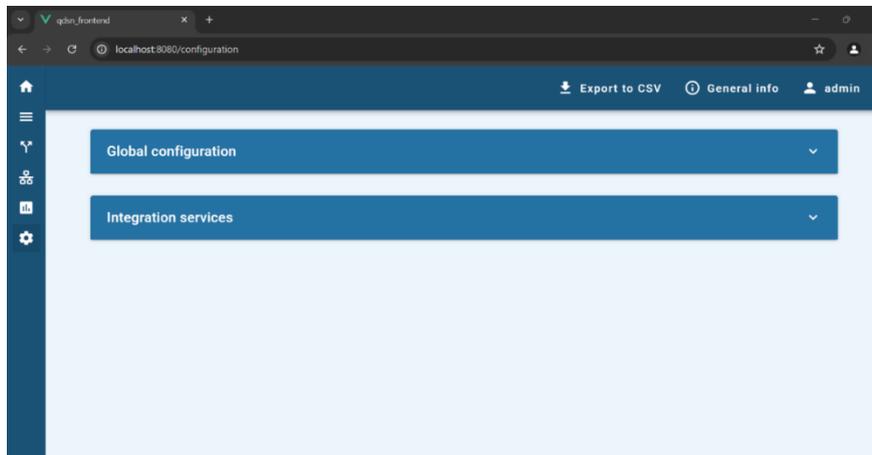


Figura 80: Pantalla de configuración con los desplegables plegados. (Fuente: Elaboración propia)

En el panel de "Configuración Global", los usuarios pueden acceder a tres menús deslizantes. El primero, "Network Configuration", permite ajustar parámetros esenciales de la conectividad de red, asegurando una operación fiable y eficiente de la red de sensores. Aquí, los usuarios pueden configurar detalles como IP, subred y Gateway, como puede observarse en la Figura 81.

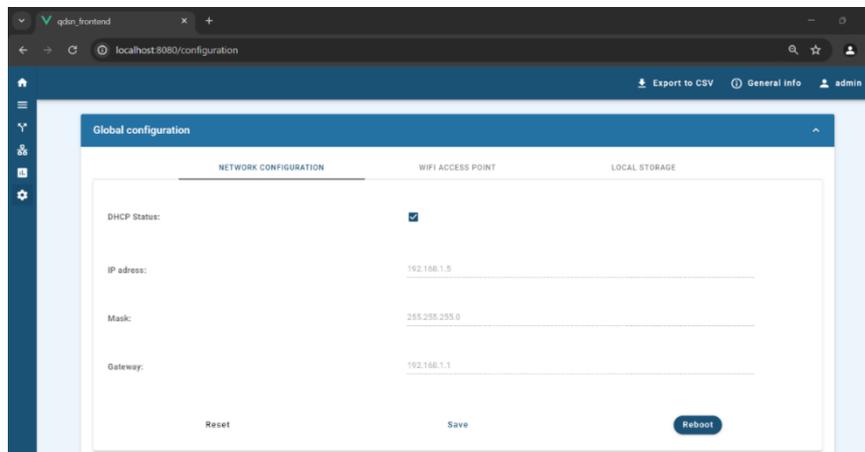


Figura 81: Configuración de red en la pantalla de configuración. (Fuente: Elaboración propia)

El segundo menú, "WiFi Access Point", ofrece opciones para gestionar la conectividad inalámbrica, incluyendo el SSID y la dirección IP del punto de acceso. Como puede observarse en la Figura 82.

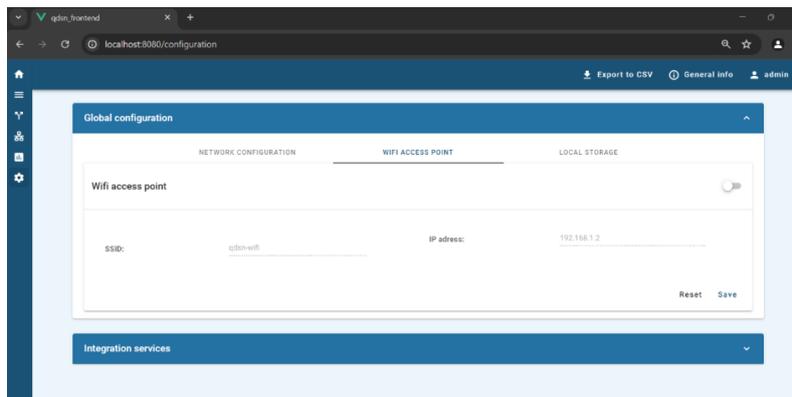


Figura 82: Configuración del punto de acceso Wi-Fi en la pantalla de configuración. (Fuente: Elaboración propia)

Por último, el menú de "Local Storage" permite definir la capacidad de almacenamiento, por ahora solo permite la capacidad de habilitar o deshabilitar el almacenaje interno para poder exportar datos en CSV como se ha visto en este documento. En un futuro se podría considerar gestionar archivos y configurar copias de seguridad, garantizando la seguridad y eficiencia del almacenamiento de datos críticos.

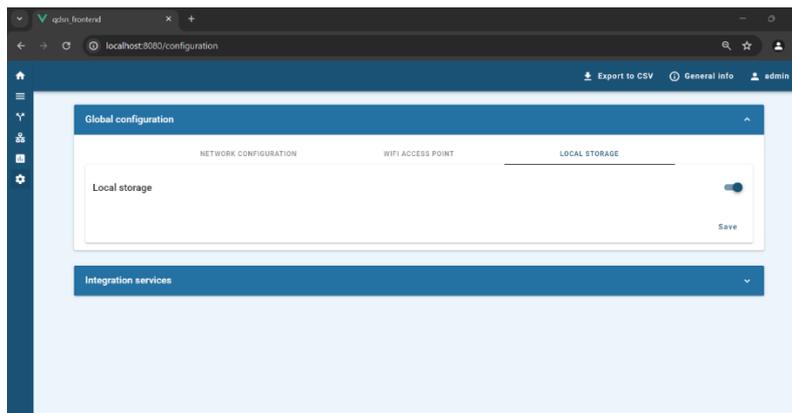


Figura 83: Almacenamiento local en pantalla de configuración. (Fuente: Elaboración propia)

El panel de "Integración de Servicios" facilita la conexión de la red de sensores con diversas plataformas y protocolos de integración. Dentro de este panel, los usuarios pueden acceder a tres menús deslizantes específicos. El menú "MQTT Client" permite habilitar o deshabilitar el cliente MQTT⁶⁶, así como configurar el broker address, el puerto, el client ID, el topic, el usuario y la contraseña, estableciendo una comunicación segura con servicios MQTT. Por defecto, el DHCP Status aparece activo, lo que hace que no se puedan cambiar los campos de la tarjeta hasta que sea desactivado. Si se realizan cambios y estos se guardan, no se actualizarán hasta que no se reinicie el sistema, por ello, al guardar los nuevos cambios aparecerá el botón 'Reboot' que reiniciará el sistema para actualizar los cambios guardados, hasta que no se reinicie no se harán efectivos los cambios.

