



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACSISTEL

INGENIERÍA EN TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

**Implementación de un prototipo de una red
IoT con tecnología LoRaWAN para el
análisis de variables eléctricas en un sistema
de distribución de baja tensión**

Carlos Bryan De La Cruz Rodríguez

Dirigido por:

Ing. Fernando Chamba Macas, Mgt.

La Libertad – 2026



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Róvira Jurado, PhD.
DIRECTOR DE LA CARRERA

Ing. Fernando Chamba Macas, Mgt.
TUTOR

Ing. Edison Stalin Coral Salinas, M.Sc.
DOCENTE ESPECIALISTA

Ing. Luis Amaya Fariño, Mgt.
DOCENTE GUÍA UIC

Ing. Corina Gonzabay De La A, Mgt.
SECRETARIA



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **De La Cruz Rodríguez Carlos Bryan**, como requerimiento para la obtención del título de Ingeniero en telecomunicaciones.

La Libertad, a los 10 días del mes de agosto del año 2026

TUTOR

Ing. Fernando Chamba Macas, Mgt.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, De La Cruz Rodríguez Carlos Bryan

DECLARO QUE:

El trabajo de Titulación, **Implementación de un prototipo de una red IoT con tecnología LoRaWAN para el análisis de variable eléctricas en sistema de distribución de baja tensión**, previo a la obtención del título en Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría. En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 10 días del mes de agosto del año 2026

EL AUTOR

De La Cruz Rodríguez Carlos



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, De La Cruz Rodríguez Carlos Bryan

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 10 días del mes de agosto del año 2026

EL AUTOR

De La Cruz Rodríguez Carlos



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **Implementación de un prototipo de una red IoT con tecnología LoRaWAN para el análisis de variable eléctricas en sistema de distribución de baja tensión**, presentado por el estudiante **De La Cruz Rodríguez Carlos Bryan** fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 01%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



Certificado de análisis
Compilatio Magister+ | UPSE-ECU

TRABAJO DE INTEGRACION CURRICULAR

ID : 178846b7f1c6597630d5224f409388f0dcc9fde9



<1%
Textos sospechosos

Nombre del fichero: TRABAJO DE INTEGRACION CURRICULAR.rtf
Tamaño del archivo original: 23,62 MB
Número de palabras: 42.809

Depositante: AMAYA FARIÑO LUIS MIGUEL
Fecha de depósito: 20 de agosto de 2026
Tipo de carga: interface
fecha de fin de análisis: 20 de agosto de 2026

TUTOR

Ing. Fernando Chamba Macas, Mgt.

Agradecimiento

En primer lugar, quiero dar las gracias a Dios por darme la fuerza, la sabiduría y la perseverancia necesarias para superar cada desafío y permitirme terminar esta importante etapa de mi formación profesional.

Quiero expresar mi más profundo agradecimiento a mis padres por su amor incondicional, sus sacrificios, su apoyo constante y la confianza que siempre depositaron en mí.

También a mis abuelos, por su cariño, sus enseñanzas y el ejemplo de esfuerzo y dedicación que me han brindado a lo largo de mi vida. Su apoyo y sus palabras de aliento han sido una fuente permanente de motivación.

A mi hermano, por estar presente en este proceso y brindarme su compañía y apoyo cuando más lo necesitaba.

A mi pareja, Cindy Balón, por su paciencia, comprensión y apoyo incondicional durante el desarrollo de esta investigación. Gracias por acompañarme en cada etapa, por motivarme a no rendirme y por celebrar conmigo cada logro.

A mis amigos, gracias por todo. Gracias por estar ahí, por sus consejos, su ánimo y su apoyo. Gracias a todos por hacer que este proceso académico fuera más llevadero.

Carlos Bryan De La Cruz Rodríguez

Dedicatoria

Dedico este trabajo, con profundo cariño y gratitud, a mis padres, por ser el pilar fundamental de mi vida, por su esfuerzo, sacrificio y apoyo incondicional en cada una de las metas que me he propuesto; a mi abuelo paterno, que, desde que tengo memoria, me ha brindado su apoyo, sus consejos y la motivación para nunca dejar de luchar por mis sueños. A mi hermano, por acompañarme y formar parte de este importante camino.

A mi novia, Cindy Balón, por convertirse en mi refugio durante este proceso, por permanecer a mi lado incluso en los días de mayor cansancio, incertidumbre y dificultad, por creer en mí cuando yo mismo dudaba, por brindarme palabras de aliento cuando más lo necesitaba y por recordarme, con su amor y paciencia, que cada esfuerzo tendría su recompensa. Gracias por acompañarme en cada paso de este camino, por celebrar conmigo cada pequeño logro y por demostrarme que los sueños se disfrutan aún más cuando se comparten con la persona adecuada.

Finalmente, dedico este logro a mi querida mascota, Bonnie, cuya compañía, cariño incondicional y alegría hicieron más llevaderos los días de estudio y esfuerzo.

Gracias a todos ustedes por ser parte fundamental de este logro, que también les pertenece.

Carlos Bryan De La Cruz Rodríguez

Resumen

El presente trabajo describe la implementación de un prototipo de red IoT basado en la tecnología LoRaWAN para monitorizar variables eléctricas en sistemas de distribución de baja tensión. El sistema integra un sensor de corriente (SCT-013) y otro de voltaje (ZMPT101B), que están conectados a un ESP32 y a un nodo CubeCell AB02. Estos dispositivos transmiten los datos hacia un gateway SenseCAP M2. La información se procesa mediante The Things Network (TTN) y se visualiza en tiempo real mediante Node-RED, que también genera alertas por Telegram ante anomalías como sobretensión o subtensión. Las pruebas realizadas confirmaron la fiabilidad, precisión y bajo consumo del prototipo, lo que evidencia su viabilidad para la supervisión remota y la detección temprana de fallas eléctricas.

Palabras clave: IoT, LoRaWAN, monitoreo eléctrico, ESP32, baja tensión.

Abstract

This work describes the implementation of an Internet of Things (IoT) network prototype based on LoRaWAN technology for monitoring electrical variables in low-voltage distribution systems. The system integrates current (SCT-013) and voltage (ZMPT101B) sensors, which are connected to an ESP32 and a CubeCell AB02 node. These nodes then transmit the data to a SenseCAP M2 gateway. This information is processed via The Things Network (TTN) and displayed in real time using Node-RED, which also generates Telegram alerts when anomalies such as overvoltage or undervoltage are detected. Tests confirmed the reliability, accuracy and low power consumption of the prototype, demonstrating its feasibility for remote supervision and the early detection of electrical faults.

Keywords: IoT, LoRaWAN, electrical monitoring, ESP32, low voltage.

Lista de abreviaturas

IoT: Internet of Things (Internet de las cosas).

LoRaWAN: Long Range Wide Area Network.

LPWAN: Low Power Wide Area Network.

WSN: Wireless Sensor Network (Red de sensores inalámbricos).

TTN: The Things Network.

NB-IoT: Narrowband IoT.

LTE/LTE-M: Long Term Evolution/LTE for Machines.

BLE: Bluetooth Low Energy.

UART: Universal Asynchronous Receiver Transmitter.

SPI: Serial Peripheral Interface.

I2C: Inter-Integrated Circuit.

SWD: Serial Wire Debug.

GPIO: General Purpose Input/Output.

PWM: Pulse Width Modulation.

TCP/IP: Transmission Control Protocol / Internet Protocol.

HTTPS: Hypertext Transfer Protocol Secure.

MQTT: Message Queuing Telemetry Transport.

UTC: Coordinated Universal Time.

SCT-013: Sensor de corriente tipo transformador (Split Current Transformer).

ZMPT101B: Sensor de voltaje.

ISM: Industrial, Scientific and Medical (Banda de frecuencia).

PCB: Printed Circuit Board (Placa de circuito impreso).

Índice General

Agradecimiento	7
Dedicatoria.....	8
Resumen.....	9
Abstract.....	10
Lista de abreviaturas	11
Índice de figuras.....	1
Índice de tablas.....	1
Introducción	1
Capítulo 1.....	2
1.1 Antecedentes	2
1.2 Descripción de proyecto.....	5
1.3 Objetivos	8
1.3.1 Objetivo general.....	8
1.3.2 Objetivos específicos	8
1.4 Justificación	9
1.5 Problemática	11
1.6 Alcance del proyecto	13
1.7 Metodología.....	15
1.7.1 Investigación descriptiva	15

1.7.2	Investigación documental	16
1.7.3	Investigación aplicada.....	16
1.8	Marco contextual.....	17
Capítulo 2		18
2.1	Marco conceptual	18
2.2	Marco teórico.....	19
2.2.1	Monitoreo de corriente y voltaje en baja tensión.....	19
2.2.2	Sensores para la medición de variables eléctricas	22
2.2.3	Internet de las cosas (IoT).....	26
2.2.4	Arquitectura de una red IoT	28
2.2.5	Tecnología de comunicación LPWAN.....	29
2.2.6	Tecnología LoRa y protocolo LoRaWAN.....	33
2.2.7	Protocolos de comunicación IoT	34
2.2.8	Gateway	36
2.2.9	Plataforma IoT y visualización de datos	38
Capítulo 3		41
3.1	Arquitectura y conexión de los componentes del sistema IoT propuesto.....	42
3.1.1	Arquitectura del sistema	42
3.2	Elementos de hardware para la adquisición y transmisión de datos.	44
3.2.1	Microcontrolador ESP32	44

3.2.2	Sensor de voltaje AC ZMPT101B	46
3.2.3	Sensor de corriente SCT-013	47
3.2.4	Nodo LoRaWAN CubeCell AB02	48
3.2.5	Gateway LoRaWAN SenseCAP M2	51
3.2.6	Batería de litio de 3,7 V	53
3.3	Herramientas de software para la gestión y visualización de datos	53
3.3.1	Plataforma IoT - The Things Network (TTN)	54
3.3.2	Node - RED: plataforma de visualización	54
3.3.3	Arduino IDE.....	55
3.3.4	Bróker Mosquito (Eclipse Mosquito)	57
3.4	Elaboración y diseño de la propuesta del sistema	57
3.4.1	Diseño electrónico de conexiones.....	58
3.4.2	Diseño esquemático del circuito en Proteus	60
3.4.3	Diseño del circuito impreso (PCB layout)	61
3.4.4	Diseño tridimensional del prototipo.....	62
3.5	Configuración e integración del sistema IoT	63
3.5.1	Configuración del gateway SenseCAP M2.....	63
3.5.2	Configuración de la plataforma The Things Network	68
3.5.3	Configuración del gateway SenseCAP M2 en la plataforma TTN	70
3.5.4	Configuración de nodos finales en la aplicación TTN.....	73

3.5.5	El decodificador de los mensajes en TTN	76
3.5.6	Configuración del servidor Node-RED	78
3.5.7	Definición de los nodos por MQTT	82
3.5.8	Diseño del panel de control gráfico con los nodos ui-gauge y ui-charge	91
3.5.9	Aplicación Telegram	98
3.5.10	Configuración del historial de registro de anomalías	105
3.5.11	Diagrama de flujo final en Node-RED	107
Capítulo 4		108
4.1	Resultados	108
4.1.1	Componentes del prototipo	108
4.2	Resultado de las pruebas de funcionamiento	110
4.2.1	Validación del rendimiento	110
4.3	Consumo energético del prototipo	115
Conclusiones		117
Recomendaciones		118
Bibliografía		119
Anexos		124
Anexo 1 – Código de microcontrolador ESP32 en Arduino IDE		125
Anexo 2 – Código del módulo CubeCell AB02 en Arduino IDE		129
Anexo 3 – JSON compact de Node-RED (flujo)		135

Anexo 4 – Payload formatters para CubeCell AB02 en TTN	141
---	------------

Índice de figuras

Figura 1. Sensores de IoT en todos sus ámbitos	26
Figura 2. Arquitectura de Sigfox	30
Figura 3. Arquitectura de una red LoRa	30
Figura 4. Arquitectura de red NB-IoT	31
Figura 5. Arquitectura de una red LTE-M	32
Figura 6. Diagrama del sistema de monitorización energética	43
Figura 7. Escenario topológico del sistema	44
Figura 8. Microcontrolador ESP32	45
Figura 9. Sensor de voltaje	46
Figura 10. Sensor de corriente	48
Figura 11. Módulo CubeCell AB02	49
Figura 12. Gateway SenseCAP M2	51
Figura 13. Batería de litio	53
Figura 14. Plataforma TTN	54
Figura 15. Diseño de conexiones del sistema	59
Figura 16. Diseño esquemático en Proteus	61
Figura 17. Diseño de PCB para el prototipo	62
Figura 18. Diseño 3D (caja)	63
Figura 19. Información y credenciales del gateway	64
Figura 20. Interfaz del gateway SenseCAP M2	64
Figura 21. Elección del plan de canal	65
Figura 22. Configuración de red en el gateway	65

Figura 23. Interfaz para escanear redes inalámbricas	66
Figura 24. Establecer una red inalámbrica	66
Figura 25. Ingresar datos para nueva red inalámbrica.....	67
Figura 26. Configuración para asignar LoRaWAN.....	68
Figura 27. Crear una cuenta en la plataforma TTN.....	68
Figura 28. Selección de región en la plataforma TTN	69
Figura 29. Interfaz inicial de TTN.....	70
Figura 30. Selección del registro del gateway	70
Figura 31. Métodos de registro	71
Figura 32. Interfaz de registro del gateway	72
Figura 33. Verificación del estado del gateway	72
Figura 34. Crear aplicación para nodos.....	73
Figura 35. Campos para aplicación en TTN.....	73
Figura 36. Crear aplicación.....	74
Figura 37. Registro del CubeCell en TTN	74
Figura 38. Credenciales para CubeCell AB02.....	75
Figura 39. Dispositivo integrado en TTN.....	75
Figura 40. Configuración uplink en TTN	76
Figura 41. Configuración de los formatters de carga útil en TTN	77
Figura 42. Acceso al servidor Node-RED.....	78
Figura 43. Dirección IP para ejecutar Node-RED	78
Figura 44. Interfaz del servidor Node-RED	79
Figura 45. Sección de administración de paletas	79