⁶⁶ MQTT (Message Queuing Telemetry Transport) es un protocolo ligero de mensajería basado en el modelo de publicación/suscripción. Es ideal para aplicaciones con recursos limitados y redes de ancho de banda reducido. MQTT se utiliza comúnmente en la comunicación máquina a máquina (M2M) y el Internet de las Cosas (IoT) debido a su eficiencia y bajo consumo de energía. Funciona mediante un broker que gestiona los mensajes entre clientes, permitiendo una comunicación asíncrona y flexible.

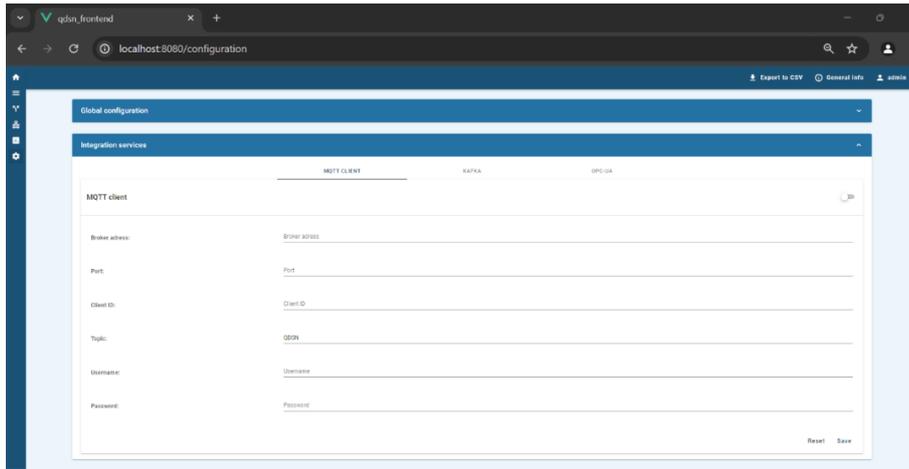


Figura 84: Configuración de cliente MQTT en pantalla de configuración. (Fuente: Elaboración propia)

El menú "Kafka" ofrece opciones similares, permitiendo habilitar o deshabilitar la integración con Apache Kafka⁶⁷, y configurar el servidor, el puerto y el topic, asegurando el procesamiento de datos en tiempo real.

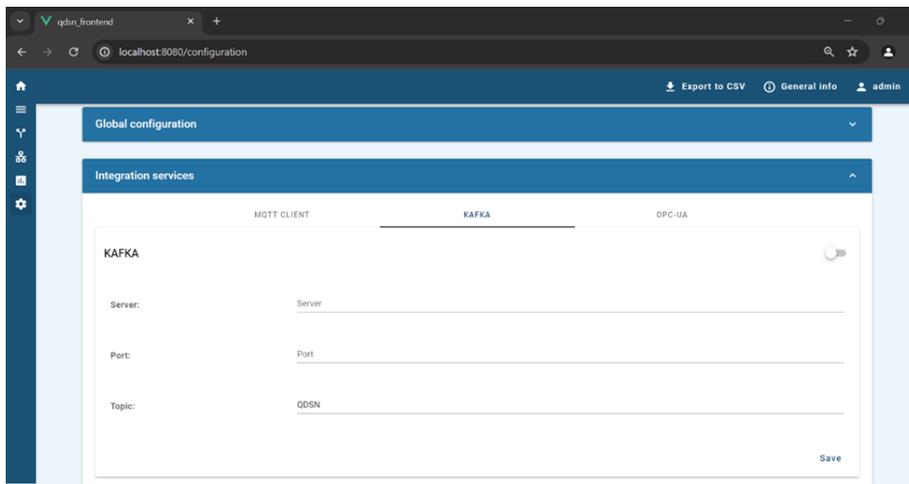


Figura 85: Kafka en pantalla de configuración. (Fuente: Elaboración propia)

El menú "OPC-UA", aunque incluye un botón para habilitar o deshabilitar el protocolo OPC-UA⁶⁸, actualmente está inutilizable y no está accesible para la configuración.

⁶⁷ Kafka es una plataforma de streaming de datos distribuida diseñada para manejar flujos de datos en tiempo real con alta capacidad y baja latencia. Desarrollada por LinkedIn y posteriormente donada a la Apache Software Foundation, Kafka se utiliza principalmente para construir pipelines de datos en tiempo real y aplicaciones de streaming. Organiza los datos en topics, a los que los productores publican mensajes y los consumidores suscriben para recibirlos. Kafka es conocido por su escalabilidad, durabilidad y capacidad de manejar grandes volúmenes de datos de manera eficiente.

⁶⁸ OPC-UA (Open Platform Communications Unified Architecture) es un protocolo de comunicación industrial diseñado para facilitar el intercambio seguro y confiable de datos en sistemas de automatización y control. Es la evolución del estándar OPC clásico y ofrece interoperabilidad entre dispositivos y aplicaciones de diferentes fabricantes. OPC-UA es independiente de la plataforma, soporta múltiples modelos de datos y proporciona características avanzadas como seguridad mejorada, modelado de información y escalabilidad, lo que lo hace adecuado para aplicaciones industriales complejas y el Internet de las Cosas (IoT).