Figura 46. Instalación de las librerías necesarias para Node-RED.....	80
Figura 47. Sección de la paleta de nodos.....	81
Figura 48. Búsqueda del nodo MQTT in	82
Figura 49. Entorno de configuración de MQTT in.....	83
Figura 50. Configuración del servidor y el puerto	83
Figura 51. Configuración de seguridad del nodo	84
Figura 52. Opción MQTT en TTN	84
Figura 53. Credenciales de MQTT en TTN.....	85
Figura 54. Credenciales en MQTT in.....	85
Figura 55. Directrices para el modelo Topic en TTN.....	86
Figura 56. Nodo MQTT in en el apartado final	86
Figura 57. Nodo JSON.....	87
Figura 58. Nodo debug	88
Figura 59. Ventana de mensajes de depuración	88
Figura 60. Nodo change.....	89
Figura 61. Configuración del nodo change.....	90
Figura 62. Muestra de datos limpios	90
Figura 63. Ventana de configuración del nodo gauge	91
Figura 64. Nodo gauge asignado para voltaje	92
Figura 65. Asignación de variable de voltaje.....	92
Figura 66. Asignación de nombre a nodo ui-page	93
Figura 67. Asignación de nombre al nodo ui-group.....	93
Figura 68. Añadir nodos para las variables de corriente y potencia.....	94

Figura 69. Configuración del nodo corriente	95
Figura 70. Configuración para nodo de potencia	95
Figura 71. Nodos independientes.....	96
Figura 72. Configuración de la etiqueta msg. topic	97
Figura 73. Conexión entre nodos	97
Figura 74. Configuración del nodo ui-charge.....	98
Figura 75. Chatbot father.....	99
Figura 76. Creación del chatbot personal.....	99
Figura 77. Creación de alertas de energía	100
Figura 78. Chat bot creado	100
Figura 79. Credencial del ID del chat ID	101
Figura 80. Configuración del nodo de filtro de anomalías V	102
Figura 81. Configuración del nodo de plantilla (sobretensión)	103
Figura 82. Configuración del nodo template subtenión.....	103
Figura 83. Node sender.....	104
Figura 84. Credenciales en el nodo sender	104
Figura 85. Configuración función para historial de anomalías.....	105
Figura 86. Configuración del historial.....	106
Figura 87. Diagrama de flujo final	107
Figura 88. Implementación del prototipo	109
Figura 89. Prototipo implementado, gateway SenseCAP M2 y antena.....	110
Figura 90. Registro de paquetes LoRaWAN.....	111
Figura 91. Registro de mensajes uplink y downlink en The Things Network.....	112

Figura 92. Panel de depuración de Node-RED con recepción de variables eléctricas desde TTN	113
Figura 93. Notificaciones de anomalías enviadas mediante el bot de Telegram.....	114
Figura 94. Presentación del dashboard final	115

Índice de tablas

Tabla 1	Comparativa de trabajos en diferentes universidades	12
Tabla 2	Anomalías comunes en sistemas de baja tensión	21
Tabla 3	Medición manual vs medición IoT.....	21
Tabla 4	Comparativa de sensores de corriente	23
Tabla 5	Tipos de sensores de voltaje	25
Tabla 6	Tipos de tecnologías LPWAN	32
Tabla 7	Tabla comparativa de los protocolos de comunicación IoT	35
Tabla 8	Equipos gateway LoRaWAN	37
Tabla 9	Características del ESP32.....	45
Tabla 10	Características del sensor de voltaje.....	47
Tabla 11	Características del sensor de corriente	48
Tabla 12	Parámetros importantes del módulo CubeCell AB02	50
Tabla 13	Descripción del gateway SenseCAP M2	52
Tabla 14	Tabla de librerías para Arduino	56
Tabla 15	Consumo energético del prototipo	115

Introducción

La supervisión de las variables eléctricas en los sistemas de distribución de baja tensión es clave para garantizar la eficiencia energética, la seguridad de las instalaciones y la detección rápida de anomalías. Tradicionalmente, este seguimiento se ha realizado mediante inspecciones manuales o instrumentos de medición puntual, lo que limita la capacidad de respuesta ante eventos transitorios y dificulta la identificación oportuna de anomalías, como sobretensiones o subtensiones.

Frente a esta problemática, el Internet de las cosas (IoT) se ha afianzado como una alternativa viable para la supervisión remota y en tiempo real de parámetros eléctricos, gracias a la disponibilidad de microcontroladores de bajo coste, sensores precisos y tecnologías de comunicación inalámbrica de largo alcance, como LoRaWAN, que permiten transmitir información a grandes distancias con un bajo consumo energético.

En este contexto, el presente trabajo propone la implementación de un prototipo de red IoT que incorpora sensores de corriente y voltaje, un microcontrolador ESP32 y un nodo LoRaWAN CubeCell AB02. Los datos se transmiten a través de un gateway SenseCAP M2 a la plataforma The Things Network (TTN). Posteriormente, la información se procesa y visualiza mediante Node-RED, una plataforma que también permite generar alertas automáticas por Telegram ante la detección de anomalías eléctricas.

El presente documento se organiza en cuatro capítulos: el primero trata los antecedentes, la problemática, los objetivos y la metodología del proyecto; el segundo despliega el marco teórico relacionado con las tecnologías empleadas; el tercero explica el diseño del prototipo y el cuarto presenta los resultados obtenidos, las pruebas de funcionamiento y las conclusiones.

Capítulo 1

1.1 Antecedentes

En la actualidad, el análisis y el seguimiento de parámetros eléctricos en sistemas de distribución de baja tensión han adquirido gran importancia debido a la necesidad de garantizar la eficiencia energética, la protección de las instalaciones y la detección temprana de las fallas. Diversos estudios han abordado el uso de la tecnología del Internet de las cosas (IoT) para diseñar prototipos escalables y económicos con capacidad para medir variables eléctricas, como el voltaje y la corriente, y para transmitir estos datos de manera inalámbrica a plataformas en la nube para su análisis. La combinación de microcontroladores, como el ESP32 y el gateway LoRaWAN SenseCAP M2, junto con los protocolos de comunicación MQTT, ha permitido integrar soluciones que combinan electrónica, telecomunicaciones y visualización en tiempo real.

En este estudio, relacionado con redes IoT y comunicación inalámbrica, se ha implementado un sistema de alarma comunitaria junto con un gateway SenseCAP M2, que permite la transmisión de datos de manera fiable mediante redes inalámbricas. La integración de sensores con el gateway permite la monitorización y el control de circunstancias en tiempo real, lo que demuestra la eficacia de una buena arquitectura para la gestión remota de información[1].

En 2022, Ramos Gallegos implementó un prototipo con tecnología LoRaWAN para la medición remota de la temperatura y la humedad, lo que demostró el potencial de las redes inalámbricas de baja potencia en el monitoreo ambiental. La conexión de sensores a una infraestructura IoT permitió la transmisión eficiente y fiable de datos a largas distancias, lo que destacó la escalabilidad y flexibilidad de esta tecnología en escenarios reales de implementación [2].

En 2021, se desarrolló un prototipo de sistema IoT para la automatización del hogar destinado a personas con movilidad reducida. La propuesta integró módulos de comunicación Wi-Fi basados en ESP8266 y Raspberry Pi, lo que permitió controlar y monitorizar diversos dispositivos mediante una plataforma accesible. Este trabajo demostró la viabilidad de construir prototipos funcionales de IoT con un enfoque práctico y orientado a mejorar la calidad de vida [3].

En la Universidad de Guayaquil, Espinoza Bazán implementó un sistema de alumbrado público inteligente que utilizó sensores conectados a un módulo ESP32 para medir la corriente y el consumo energético. La comunicación se realizó mediante LoRaWAN y el almacenamiento de datos a través de la plataforma Ubidots, lo que permitió gestionar alertas y visualizar información en tiempo real. Este desarrollo demuestra el potencial de las tecnologías IoT para el monitoreo energético a partir de parámetros eléctricos [4].

En 2021, Del Rosario y Meza Mora implementaron una red IoT basada en LoRaWAN con un gateway centralizado y una interfaz para visualizar los datos remotos. La investigación se centró en la integración de nodos censados con capacidad de transmisión inalámbrica, lo que demostró la utilidad de las redes IoT para el envío y análisis de datos. De esta forma, se puso de manifiesto un antecedente significativo sobre cómo estas arquitecturas pueden convertir los procesos de supervisión y control en distintos entornos tecnológicos [5].

En este trabajo, se diseñó e implementó un prototipo de IoT orientado al monitoreo remoto de la calidad del agua en sistemas de piscicultura, utilizando un microcontrolador ESP8266 y una Raspberry Pi como unidad de procesamiento. La arquitectura contempló la integración de sensores especializados y el uso de plataformas en la nube, como Thingspeak y Grafana, para la recolección, almacenamiento y visualización de datos en tiempo real. El proyecto demostró la eficacia de la tecnología IoT en el control de variables críticas en entornos productivos, destacando su

aplicabilidad en escenarios donde la información oportuna es esencial para la toma de decisiones [6].

En la Universidad Nacional de Loja, Castillo implementó un sistema de IoT para monitorizar variables eléctricas de baja tensión en viviendas, empleando sensores ZMPT101B para voltaje y SCT-013 para corriente. El prototipo, basado en ESP32, permitió registrar en tiempo real el consumo eléctrico y analizar los patrones de uso de energía en el entorno doméstico, con los datos almacenados en la plataforma Ubidots [7].

En este estudio se desarrolló un sistema de recogida de residuos basado en tecnologías LPWAN e IoT, en el que se empleó un gateway SenseCAP M2 como nodo central de la red. Los datos adquiridos por los sensores ultrasónicos se procesaron a través de Node-RED, lo que permitió gestionar la información de manera eficiente y generar notificaciones automáticamente en la aplicación Telegram, demostrando así la efectividad de esta arquitectura en entornos urbanos [8].

En 2023 se propuso un prototipo con gateway para el monitoreo y control de un variador de frecuencia, que incorporaba comunicación Modbus RS-485 y usaba la plataforma V-Net. El sistema permitió registrar variables eléctricas, como la corriente, el voltaje y la frecuencia, con un nivel de error inferior al 0,25 %, lo que demostró la calidad del proyecto, su precisión y su aplicabilidad en entornos industriales [9].

Villacis Moran desarrolló un dispositivo IoT de bajo costo destinado al monitoreo de tableros eléctricos industriales con capacidad para medir parámetros como la corriente y el voltaje. El sistema emplea un ESP32 y se comunica mediante el protocolo MQTT, lo que posibilita la

transmisión de datos en tiempo real y demuestra su utilidad para la gestión eficiente del consumo eléctrico [10].

Por tanto, se espera que este proyecto contribuya al desarrollo de un prototipo IoT funcional para el monitoreo de variables eléctricas y sirva de referencia para futuras investigaciones, fortaleciendo las capacidades técnicas de estudiantes y profesionales, al mismo tiempo que se promueve el uso de tecnologías digitales para optimizar la eficiencia energética y la supervisión de sistemas eléctricos de baja tensión.

1.2 Descripción de proyecto

El presente proyecto tiene como finalidad desarrollar un prototipo de red IoT destinado al monitoreo de parámetros eléctricos en sistemas de distribución de baja tensión, con un enfoque particular en la detección de irregularidades en las líneas. La propuesta no solo busca registrar valores de corriente y voltaje en las cuatro líneas del panel de control, sino también identificar posibles fallos, como la presencia de corriente en la línea de tierra, que podría indicar un sistema defectuoso o un cableado inadecuado. Este enfoque está respaldado por estudios que resaltan la importancia de la supervisión en tiempo real para optimizar la eficiencia energética y garantizar la seguridad de los sistemas eléctricos de baja tensión [11].

La arquitectura del sistema se basa en la integración de sensores de corriente (SCT-013) y de voltaje (ZMPT101B), que están conectados a un microcontrolador ESP32 y a un CubeCell AB02 Plus. Estos dispositivos permiten adquirir datos eléctricos de manera precisa y eficiente. La elección de estos componentes se basa en investigaciones que demuestran su eficacia en

aplicaciones de monitorización energética y ofrecen una solución económica y altamente fiable [10].

La transmisión de datos se lleva a cabo mediante un gateway SenseCAP M2, que emplea la tecnología LoRaWAN para enviar la información recopilada a través de redes de área amplia de baja potencia. Esta tecnología se utiliza ampliamente en aplicaciones de monitorización remota, ya que ofrece ventajas como un bajo consumo de energía y un gran alcance, características importantes para la supervisión de sistemas eléctricos en entornos residenciales e industriales.

Una vez que el gateway ha recibido los datos, se utiliza la comunicación Wi-Fi para visualizarlos en tiempo real. Para procesar y presentar la información de manera accesible, se emplearán plataformas como The Things Network (TTN) y Node-RED. Esta infraestructura permite generar alertas automáticas ante la detección de alguna anomalía, lo que facilita la intervención temprana y la toma de decisiones informadas.

El sistema propuesto no solo se centra en la medición de parámetros eléctricos, sino también en la identificación de anomalías específicas, como la presencia de corriente en la línea de tierra. Este tipo de fallos puede indicar problemas como conexiones incorrectas o fallos en el aislamiento, que ponen en riesgo la seguridad del sistema eléctrico. La capacidad de detectar y notificar estas irregularidades en tiempo real es fundamental para mantener la integridad y la seguridad de las instalaciones.

Asimismo, el proyecto pretende demostrar la viabilidad de integrar la tecnología IoT con redes inalámbricas de baja potencia para monitorizar sistemas eléctricos. Esta integración permite la supervisión continua y remota, reduce la necesidad de intervenciones manuales y aumenta la eficiencia operativa. Estudios recientes han puesto de manifiesto la eficacia de este tipo de

soluciones para mejorar la gestión energética y prevenir fallos en las infraestructuras eléctricas [12].

Finalmente, se espera que este prototipo sirva de referencia para futuras investigaciones y desarrollos en el ámbito del monitoreo eléctrico inteligente. Al proporcionar una solución accesible y escalable, el proyecto contribuye al avance de las tecnologías de supervisión energética y al fortalecimiento de la capacidad técnica de los estudiantes y profesores del sector eléctrico.

1.3 Objetivos

1.3.1 Objetivo general

Implementar un prototipo de red IoT mediante tecnología LoRaWAN para el análisis de variables eléctricas en un tablero de distribución de baja tensión, con el fin de supervisar en tiempo real las condiciones eléctricas y facilitar la detección de anomalías.