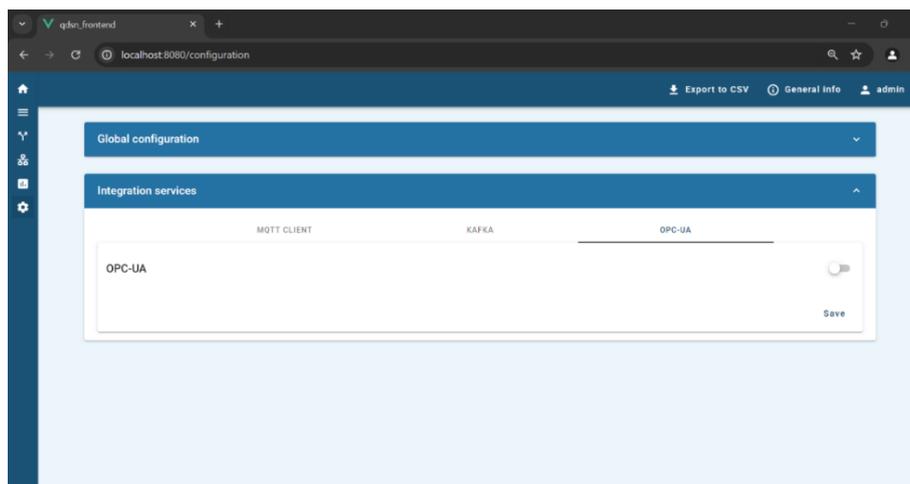


Figura 86: OPC-UA en pantalla de configuración. (Fuente: Elaboración propia)

La experiencia de usuario en la pantalla de "Configuración" está diseñada para ser intuitiva y eficiente. Los paneles desplegados permiten expandir y contraer secciones según las necesidades del usuario, proporcionando un acceso rápido a las configuraciones relevantes sin sobrecargar la interfaz. Esta organización permite a los usuarios encontrar y ajustar configuraciones específicas de manera sencilla y directa.

Cada menú deslizante está cuidadosamente diseñado para ser fácil de usar, con opciones y campos claramente etiquetados que facilitan la comprensión y modificación de los parámetros configurables. La interfaz proporciona una navegación fluida y una experiencia de usuario optimizada, asegurando que los ajustes necesarios se realicen de manera rápida y sin complicaciones.

En resumen, la pantalla de "Configuración" ofrece a los usuarios un control completo sobre los aspectos críticos de la red de sensores WSN y su integración con servicios externos. La estructura de paneles desplegados y menús deslizantes proporciona una experiencia de configuración robusta, flexible y accesible, mejorando la eficiencia operativa y la capacidad de respuesta del sistema. La capacidad de habilitar o deshabilitar servicios específicos como MQTT y Kafka, así como la futura accesibilidad del OPC-UA, aseguran que la aplicación se mantenga adaptable y preparada para futuras expansiones y ajustes.

7. Conclusiones

En el desarrollo de la plataforma web para la gestión y planificación de redes de sensores inalámbricos definidas por software (SDWSN), se han alcanzado conclusiones significativas que abarcan tanto los aspectos técnicos como las decisiones de diseño y el impacto potencial de la solución.

Desde el punto de vista conceptual, el análisis del marco teórico sobre redes de sensores inalámbricos (WSN) y tecnologías de comunicación ha permitido comprender la relevancia de estas redes en una amplia gama de aplicaciones, desde el monitoreo ambiental hasta la agricultura de precisión y la gestión urbana. Las WSN ofrecen una infraestructura crítica para la recopilación de datos en tiempo real y la automatización de procesos, aunque presentan desafíos relacionados con la gestión descentralizada y la complejidad en su configuración. La integración de Redes Definidas por Software (SDN), que permite un control centralizado y flexible de las redes, ha sido una solución clave para superar estas limitaciones, y el proyecto ha abordado este desafío desarrollando una plataforma que simplifica la gestión de redes SDWSN, haciéndola accesible a usuarios no especializados.

Otro punto destacado es la decisión de que la plataforma se ejecute en contenedores Docker. Esta elección responde a varias razones estratégicas. Docker permite crear un entorno de ejecución aislado y reproducible, facilitando el despliegue de la aplicación en diferentes entornos, desde servidores locales hasta nubes públicas o privadas, sin problemas de compatibilidad. Además, Docker mejora la escalabilidad y la portabilidad, lo que es esencial para una plataforma que pretende ser accesible desde cualquier dispositivo. Esta solución ofrece la posibilidad de crear instancias múltiples de la aplicación con facilidad, lo que resulta especialmente útil cuando se manejan simulaciones y operaciones simultáneas en entornos distribuidos. Asimismo, Docker reduce la dependencia de configuraciones específicas de hardware o software, minimizando los problemas de implementación y mantenimiento.

En cuanto a las tecnologías utilizadas para el desarrollo del frontend, se ha decidido utilizar Vue.js en lugar de alternativas como React.js o Angular. Vue.js ofrece una curva de aprendizaje más suave en comparación con Angular, lo que permite un desarrollo más rápido y eficiente, ideal para proyectos con recursos limitados o plazos ajustados. Vue.js también destaca por su ligereza y simplicidad, manteniendo al mismo tiempo un excelente rendimiento, lo que es crucial para una aplicación orientada a la visualización en tiempo real y la gestión de múltiples vistas interactivas. Además, Vue.js proporciona una gran flexibilidad sin imponer una estructura rígida al desarrollo, lo que facilita la integración de bibliotecas y módulos adicionales a medida que la plataforma evoluciona.

Por otro lado, aunque React.js también es una opción popular, Vue.js fue preferido por su mejor integración nativa con tecnologías modernas como Webpack y su enfoque orientado a componentes, lo que simplifica la modularidad y la reutilización del código. Vue.js también tiene una comunidad activa y una documentación más accesible, lo que garantiza un soporte adecuado durante el desarrollo y una facilidad para resolver problemas que puedan surgir en el futuro.

El proyecto también ofrece soluciones innovadoras al integrar un simulador de recursos radio dentro de la plataforma, lo que permite a los usuarios validar configuraciones antes de implementarlas en hardware físico. Esta funcionalidad reduce el riesgo de errores y optimiza el proceso de despliegue de redes WSN, lo que ahorra tiempo y recursos. La capacidad de simular entornos de red en un entorno seguro aporta un valor añadido significativo al proyecto, permitiendo a los usuarios ajustar y optimizar sus redes antes de la puesta en marcha real.

En términos generales, la plataforma desarrollada no solo simplifica la gestión de redes SDWSN, sino que también reduce barreras de entrada para usuarios con menos conocimientos técnicos, democratizando el uso de estas tecnologías avanzadas en sectores como la agricultura, la industria y la gestión urbana. Al ser accesible desde cualquier dispositivo a través de la web, gracias al enfoque Bring Your Own Device (BYOD), la solución facilita el monitoreo y control en tiempo real de redes distribuidas sin necesidad de infraestructura costosa o especializada.

Por último, cabe destacar que, en el estudio de mercado, se analizaron soluciones de Ackcio, VersaSense, Analog Devices, Adeunis, y el desarrollo Zigbee de NXP. La plataforma más similar a la propuesta en este proyecto es la de VersaSense, que ofrece una solución modular para la integración y monitoreo de redes de sensores inalámbricos, facilitando tanto su despliegue como la gestión en tiempo real. Las demás compañías, como Ackcio, Analog Devices, Adeunis, y NXP, se enfocan principalmente en el desarrollo de hardware y la implementación de tecnologías de comunicación para redes WSN, pero no ofrecen plataformas de software integrales que asistan en la configuración y monitoreo de las redes. Esto resalta la necesidad de una herramienta accesible y completa para el despliegue y gestión centralizada de redes SDWSN, como la propuesta en este proyecto.