1.3.2 Objetivos específicos

- Desarrollar un sistema de adquisición de variables eléctricas mediante sensores de corriente SCT-013 y voltaje ZMPT101B, integrados a un microcontrolador ESP32.
- Implementar la transmisión inalámbrica de datos mediante tecnología LoRaWAN, empleando un nodo CubeCell AB02 y un gateway SenseCAP M2, con su respectiva configuración.
- Configurar la plataforma The Things Network (TTN) e integrar protocolos MQTT con Node-RED para la visualización en tiempo real y la implementación de un registro histórico de las anomalías eléctricas.
- Realizar pruebas del funcionamiento del sistema para evaluar su medición, eficiencia y fiabilidad.

1.4 Justificación

La implementación de un prototipo de red IoT para el monitoreo de sistemas eléctricos es una herramienta estratégica para optimizar la gestión energética y garantizar la seguridad de los usuarios. Este tipo de sistemas permite supervisar de manera remota el comportamiento de la corriente y el voltaje, lo que facilita la detección temprana de irregularidades que podrían derivar en fallos críticos o sobrecargas. La capacidad de monitorización en tiempo real es esencial para mejorar la eficiencia de la infraestructura eléctrica y reducir los riesgos operativos [11].

La integración de sensores de corriente y voltaje, junto con microcontroladores como el ESP32 y el CubeCell AB02 Plus, permite recolectar datos de manera precisa y fiable. Esta información no solo permite observar el comportamiento normal del sistema eléctrico, sino también identificar anomalías específicas, como la presencia de corriente en la línea de tierra, que indica fallos en el aislamiento o en la conexión del sistema. De esta forma, se promueve una supervisión inteligente que anticipa problemas antes de que se conviertan en incidentes mayores [10].

La adopción de tecnologías de comunicación inalámbrica, como LoRaWAN y Wi-Fi, potencia la eficacia del sistema, ya que permite la transmisión de datos a largas distancias con bajo consumo energético. Esta característica es fundamental en entornos residenciales e industriales, donde la infraestructura eléctrica se distribuye en diferentes puntos. La implementación de estas tecnologías garantiza que los datos de los sensores lleguen de manera segura a las plataformas de visualización y análisis, lo que consolida un sistema fiable y escalable.

La información puede visualizarse en tiempo real mediante plataformas como Node-RED y TTN, lo que facilita su interpretación y permite generar alertas automáticas cuando se detectan anomalías. De este modo, se aumenta la capacidad de respuesta ante eventos eléctricos adversos y

se ayuda a prevenir daños en los equipos o accidentes eléctricos. Además, la visualización clara de la información permite a los usuarios tomar decisiones informadas sobre el funcionamiento del sistema [13].

La integración de sistemas IoT en la formación académica y profesional es un elemento de valor añadido para el desarrollo de competencias técnicas. Los estudiantes y profesionales que participan en la implementación y el análisis de estos sistemas adquieren habilidades prácticas esenciales para la gestión de infraestructuras eléctricas modernas, lo que fomenta la innovación y el conocimiento aplicado en entornos controlados [12].

La implementación de este prototipo también tiene implicaciones en términos de sostenibilidad y eficiencia energética. La supervisión continua permite identificar patrones de consumo, prevenir el desperdicio de energía y optimizar el uso de los recursos eléctricos. Esto no solo contribuye a la reducción de los costes operativos, sino que también fomenta un uso responsable de la energía, en consonancia con las prácticas de eficiencia energética y desarrollo tecnológico sostenible.

1.5 Problemática

La ausencia de monitorización continua en los sistemas de distribución de baja tensión limita la capacidad de detectar averías y anomalías de manera oportuna. Sin un sistema que recoja datos automáticamente, los problemas de voltaje o de corriente pueden pasar desapercibidos, lo que aumenta el riesgo de sobrecargas, daños en los equipos y posibles accidentes eléctricos [14].

La detección de anomalías, como la presencia de corriente en la línea de tierra, supone un gran desafío. La falta de instrumentos o sistemas inteligentes que detecten este tipo de irregularidades aumenta el riesgo de accidentes eléctricos y compromete la seguridad de los usuarios, por lo que es evidente la necesidad de soluciones que permitan un control más preciso [10].

La limitada integración de tecnologías de comunicación inalámbrica en la supervisión de sistemas eléctricos reduce la capacidad de respuesta en tiempo real de las plataformas de análisis ante eventos transitorios. Esta restricción reduce la capacidad de respuesta ante fallos y limita la eficiencia en la gestión de los sistemas eléctricos, lo que obliga a realizar intervenciones manuales que consumen tiempo y recursos [13].

La falta de plataformas de visualización accesibles para los datos recopilados por sensores eléctricos dificulta la interpretación y el análisis de la información. Esto provoca retrasos en la toma de decisiones y reduce la eficacia de los procesos de supervisión, por lo que no se pueden aplicar medidas preventivas oportunas [13].

Tabla 1*Comparativa de trabajos en diferentes universidades*

Autor / Año	Universidad	Tipo de tesis / Trabajo	Tema principal / Aporte
UPSE			
Ricardo Plúas & Olaves Rosales (2025)	UPSE	Proyecto de titulación	Sistema de alarma comunitaria usando LPWAN e IoT para seguridad barrial.
Yagual Solano (2025)	UPSE	Proyecto de titulación	IoT y LPWAN aplicados a la gestión de residuos en el sector Ballenita.
EPN			
Ramos Gallegos (2022)	EPN	Proyecto de titulación	Estudio e implementación de LoRaWAN para medición de humedad y temperatura en agricultura.
Espinoza Altamirano (2021)	EPN	Proyecto de titulación	Prototipo IoT en Raspberry Pi con servidor web para apoyo a personas con discapacidad motriz.
ESPOL			
Bazán (2020)	ESPOL	Proyecto de titulación	IoT y WSN aplicados a iluminación inteligente en carreteras urbanas.
Villacís Morán (2023)	ESPOL	Proyecto de titulación	IoT para monitoreo de tensión y corriente en tableros industriales.
Ramírez Prado (2020)	ESPOL	Proyecto de titulación	Plataforma IoT para monitoreo de consumo eléctrico residencial.
UPS			
Delgado Tapia & Valencia (2021)	UPS	Proyecto de titulación	Prototipo IoT para monitoreo remoto de calidad del agua en estanques de tilapia.
Sánchez Macías & Baldeon (2022)	UPS	Proyecto de titulación	Supervisión de energía con IoT para mantenimiento en una planta industrial.
García Álvarez & López Solano (2024)	UPS	Proyecto de titulación	IoT para monitoreo ambiental y radiación solar con energía renovable.
Universidad de Guayaquil			
Meza Mora & Del Rosario (2021)	UG	Proyecto de titulación	Red IoT basada en LoRaWAN para control de acceso y proyectos académicos.
UNL			
Castillo Yaguana (2024)	UNL	Proyecto de titulación	Sistema IoT para monitoreo de variables eléctricas en gestión energética.
UTC			

Zambrano Cajas (2023)	UTC	Proyecto de titulación	Prototipo IoT para monitoreo de parámetros de variadores de frecuencia en tiempo real.
ESPOCH			
López Freire (2023)	ESPOCH	Proyecto de titulación	Prototipo IoT para monitoreo y control del consumo energético residencial.

Notas. Fuente: Elaboración propia.

En muchos entornos educativos y productivos, la falta de prototipos de IoT funcionales para el monitoreo eléctrico limita la formación práctica de estudiantes y técnicos, lo que reduce la preparación de profesionales capaces de diseñar, implementar y mantener soluciones inteligentes en sistemas eléctricos modernos.

En resumen, las problemáticas identificadas ponen de manifiesto que la falta de herramientas inteligentes para el análisis de sistemas eléctricos dificulta la gestión de la energía y limita la innovación y el desarrollo de capacidades técnicas en entornos educativos y productivos. Esta situación resalta la importancia de diseñar soluciones tecnológicas que integren el seguimiento, el análisis y la visualización de datos, fomentando un enfoque proactivo en la prevención de fallos y en la optimización de los recursos eléctricos.

1.6 Alcance del proyecto

El presente proyecto de titulación tiene como objetivo implementar un prototipo de red IoT para analizar la corriente y el voltaje en sistemas de distribución de baja tensión mediante tecnologías de sensado y comunicación inalámbrica. La iniciativa se desarrollará en el laboratorio de telecomunicaciones de la Universidad Estatal Península de Santa Elena (UPSE) y se utilizará un tablero de control que dispone de neutro, fase 1, fase 2 y tierra. En este entorno se instalarán y configurarán un sensor de corriente y otro de voltaje, que serán esenciales para la adquisición de

datos eléctricos de las líneas del sistema y permitirán detectar fugas de corriente o anomalías en las mediciones.

La implementación del prototipo incluirá el uso de un microcontrolador ESP32, encargado del procesamiento inicial de los datos, y de un módulo CubeCell AB02 Plus, que transmitirá la información mediante el protocolo LoRaWAN. Esta comunicación será recibida por un gateway SenseCAP M2, que se integrará con la plataforma The Things Network (TTN) y permitirá enlazar los datos con un servidor de visualización en Node-RED a través de MQTT. Este esquema garantizará la conectividad IoT del sistema, lo que facilitará la supervisión remota y la integración con aplicaciones de análisis en tiempo real.

También se desarrollará un panel de control en Node-RED para presentar las variables medidas, como los voltajes y las corrientes RMS, en cada línea del tablero. Este panel permitirá identificar condiciones anómalas, como corrientes residuales en el neutro o presencia de corriente en el conductor de tierra, que suponen un riesgo para la instalación eléctrica. El objetivo es proporcionar un sistema de monitorización en tiempo real y contribuir a la prevención de fallos eléctricos.

El proyecto incluye la validación experimental del prototipo en el laboratorio para comprobar la precisión de las mediciones y la fiabilidad de la transmisión de datos a la nube. Se realizarán pruebas controladas que incluyan cargas simuladas y condiciones de fuga para verificar la respuesta del sistema en distintos escenarios. En esta fase, la implementación del prototipo se limitará únicamente al ámbito académico y de experimentación, sin extenderse a redes de distribución reales ni entornos industriales.

Finalmente, el prototipo podrá servir de base para prácticas de laboratorio destinadas a la formación en tecnologías IoT aplicadas a la energía eléctrica. De este modo, los estudiantes podrán adquirir experiencia práctica en el diseño de sistemas de monitorización, el manejo de protocolos de comunicación inalámbrica y la integración de paneles de control en la nube, lo que fortalecerá sus competencias en innovación tecnológica en el área de telecomunicaciones y sistemas eléctricos.

1.7 Metodología

La metodología adoptada en este proyecto combina enfoques de investigación descriptiva, documental y aplicada para garantizar un proceso sistemático, riguroso y replicable en el diseño e implementación de un prototipo de red IoT. Esta combinación permite analizar y documentar las características de los sensores de corriente y voltaje, los microcontroladores ESP32 y los módulos de comunicación LoRaWAN, así como validar experimentalmente la adquisición, transmisión y visualización de datos eléctricos en un tablero de distribución de baja tensión. De esta manera, no solo se garantiza la precisión y la fiabilidad del prototipo, sino también la generación de un modelo replicable que pueda utilizarse en entornos académicos para la formación práctica en tecnologías de IoT aplicadas a sistemas eléctricos.

1.7.1 Investigación descriptiva

En este proyecto se describe cómo las tecnologías del Internet de las cosas (IoT) han permitido innovar en el monitoreo de sistemas eléctricos de baja tensión, lo que ha abierto oportunidades para el análisis y la detección de anomalías en entornos controlados. La investigación se centró en detallar las características de los sensores de corriente y voltaje, el

funcionamiento de los microcontroladores, como el ESP32, y su integración con módulos LoRaWAN, así como en la presentación de datos en plataformas de visualización en la nube [1].

1.7.2 Investigación documental

Se revisaron trabajos previos, artículos científicos, manuales técnicos y tesis relacionados con la aplicación de la IoT en la medición de variables eléctricas. Esta revisión permitió conocer los métodos utilizados para integrar sensores, microcontroladores y gateways, así como las soluciones de comunicación inalámbrica empleadas en proyectos similares. La recopilación de esta información fue esencial para fundamentar teóricamente la implementación del prototipo y definir los parámetros de diseño y validación [7].

1.7.3 Investigación aplicada

El trabajo propuesto se centró en el diseño, la configuración de un prototipo de monitorización eléctrica en el laboratorio de telecomunicaciones de la UPSE. Para ello, se implementó un sistema de adquisición de datos basado en sensores de corriente y voltaje conectados a un microcontrolador ESP32 que procesa las señales y las transmitirá mediante un módulo CubeCell AB02 Plus bajo el protocolo LoRaWAN a un gateway SenseCAP M2. Posteriormente, los datos se enlazaron a la nube a través de MQTT, lo que permitió su visualización en un panel de control en Node-RED. Finalmente, se realizaron pruebas experimentales en el tablero de control para verificar el funcionamiento del sistema y validar su capacidad para detectar fugas o condiciones anómalas en las mediciones.

1.8 Marco contextual

Este proyecto tiene como objetivo contribuir al desarrollo tecnológico y académico de la provincia de Santa Elena mediante la aplicación de las telecomunicaciones en sistemas eléctricos de baja tensión. La integración de tecnologías emergentes, como el Internet de las cosas, es un elemento clave que debe gestionarse de manera estratégica para optimizar los procesos de supervisión, control y seguridad en los espacios universitarios. Para lograr un avance efectivo, es esencial analizar las condiciones actuales de la infraestructura tecnológica, comprender las limitaciones existentes y aprovechar las oportunidades que ofrecen las nuevas herramientas de conectividad.

En el primer capítulo de este proyecto se explica el contexto que sustenta la elección de la tecnología. En este análisis se tiene en cuenta la infraestructura disponible en el laboratorio de telecomunicaciones, donde se implementará la red IoT destinada a medir la corriente y el voltaje en un tablero de baja tensión. También se compararán las características de los sistemas convencionales de monitorización eléctrica y se destacará la necesidad de incorporar soluciones inalámbricas y plataformas en la nube que permitan un control más eficiente y seguro.