En conclusión, el proyecto combina una sólida base teórica, decisiones tecnológicas estratégicas y una orientación práctica hacia la usabilidad y escalabilidad, ofreciendo una herramienta versátil para la gestión y planificación de redes WSN. El uso de Vue.js para el frontend y Docker para el despliegue de la aplicación asegura una plataforma eficiente, escalable y accesible, capaz de adaptarse a una amplia variedad de escenarios y necesidades tecnológicas.

Bibliografía

- [1] A. A. H. M. Shuker Mahmoud, «A Study of Efficient Power Consumption Wireless Communication Techniques/Modules for Internet of Things (IoT) Applications», en *Advances in Internet of Things*, vol. 6, no. 2, 2016.
- [2] IEEE Computer Society, *IEEE Standard for Information Technology, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LRWPANs)*, 2015.
- [3] K.-H. Chang and B. Mason, «The IEEE 802.15.4g Standard for Smart Metering Utility Networks», en *IEEE Smart Grid Communications (SmartGridComm)*, 2012.
- [4] Y. Wang, Y. Zhang, J. Liu, y R. Bhandari, «Coverage, Connectivity, and Deployment in Wireless Sensor Networks», en *Recent Developments in Wireless Sensor Ad-hoc Networks*, pp. 25-44, 2015.
- [5] M. R. Senouci y A. Mellouk, *Deploying Wireless Sensor Networks: Theory and Practice*, ISTE Press, 2016.
- [6] W.-H. Liao, Y. Kao, y Y.-S. Li, «A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks», *Expert Systems with Applications*, vol. 38, no. 10, pp. 12180-12188, 2011.
- [7] W.-H. Liao, Y. Kao, y R.-T. Wu, «Ant colony optimization-based sensor deployment protocol for wireless sensor networks», *Expert Systems with Applications*, vol. 38, no. 6, pp. 6599-6605, 2011.
- [8] Y. Yoon y Y. H. Kim, «An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks», *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473-1483, 2013.
- [9] A. Rakavi, M. Manikandan, y K. Hariharan, «Grid Based Mobile Sensor Node Deployment for Improving Area Coverage in Wireless Sensor Networks», en *3rd International Conference on Signal Processing, Communication and Networking*, 2015.
- [10] S. Mondal, S. Ghosh, y U. Biswas, «ACOH: Ant Colony Optimization based Hierarchical Clustering in Wireless Sensor Network», en *International Conference on Emerging Technological Trends (ICETT)*, 2016.
- [11] S. Mini, S. K. Udgata, y S. L. Sabat, «Sensor deployment and scheduling for target coverage problem in wireless sensor networks», *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636-644, 2014.
- [12] G. Han, L. Liu, J. Jiang, y otros, «Analysis of Energy-Efficient Connected Target Coverage Algorithms for Industrial Wireless Sensor Networks», *IEEE Transactions on Industrial Informatics*, 2017.
- [13] M. Sen, I. Banerjee, M. Chatterjee, y T. Samanta, «A Node Deployment Mechanism Accounting into Received Signal Strength and Frequency Diversity for a Wireless Sensor Network», en *IEEE Sensors*, pp. 1-3, 2016.
- [14] S. Sengupta, S. Das, M. D. Nasir, y B. K. Panigrahi, «Multi-objective node deployment in WSNs: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity», *Engineering Applications of Artificial Intelligence*, vol. 26, 2013.

- [15] H. M. Ammari y S. Das, «A study of k-coverage and measures of connectivity in 3D wireless sensor networks», *IEEE Transactions on Computers*, vol. 59, no. 2, 2010.
- [16] T. A. C. De Melo, F. D. M. De Oliveira, R. S. Semente, X. C. Benjamim, y A. O. Salazar, «WINSS: A Simulation Tool of the IEEE 802.15.4 Standard for Network Simulator 2», en *International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*, 2016.
- [17] B. Musznicki y P. Zwierzykowski, «Survey of Simulators for Wireless Sensor Networks», *International Journal of Grid and Distributed Computing*, vol. 5, no. 3, 2012.
- [18] R. Sasnauskas, O. Landsiedel, M. H. Alizai, C. Weise, S. Kowalewski, y K. Wehrle, «KleeNet: Discovering Insidious Interaction Bugs in Wireless Sensor Networks Before Deployment», en *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010.
- [19] Honeywell, «RF-Lite Tool», [En línea]. Disponible: <https://prod-edam.honeywell.com/content/dam/honeywell-edam/pmt/hps/products/pas/wireless/OneWireless-RF-Lite-Planning-Tool.pdf>. [Fecha de acceso: Sep. 21, 2024].
- [20] M. A. Hany y A. T. Abdel-Hamid, «ZEIN: An indoor Zigbee planning tool», en *International Conference on Innovations in Information Technology (IIT)*, 2015.
- [21] R. Tynan, A. Ruzzelli, y G. M. P. O'Hare, «A Methodology for the Deployment of Multi-Agent Systems on Wireless Sensor Networks», en *Proceedings of the 2nd KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications*, 2008.
- [22] S. M. Chad, T. Camp, M. Colagrosso, y O. Chase, «TestbedProfiler: A validation tool for wireless sensor network testbed deployment», en *IEEE 35th Conference on Local Computer Networks (LCN)*, 2010.
- [23] R. C. Luo y O. Chen, «Mobile Sensor Node Deployment and Asynchronous Power Management for Wireless Sensor Networks», *IEEE Transactions on Industrial Electronics*, vol. 59, no. 5, 2012.
- [24] M. Amarlingam, I. Adithyan, P. Rajalakshmi, y otros, «Deployment Adviser tool for Wireless Sensor Networks», en *IEEE World Forum on Internet of Things (WF-IoT)*, 2014.
- [25] G. Mujica, A. Garcia, J. Gordillo, J. Portilla, y T. Riesgo, «A Novel On-site Deployment, Commissioning and Debugging Technique to Assess and Validate WSN Based Smart Systems», en *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015.
- [26] Phoenix Contact, «WirelessHART Network Commissioning Tool», [En línea]. Disponible: <https://www.phoenixcontact.com/en-pc/products/industrial-communication/industrial-wireless>. [Fecha de acceso: Sep. 21, 2024].
- [27] A. Dinh-Duc, T. Dang-Ha, y N. Lam, «NViz - A General Purpose Visualization Tool for Wireless Sensor Networks», en *IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2012.