Además, se analizarán distintas opciones tecnológicas para la transmisión de datos, como los protocolos LoRaWAN y MQTT, que resultan viables por su bajo costo, su amplio alcance y su facilidad de integración con plataformas de visualización, como Node-RED. Este proyecto permitirá poner de manifiesto las ventajas de aplicar tecnologías IoT en el campo de las telecomunicaciones, fomentando la innovación educativa y el desarrollo de habilidades técnicas que potencien la formación profesional en la región.

Capítulo 2

2.1 Marco conceptual

El presente proyecto se centra en el análisis y la supervisión de variables eléctricas esenciales en sistemas de distribución de baja tensión, como la corriente y el voltaje, cuya monitorización es fundamental para garantizar la seguridad, la eficiencia energética y la continuidad del servicio. En redes de baja tensión, las variaciones inesperadas, las sobrecargas y las fugas eléctricas pueden ocasionar pérdidas de energía, fallos en los equipos y riesgos para las instalaciones, por lo que el uso de sistemas de medición y monitorización continuos se vuelve una necesidad técnica y operativa.

No obstante, la contribución principal del proyecto se enmarca en el ámbito de las telecomunicaciones, ya que integra tecnologías de transmisión inalámbrica orientadas al Internet de las cosas (IoT). El sistema incorpora sensores eléctricos conectados a un microcontrolador ESP32 que digitaliza las señales obtenidas y las transmite mediante redes de baja potencia y largo alcance (LPWAN), utilizando el estándar LoRaWAN, muy empleado en sistemas de telemetría distribuidos.

El enlace de comunicación se establece entre los nodos sensores y un gateway LoRaWAN SenseCAP M2, que actúa como punto de acceso a la red de telecomunicaciones. Este gateway reenvía los datos a plataformas en la nube, como The Things Network, donde se procesan y visualizan posteriormente mediante herramientas como Node-RED, empleando protocolos de comunicación propios del área de las telecomunicaciones, como MQTT, HTTP/HTTPS y webhooks.

Este marco conceptual integra la parte eléctrica del monitoreo con los principios esenciales de las telecomunicaciones, lo que permite la recolección, transmisión y análisis remoto de información en tiempo real. De este modo, se establece una base técnica coherente con el perfil profesional de la carrera, que destaca la importancia del procesamiento y transporte de datos en una arquitectura IoT moderna.

2.2 Marco teórico

2.2.1 Monitoreo de corriente y voltaje en baja tensión

El monitoreo de la corriente y el voltaje en sistemas de baja tensión es fundamental para garantizar la eficiencia energética, la continuidad operativa y la seguridad de las instalaciones eléctricas. En una red de distribución, la corriente representa el flujo de carga eléctrica que circula por los conductores, mientras que el voltaje corresponde a la diferencia de potencial que impulsa dicho flujo. Estas variables determinan el comportamiento de la red y permiten evaluar en tiempo real el estado del suministro eléctrico. Si no se supervisan adecuadamente, pueden producirse condiciones anómalas que deriven en fallos, pérdidas económicas o riesgos para los usuarios.

Las redes de baja tensión están expuestas a variaciones dinámicas provocadas por cargas conectadas, fluctuaciones en el consumo, deterioro de las instalaciones o fallos en los equipos. Entre las anomalías más comunes se encuentran las fugas de corriente, generadas por un aislamiento deteriorado o defectos en las conexiones; las sobrecargas, que elevan la temperatura de los conductores y pueden causar incendios; las caídas de tensión, que afectan al rendimiento de los dispositivos sensibles, y las fallas a tierra, que comprometen la integridad del sistema y la seguridad del usuario. Estas condiciones suelen pasar desapercibidas cuando las mediciones se realizan de manera manual o esporádica, lo que dificulta la detección temprana de eventos críticos.

La ausencia de un sistema de monitorización continua también limita la capacidad para identificar patrones de comportamiento eléctrico, como picos de consumo en horarios específicos, cargas desequilibradas o variaciones anormales en circuitos particulares. En instalaciones residenciales o educativas, estas situaciones pueden provocar cortes de suministro inesperados, el deterioro prematuro de los equipos y un aumento del riesgo eléctrico. Por ello, es esencial supervisar constantemente la corriente y el voltaje para mantener la estabilidad operativa y reducir los problemas asociados a una gestión energética deficiente.

La incorporación de tecnologías IoT en este contexto permite transformar el proceso de monitorización tradicional en un sistema inteligente capaz de registrar, transmitir y analizar datos en tiempo real. Gracias a sensores adecuados y redes de comunicación de largo alcance, como LoRaWAN, es posible obtener información precisa de diferentes puntos de la instalación sin necesidad de intervención manual. Este enfoque facilita la visualización remota de variables eléctricas, la generación de alertas instantáneas ante anomalías y la implementación de estrategias de mantenimiento predictivo. Además, el análisis de datos históricos permite evaluar tendencias de consumo, optimizar la distribución de cargas y tomar decisiones informadas para mejorar la eficiencia energética.

En conjunto, el monitoreo de la corriente y el voltaje mediante tecnologías IoT ofrece una solución moderna, segura y escalable para la gestión de redes eléctricas de baja tensión. Su implementación contribuye a reducir riesgos, evitar averías imprevistas, mejorar la calidad del suministro y garantizar el correcto funcionamiento de los sistemas eléctricos, por lo que resulta una herramienta imprescindible en entornos donde la continuidad del servicio y la protección de los usuarios son prioritarias.

Tabla 2*Anomalías comunes en sistemas de baja tensión*

Anomalía eléctrica	Causas comunes	Efectos en la instalación	Riesgos asociados
Fuga de corriente	Aislamiento deteriorado, humedad, defectos en conexiones	Incremento del consumo, pérdida de eficiencia.	Riesgo de electrocución, incendios por derivaciones.
Sobrecarga	Exceso de carga conectada, conductores subdimensionados.	Calentamiento de cables, caída de voltaje.	Incendios, daño en equipos, interrupciones del servicio.
Caída de voltaje	Picos de demanda, cables largos o en mal estado.	Mal funcionamiento de equipos sensibles.	Apagado inesperado, reducción de vida útil de dispositivos
Falla a tierra	Conexiones defectuosas, humedad, equipos dañados.	Activación de protecciones, interrupciones.	Riesgo alto para personas, quemado de equipos.

Notas. Fuente: Elaboración propia.

La tabla 2 muestra de forma estructurada las principales anomalías que pueden presentarse en una red de baja tensión, lo que facilita la comprensión de los riesgos y justifica la necesidad de un sistema de monitorización que permita detectarlas a tiempo mediante tecnologías IoT.

Tabla 3*Medición manual vs medición IoT*

Criterio	Medición tradicional	Medición IoT en tiempo real
Frecuencia de medición	Ocasional o por inspección	Continua 24/7
Detección de fallas	Tardía o reactiva	Temprana y preventiva
Almacenamiento de datos	Limitado o inexistente	Histórico y analítico
Nivel de riesgo	Alto por desconocimiento del estado real	Bajo por alertas inmediatas
Eficiencia energética	No evaluada en tiempo real	Optimizable según consumo

Notas. Fuente: Elaboración propia.