- [28] Z. Zhao, W. Huangfu, y L. Sun, «NSSN: A network monitoring and packet sniffing tool for wireless sensor networks», en *IWCMC 2012 - 8th International Wireless Communications and Mobile Computing Conference*, 2012.
- [29] M. Santos, S. Tennina, M. Alves, M. Bourouche, V. Cahill, D. Almeida, G. Carrozza, A. Hill, y G. Chasapis, «EMMON - EMbedded MONitoring», en *Embedded World Conference*, 2012.
- [30] P. Sommer y B. Kusy, «Minerva: Distributed Tracing and Debugging in Wireless Sensor Networks», en *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013.
- [31] F. P. Garcia, R. M. C. Andrade, C. T. Oliveira, y J. Neuman de Souza, «EPMOST: An Energy-Efficient Passive Monitoring System for Wireless Sensor Networks», *Sensors*, vol. 14, no. 6, 2014.
- [32] A. Khelil y R. Beghdad, «ESA: An Efficient Self-deployment Algorithm for Coverage in Wireless Sensor Networks», *Procedia Computer Science*, vol. 98, 2016.
- [33] SoftDEL, «Wireless Network Monitor», [En línea]. Disponible: <https://www.softdel.com/>. [Fecha de acceso: Sep. 21, 2024].
- [34] Vue.js, "Vue.js: The Progressive JavaScript Framework," Vue.js. [En línea]. Disponible: <https://vuejs.org/>. [Fecha de acceso: Sep. 21, 2024].
- [35] S. Krause, "JS Framework Benchmark," Krause's Blog. [En línea]. Disponible: <https://krausest.github.io/js-framework-benchmark/current.html>. [Fecha de acceso: Sep. 21, 2024].
- [36] GitLab, "The DevSecOps Platform," GitLab. [En línea]. Disponible: <https://gitlab.com>. [Fecha de acceso: Sep. 21, 2024].
- [37] Docker, "Empowering App Development for Developers," Docker. [En línea]. Disponible: <https://www.docker.com/>. [Fecha de acceso: Sep. 21, 2024].
- [38] Q. Wang y J. Jiang, «Comparative examination on architecture and protocol of industrial wireless sensor networks standards», *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 2197-2219, 2016.
- [39] *WirelessHART™, EC 62591:2016 – Industrial Networks – Wireless Communication Network and Communication Profiles*, 2016.
- [40] *ISA 100.11a, IEC 62734:2014 – Industrial Networks – Wireless Communication Network and Communication Profiles*, 2014.
- [41] Thread Group, *Thread Specification*, 2017. [En línea]. Disponible: <https://www.threadgroup.org>. [Fecha de acceso: Sep. 21, 2024].
- [42] H. Harada, K. Mizutani, J. Fujiwara, K. Mochizuki, K. Obata, y R. Okumura, «IEEE 802.15.4g Based Wi-SUN Communication System», *IEICE Transactions on Communications*, vol. 100, no. 7, pp. 1032-1043, 2017.
- [43] K.-H. Chang y B. Mason, «The IEEE 802.15.4g Standard for Smart Metering Utility Networks», en *IEEE Smart Grid Communications (SmartGridComm)*, 2012.

- [44] LoRa Alliance Technical Committee, *LoRaWAN Specification*, 1.1 ed., N. Sornin, 2017. [En línea]. Disponible: <https://lora-alliance.org/lorawan-for-developers/>. [Fecha de acceso: Sep. 21, 2024].
- [45] Bluetooth Special Interest Group (SIG), *Bluetooth Core Specification v5.0*, 2016. [En línea]. Disponible: <https://www.bluetooth.com/specifications/core-specifications>. [Fecha de acceso: Sep. 21, 2024].
- [46] Bluetooth Special Interest Group (SIG), "Bluetooth 5. Go Faster. Go Further", [En línea]. Disponible: <https://www.bluetooth.com/bluetooth-5>. [Fecha de acceso: Sep. 21, 2024].
- [47] S. Yu, R. Wang, H. Xu, W. Wan, Y. Gao, y Y. Jin, «WSN nodes deployment based on artificial fish school algorithm for traffic monitoring system», en *IET International Conference on Smart and Sustainable City*, 2011.
- [48] C. Öztürk, D. Karaboga, y B. Görkemli, «Artificial bee colony algorithm for dynamic deployment of wireless sensor networks», *Turk J. Elec. Eng. & Comp. Sci.*, vol. 20, no. 2, 2012.
- [49] X. Liu y D. He, «Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks», *Journal of Network and Computer Applications*, vol. 45, pp. 91-101, 2014.
- [50] Y. Li, Y. Q. Song, Y. H. Zhu, y R. Schott, «Deploying wireless sensors for differentiated coverage and probabilistic connectivity», en *IEEE Wireless Communications and Networking Conference (WCNC)*, 2010.
- [51] N. Aitsaadi, N. Achir, K. Boussetta, y G. Pujolle, «Multi-Objective WSN Deployment: Quality of Monitoring, Connectivity and Lifetime», en *IEEE International Conference on Communications*, 2010.
- [52] A. Konstantinidis, K. Yang, Q. Zhang, y D. Zeinalipour-Yazti, «A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks», *Computer Networks*, vol. 54, no. 6, pp. 960-976, 2010.
- [53] Z. Yun, X. Bai, D. Xuan, T. H. Lai, y W. Jia, «Optimal Deployment Patterns for Full Coverage and k-Connectivity ($k \leq 6$) Wireless Sensor Networks», *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 820-833, 2010.
- [54] X. Wang y S. Wang, «Hierarchical Deployment Optimization for Wireless Sensor Networks», *IEEE Transactions on Mobile Computing*, vol. 10, no. 7, pp. 1028-1041, 2011.
- [55] Emerson, «Wireless Planning Tool», Emerson. [En línea]. Disponible: <https://www3.emersonprocess.com/ams/wirelessplanningtool/>. [Fecha de acceso: Sep. 21, 2024].
- [56] Pepperl+Fuchs, «Wireless Network Checker», Pepperl+Fuchs. [En línea]. Disponible: <https://www.pepperl-fuchs.com/global/en/index.htm>. [Fecha de acceso: Sep. 21, 2024].
- [57] Phoenix Contact, «WirelessHART Network Planner Tool», Phoenix Contact. [En línea]. Disponible: <https://www.phoenixcontact.com/en-us/>. [Fecha de acceso: Sep. 21, 2024].

- [58] Emerson, «Product Data Sheet: AMS Wireless Snap-On Application», [En línea]. Disponible: <https://www.emerson.com/documents/automation/product-data-sheet-ams-wireless-snap-on-application-ams-en-38378.pdf>. [Fecha de acceso: Sep. 21, 2024].
- [59] Emerson, «Wireless Estimator», [En línea]. Disponible: <https://www3.emersonprocess.com/rosemount/wirelessestimator/default.aspx>. [Fecha de acceso: Sep. 21, 2024].
- [60] SoftDEL, «Wireless Network Planner», [En línea]. Disponible: <https://www.softdel.com/>. [Fecha de acceso: Sep. 21, 2024].
- [61] A. Rodrigues, T. Camilo, J. S. Silva, y F. Boavida, «Diagnostic tools for wireless sensor networks: A comparative survey», *Journal of Network and Systems Management*, vol. 21, no. 3, pp. 408-452, 2013.
- [62] J. Vera Pérez, *Diseño y evaluación de mecanismos de optimización en redes de sensores inalámbricas industriales*, [Tesis doctoral], Universitat Politècnica de València, 2021. [En línea]. Disponible: <https://doi.org/10.4995/Thesis/10251/179700>.
- [63] D. Todoli Ferrandis, J. Silvestre-Blanes, S. Santonja Climent, V. M. Sempere Paya, y J. Vera-Pérez, «Deploy&Forget Wireless Sensor Networks for itinerant applications», *Computer Standards & Interfaces*, vol. 56, pp. 27-40, 2018. [En línea]. Disponible: <https://doi.org/10.1016/j.csi.2017.09.002>.
- [64] J. Vera-Pérez, D. Todoli Ferrandis, S. Santonja Climent, J. Silvestre-Blanes, y V. M. Sempere Paya, «A Joining Procedure and Synchronization for TSCH-RPL Wireless Sensor Networks», *Sensors*, vol. 18, no. 10, 2018. [En línea]. Disponible: <https://doi.org/10.3390/s18103556>.
- [65] Vuetify, "Vuetify: Material Design Component Framework," Vuetify. [En línea]. Disponible: <https://v2.vuetifyjs.com/en/>. [Fecha de acceso: Sep. 21, 2024].
- [66] Pictogrammers, "Material Design Icons Library," Pictogrammers. [En línea]. Disponible: <https://pictogrammers.com/library/mdi/>. [Fecha de acceso: Sep. 21, 2024].

Anexo I: Endpoints de la API REST de la aplicación web.

Este Anexo está dedicado a la API REST de QDSN, la aplicación web de administración de una WSN, por lo que se exponen los diferentes endpoints que forman la API, diseñada para facilitar la interacción entre el back-end y el front-end de la aplicación, y el control eficiente de la red de sensores.

A través de esta API, los desarrolladores tienen acceso a una variedad de funcionalidades que abarcan desde la configuración y gestión de dispositivos hasta la recopilación y análisis de datos sensoriales. Cada endpoint ofrece una interfaz clara para ejecutar operaciones concretas, facilitando la integración sin problemas con sistemas externos y la automatización de procesos en el ecosistema de QDSN.

En esta sección, se detallan exhaustivamente cada uno de los endpoints disponibles, destacando sus parámetros, métodos de solicitud y respuestas esperadas, para brindar una guía completa a los desarrolladores que podrán aprovechar al máximo el uso de la API REST de QDSN.

➤ Endpoint de Inicio de Sesión:

Este endpoint es llamado por el front-end cuando el usuario realiza las acciones en la aplicación para iniciar sesión.

- Método HTTP: **POST**
- Ruta (Path): **/api/v1/login**
- Cuerpo de la solicitud (request body): El cuerpo de la solicitud contiene las credenciales del usuario, es decir, un nombre de usuario y una contraseña.

```
{  
  "username": string  
  "password": string  
}
```

- Respuesta (response): La respuesta proporciona un token JWT (JSON Web Token) que se utilizará para autorizar y autenticar al usuario en las solicitudes subsiguientes a la API. Este token se puede incluir en los encabezados de las solicitudes posteriores para acceder a recursos protegidos por autenticación.

```
{ "jwt": "token" }
```

➤ Endpoint para actualizar el nombre de un nodo en la WSN:

Este endpoint se utiliza para actualizar el nombre de un nodo específico en la WSN a través de una solicitud PUT a la API REST.

- Método HTTP y ruta (Path): **PUT**
- Ruta (Path): **/api/v1/wsn/<nodeid>/name**
Este endpoint requiere un parámetro de ruta <nodeid>, que especifica el identificador único del nodo del que se desea obtener información.
- Cuerpo de la solicitud (request body): El cuerpo de la solicitud contiene el nuevo nombre que se desea asignar al nodo.

```
{ "name": string }
```

- Respuesta (response): La respuesta confirma si el nombre del nodo se ha actualizado con éxito.

➤ Endpoint para obtener el estado del gateway:

Este endpoint es utilizado para obtener el estado del Gateway de una red a través de una solicitud GET a la API REST.

- Método HTTP: **GET**
- Ruta (Path): **/api/v1/gatewaystatus**
- Cuerpo de la solicitud (request body): Este endpoint no requiere ningún parámetro de entrada adicional en la solicitud.
- Respuesta (response): La respuesta proporciona información detallada sobre el estado del gateway en diferentes interfaces de red:

```
{
  "ethernet": {
    "dhcp": "Enabled",
    "ip": "192.168.2.1",
    "state": "connected",
    "mDNS": "qdsn-16-38.local"
  },
  "wifi": {
    "ip": "192.168.1.2",
    "state": "disabled",
    "ssid": "qdsn-16-38-wifi"
  },
  "lowpan": {
    "ip": "fd00::1",
    "state": "connected",
    "conf_nodes": 5,
    "unconf_nodes": 5,
    "sensors": 8
  }
}
```

→ Proporciona información sobre la conexión Ethernet.
→ Indica si DHCP está habilitado o no.
→ Proporciona la dirección IP asignada.
→ Indica el estado de la conexión.
→ Proporciona el nombre de dominio multicast (mDNS).

→ Proporciona información sobre la conexión Wi-Fi.
→ Proporciona la dirección IP asignada.
→ Indica el estado de la conexión.
→ Proporciona el nombre de la red inalámbrica (SSID).

→ Proporciona información sobre la conexión lowpan.
→ Proporciona la dirección IP asignada.
→ Indica el estado de la conexión.
→ Indica el número de nodos configurados.
→ Indica el número de nodos NO configurados.
→ Indica el número de sensores conectados.

➤ Endpoint para obtener información sobre WSN:

Este endpoint es utilizado para obtener información sobre la WSN a través de una solicitud GET a la API REST.

- Método HTTP: **GET**
- Ruta (Path): **/api/v1/wsn**
- Cuerpo de la solicitud (request body): Este endpoint no requiere ningún parámetro de entrada adicional en la solicitud.
- Respuesta (response): La respuesta proporciona una lista de objetos, cada uno representando un nodo en la red de sensores inalámbricos (WSN). Cada objeto contiene información detallada sobre un nodo específico, incluyendo:

```
[
  {
    "nodeid": 1,           → Identificador único del nodo en la red.
    "name": "None",       → Nombre asignado al nodo.
    "ipv6": "fd00::1",    → Dirección IPv6 asignada al nodo.
    "state": "configured", → Estado actual del nodo (ejemplo: configurado).
    "parent": 4,          → Identificador del nodo padre en la jerarquía de la red.
    "battery": "",        → Estado de la batería del nodo.
    "lifetime": "",       → Estado de la batería del nodo (si corresponde).
    "sensors": [         → Lista de sensores conectados al nodo.
      {
        "id": "1",
        "interface": "interno",
        "sensor": "temperature",
        "period": 500,
        "units": "°C",
        "tmin": "-20",
        "tmax": "60"
      },
      ...
    ]
  },
  ...
]
```

➤ Endpoint para obtener información sobre un nodo específico en la WSN:

Este endpoint se utiliza para obtener información detallada sobre un nodo específico en la WSN a través de una solicitud GET a la API REST.

- Método HTTP: **GET**
- Ruta (Path): **/api/v1/wsn/<nodeid>**
Este endpoint requiere un parámetro de ruta <nodeid>, que especifica el identificador único del nodo del que se desea obtener información.
- Respuesta (response): La respuesta proporciona información detallada sobre el nodo específico identificado por <nodeid>.

```
{
  "nodeid": 1,           → Identificador único del nodo en la red.
  "name": "None",       → Nombre asignado al nodo.
  "ipv6": "fd00::1",    → Dirección IPv6 asignada al nodo.
  "state": "configured", → Estado actual del nodo (ejemplo: configurado).
  "parent": 4,          → Identificador del nodo padre en la jerarquía de la red.
  "battery": "",        → Estado de la batería del nodo.
  "lifetime": "",       → Estado de la batería del nodo (si corresponde).
  "sensors": [         → Lista de sensores conectados al nodo.
    {
      "id": "1",
      "interface": "interno",
      "sensor": "temperature",
      "period": 500,
      "units": "°C",
      "tmin": "-20",
      "tmax": "60"
    },
    ...
  ]
}
```

➤ Endpoint para obtener la lista de sensores disponibles:

Este endpoint se utiliza para obtener la lista de sensores disponibles a través de una solicitud GET a la API REST.

- Método HTTP: **GET**
- Ruta (Path): **/api/v1/sensors**
- Cuerpo de la solicitud (request body): Este endpoint no requiere ningún parámetro de entrada adicional en la solicitud.
- Respuesta (response): La respuesta proporciona una lista de objetos, cada uno representando una interfaz de sensores disponibles.

```
[
  {
    "name": "interno",           → Nombre de la interfaz de sensores.
    "sensors": [               → Lista de sensores disponibles en esa interfaz.
      {
        "name": "temperatura"   → Nombre que describe el tipo de sensor.
      },
      ...
    ]
  },
  ...
]
```

➤ Endpoint para configurar un nuevo sensor en un nodo en la WSN:

Este endpoint se utiliza para configurar un nuevo sensor en un nodo específico en la WSN a través de una solicitud POST a la API REST.

- Método HTTP: **POST**
- Ruta (Path): **/api/v1/wsn/<nodeid>/sensors**
Este endpoint requiere un parámetro de ruta <nodeid>, que especifica el identificador único del nodo al que se desea agregar el sensor.
- Cuerpo de la solicitud (request body): El cuerpo de la petición contiene la información necesaria para configurar el nuevo sensor en el nodo, incluyendo:

```
{
  "interface": "4-20mA",       → Nombre de la interfaz del sensor.
  "sensor": "temperature",     → Tipo de sensor.
  "period": 500,              → Periodo de muestreo del sensor en milisegundos.
  "units": "°C",              → Unidades de medida del sensor.
  "tmin": "-20",              → Mínimo del rango de medición del sensor.
  "tmax": "60"                → Máximo del rango de medición del sensor.
}
```

- Respuesta (response): La respuesta confirma si el sensor del nodo se ha configurado con éxito.

➤ Endpoint para obtener la lista de sensores asociados a un nodo en la WSN.

Este endpoint se utiliza para obtener una lista de sensores conectados a un nodo específico en la WSN a través de una solicitud GET a la API REST.

- Método HTTP: **GET**
- Ruta (Path): **/api/v1/wsn/<nodeid>/sensors**
Este endpoint requiere un parámetro de ruta <nodeid>, que especifica el identificador único del nodo del que se desea obtener información.
- Respuesta (response): La respuesta proporciona una lista de objetos, cada uno representando un sensor conectado al nodo específico en la red de sensores inalámbricos (WSN). Cada objeto contiene:

```
[
  {
    "id": "1",                → Identificador único del sensor.
    "interface": "interno",  → Nombre de la interfaz a la que está conectado
    el sensor.
    "sensor": "temperature", → Tipo de sensor.
    "period": 500,           → Periodo de muestreo del sensor en ms.
    "units": "°C",           → Unidades de medida del sensor.
    "tmin": "-20",           → Mínimo del rango de medición del sensor.
    "tmax": "60"             → Máximo del rango de medición del sensor.
  },
  ...
]
```

➤ Endpoint para eliminar un sensor de un nodo en la WSN

Este endpoint se utiliza para eliminar un sensor específico de un nodo en la WSN a través de una solicitud DELETE a la API REST.

- Método HTTP: **DELETE**
- Ruta (Path): **/api/v1/wsn/<nodeid>/sensors**
Este endpoint requiere un parámetro de ruta <nodeid>, que especifica el identificador único del nodo al que se desea agregar el sensor.
- Cuerpo de la solicitud (request body): El cuerpo de la petición contiene el identificador único del sensor que se desea eliminar.

```
{
  "id": "21"                 → Identificador único del sensor que se desea eliminar.
}
```

- Respuesta (response): La respuesta confirma si el sensor del nodo se ha eliminado con éxito.

➤ Endpoint para actualizar un sensor en un nodo en la WSN:

Este endpoint se utiliza para actualizar la configuración de un sensor existente en un nodo específico en la WSN a través de una solicitud PUT a la API REST.

- Método HTTP: **PUT**
- Ruta (Path): **/api/v1/wsn/<nodeid>/sensors**
Este endpoint requiere un parámetro de ruta <nodeid>, que especifica el identificador único del nodo al que se desea agregar el sensor.
- Cuerpo de la solicitud (request body): El cuerpo de la petición contiene la información necesaria para configurar el nuevo sensor en el nodo.

```
{
  "id": "21",           → Identificador único del sensor que se desea
  actualizar.
  "interface": "4-20mA", → Nombre de la interfaz a la que está conectado el
  sensor.
  "sensor": "temperature", → Tipo de sensor.
  "period": 500,        → Periodo de muestreo del sensor en milisegundos.
  "units": "°C",       → Unidades de medida del sensor.
  "tmin": "-20",       → Mínimo del rango de medición del sensor.
  "tmax": "60"        → Máximo del rango de medición del sensor.
}
```

- Respuesta (response): La respuesta confirma si el sensor del nodo se ha configurado con éxito.

Anexo II: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030

Anexo al Trabajo de Fin de Grado: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030.

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

| Objetivos de Desarrollo Sostenibles | Alto | Medio | Bajo | No Procede |
|---|------|-------|------|------------|
| ODS 1. Fin de la pobreza. | | | | |
| ODS 2. Hambre cero. | | | | |
| ODS 3. Salud y bienestar. | | | | |
| ODS 4. Educación de calidad. | | | | |
| ODS 5. Igualdad de género. | | | | |
| ODS 6. Agua limpia y saneamiento. | | | | |
| ODS 7. Energía asequible y no contaminante. | | | | |
| ODS 8. Trabajo decente y crecimiento económico. | | | | |
| ODS 9. Industria, innovación e infraestructuras. | | | | |
| ODS 10. Reducción de las desigualdades. | | | | |
| ODS 11. Ciudades y comunidades sostenibles. | | | | |
| ODS 12. Producción y consumo responsables. | | | | |
| ODS 13. Acción por el clima. | | | | |
| ODS 14. Vida submarina. | | | | |
| ODS 15. Vida de ecosistemas terrestres. | | | | |
| ODS 16. Paz, justicia e instituciones sólidas. | | | | |
| ODS 17. Alianzas para lograr objetivos. | | | | |

Descripción de la alineación del TFG con los ODS con un grado de relación más alto:

- **ODS 3. Salud y bienestar (relación MEDIA)**
Las redes de sensores inalámbricos pueden ser utilizadas en el ámbito de la salud para monitorizar parámetros ambientales, como la calidad del aire o el agua, que afectan la salud pública, o para el seguimiento remoto de pacientes a través de sensores. Aunque el enfoque principal del proyecto no está en la salud, la plataforma facilita la creación y despliegue de redes WSN, por lo que tiene el potencial de ser aplicada en este ámbito, mejorando así el bienestar de las personas.

- **ODS 6. Agua limpia y saneamiento (relación ALTA)**
Las redes de sensores inalámbricos son una herramienta eficaz para monitorear la calidad del agua y detectar fugas o ineficiencias en los sistemas de distribución. A través de la plataforma propuesta, se puede facilitar la implementación de redes WSN que optimicen la gestión de recursos hídricos, contribuyendo de manera significativa a la sostenibilidad del agua y a garantizar su disponibilidad para todas las personas.
- **ODS 7. Energía asequible y no contaminante (relación ALTA)**
Las redes de sensores y las redes definidas por software juegan un papel clave en la creación de redes eléctricas inteligentes (smart grids), que optimizan el uso de la energía y permiten una mayor integración de fuentes renovables. La plataforma web desarrollada permite gestionar eficientemente estas redes, contribuyendo a la reducción de desperdicios energéticos y promoviendo una transición hacia fuentes de energía más limpias y accesibles.
- **ODS 9. Industria, innovación e infraestructura (relación ALTA)**
El trabajo fomenta la innovación tecnológica y mejora las infraestructuras a través de la implementación de tecnologías avanzadas como las redes definidas por software. La plataforma que se está desarrollando no solo facilita el uso de tecnologías complejas, sino que también reduce la barrera de entrada para usuarios menos familiarizados con las redes de sensores, democratizando su acceso y fomentando la innovación en diversos sectores industriales y tecnológicos.
- **ODS 11. Ciudades y comunicaciones sostenibles (relación ALTO)**
Las redes de sensores inalámbricos son esenciales para la implementación de soluciones en las ciudades inteligentes, ya que permiten la monitorización y gestión eficiente de servicios urbanos, como el tráfico, el consumo de energía y la calidad del aire. La plataforma propuesta facilita el despliegue y gestión de estas redes, contribuyendo directamente a mejorar la sostenibilidad de las ciudades y la calidad de vida de sus habitantes.
- **ODS 12. Producción y consumo responsables (relación MEDIA)**
Las redes de sensores pueden optimizar el uso de recursos en procesos industriales y agrícolas. Al permitir la monitorización en tiempo real y la simulación de redes, el proyecto contribuye a una producción más eficiente y sostenible, reduciendo el desperdicio y promoviendo un uso más racional de los recursos, especialmente en sectores clave como la agricultura de precisión y la industria.
- **ODS 13. Acción por el clima (relación ALTA)**
Las redes de sensores son herramientas valiosas para el monitoreo ambiental y la gestión de recursos naturales, permitiendo medir y reducir el impacto de diversas actividades humanas sobre el clima. El uso de la plataforma para desplegar estas redes facilita la recopilación de datos cruciales para mitigar los efectos del cambio climático, como la deforestación, el control de emisiones y el monitoreo de ecosistemas vulnerables.

- **ODS 15. Vida de ecosistemas terrestres (relación ALTA)**
Las redes de sensores pueden ser utilizadas para monitorear la salud de los ecosistemas forestales y agrícolas, así como para detectar cambios en la biodiversidad o la calidad del suelo y del agua. La plataforma que facilita el despliegue de estas redes contribuye a la conservación de los ecosistemas y a la gestión sostenible de los recursos naturales, aspectos clave para la protección del medio ambiente y la biodiversidad.
- **ODS 17. Alianzas para lograr los objetivos (relación MEDIA)**
Al ofrecer una plataforma tecnológica accesible y flexible, se fomenta la colaboración entre diferentes sectores y regiones, facilitando el intercambio de conocimientos y promoviendo la transferencia tecnológica. Esta plataforma podría ser utilizada como parte de iniciativas globales para mejorar la cooperación técnica en áreas clave como la gestión de redes inteligentes y la monitorización ambiental, apoyando así los esfuerzos para alcanzar los objetivos de desarrollo sostenible a nivel mundial.