La comparación pone de manifiesto las limitaciones de los métodos tradicionales frente a los sistemas IoT, y muestra cómo la supervisión remota permite mejorar la detección de fallos, reducir los riesgos operativos y optimizar la gestión energética de manera más eficiente.

2.2.2 Sensores para la medición de variables eléctricas

La implementación de sistemas de monitorización en baja tensión requiere sensores que puedan capturar con precisión las magnitudes eléctricas. Estos dispositivos constituyen la capa de percepción dentro de la arquitectura IoT, ya que transforman señales analógicas de alta energía en valores digitalizables que pueden transmitirse y analizarse posteriormente. La correcta selección y calibración de los sensores es fundamental para garantizar la fiabilidad del sistema.

2.2.2.1 *Sensores de corriente*

Para medir corriente alterna hay múltiples tecnologías disponibles, pero los sistemas de monitorización energética en baja tensión emplean principalmente transformadores de corriente no invasivos, sensores basados en el efecto Hall y transformadores de precisión miniatura. Entre los dispositivos más utilizados destacan el SCT-013, el ACS712 y el ZMCT103C, cada uno con características particulares que determinan su aplicabilidad.

El SCT-013 es un transformador de corriente del tipo pinza que funciona según el principio de inducción magnética. Gracias a su diseño no invasivo, permite medir corrientes de entre 20 y 100 amperios sin interrumpir el conductor, por lo que es una alternativa segura para instalaciones residenciales y comerciales. Además, su aislamiento galvánico protege el sistema IoT de voltajes peligrosos, lo que reduce el riesgo para la electrónica de control.

Por otro lado, el ACS712 es un sensor basado en el efecto Hall que puede medir corriente alterna y continua, lo que supone una ventaja frente a los transformadores inductivos. Sin embargo,

este módulo no tiene aislamiento galvánico, por lo que la línea medida está conectada eléctricamente al circuito electrónico. Este aspecto supone un riesgo en sistemas conectados a la red de corriente alterna y limita su uso en instalaciones residenciales. A pesar de ello, su buena linealidad y resolución lo han convertido en un sensor muy utilizado en proyectos educativos y prototipos de bajo consumo.

Por último, el ZMCT103C es un transformador de corriente en miniatura diseñado para medir corrientes pequeñas (entre 0 y 5 amperios). Su reducido tamaño y su aislamiento intrínseco lo hacen adecuado para dispositivos, electrodomésticos y cargas de baja potencia. No obstante, su rango limitado impide su aplicación en mediciones de corrientes generales de una vivienda o cuadro eléctrico, donde los consumos suelen superar los 10 A. Estudios realizados en la Universidad de Cuenca señalan que este sensor tiene un buen rendimiento en cargas pequeñas, pero pierde precisión cuando se somete a corrientes superiores a su rango nominal.

Tabla 4

Comparativa de sensores de corriente

Sensor de corriente	Tipo / Principio	Rango de medición	Aislamiento Galvánico	Precisión	Ventajas	Desventajas
SCT-013 (Clamp CT)	Transformador de corriente por inducción.	20 A – 100 A AC (según modelo).	Sí.	$\pm 2-3 \%$	No invasivo, seguro, económico, compatible con IoT, buena estabilidad.	Requiere calibración y resistencia a burden.
ACS712 (Hall effect)	Sensor Hall para AC/DC.	5 A, 20 A, 30 A AC/DC.	No (conexión directa).	$\pm 1.5 \%$	Mide AC y DC, buena resolución, fácil integración.	Riesgo al no tener aislamiento, limitado para corrientes altas, sensible a ruido.

ZMCT103C (Mini CT)	Transformador de corriente.	0–5 A AC	Sí	$\pm 3 \%$	Ideal para corrientes pequeñas.	No apto para corrientes altas, requiere amplificación.
-------------------------------	-----------------------------	----------	----	------------	---------------------------------	--

Notas. Fuente: Elaboración propia.

En conjunto, los análisis y la tabla 4 demuestran que el SCT-013 ofrece la mejor relación entre seguridad, precisión y aplicabilidad para sistemas residenciales, por lo que es la opción más adecuada para este proyecto.

2.2.2.2 *Sensores de voltaje*

En cuanto a la medición de voltaje alterno, existen diversas estrategias que incluyen divisores resistivos, transformadores de voltaje y módulos de medición especializados. Entre los sensores evaluados se encuentran el ZMPT101B, el método del divisor resistivo y el chip HLW8012, cada uno con características particulares.

El ZMPT101B es un módulo basado en un transformador de voltaje de alta precisión acompañado de un amplificador operacional que permite ajustar la ganancia mediante un potenciómetro. Este diseño proporciona aislamiento eléctrico entre la red de corriente alterna y el microcontrolador, lo que es esencial para proteger la electrónica en aplicaciones de Internet de las cosas.

Otro método común es el divisor resistivo, que se utiliza ampliamente por su bajo costo y simplicidad. Sin embargo, este sistema no proporciona aislamiento galvánico, por lo que el microcontrolador queda expuesto a la red eléctrica. Además, su precisión puede verse afectada por variaciones térmicas y por el ruido inherente a la línea de corriente alterna. Debido a estas limitaciones, el divisor resistivo se utiliza únicamente en prototipos o simulaciones, pero no se

recomienda para sistemas que estarán en contacto directo con la red eléctrica residencial o comercial.

El tercer dispositivo evaluado es el HLW8012, un chip especializado que permite medir simultáneamente el voltaje, la corriente y la potencia activa, lo que resulta atractivo para los sistemas de gestión energética. Este componente es preciso y estable, pero no incluye aislamiento interno, por lo que requiere un diseño de protección y filtrado más complejo. Según la literatura especializada, si bien es útil en medidores inteligentes y tarjetas comerciales, su integración en proyectos académicos exige mayores consideraciones de diseño y seguridad.

La comparación de estos mecanismos permite concluir que el ZMPT101B es el sensor más adecuado para instalaciones residenciales, ya que ofrece aislamiento, precisión y facilidad de integración, que son características esenciales en un sistema IoT seguro y fiable.

Tabla 5

Tipos de sensores de voltaje

Sensor de Voltaje	Tipo / Principio	Rango de Medición	Aislamiento Galvánico	Precisión	Ventajas	Desventajas
ZMPT101B	Transformador de voltaje + amplificador operacional.	0–250 V AC	Sí	$\pm 1-2 \%$	Alta precisión, bajo ruido, ajuste por potenciómetro, seguro.	Requiere calibración inicial.
Voltage Divider (Resistivo)	División de tensión.	Hasta 250 V AC/DC*	No	$\pm 5 \%$	Muy económico, fácil de implementar.	No posee aislamiento, riesgo eléctrico, menor precisión.
HLW8012	Chip especializado para	80–260 V AC	No	$\pm 1 \%$	Mide potencia, corriente y	No aislado, requiere

medición de energía.	voltaje, alta precisión.	diseño seguro.
-------------------------	-----------------------------	-------------------

Notas. Fuente: Elaboración propia.

El Sensor ZMPT101B fue seleccionado porque proporciona aislamiento galvánico completo, lo que evita que el microcontrolador esté expuesto a los 110-220 V de corriente alterna de la red eléctrica, a diferencia de los divisores resistivos, que carecen de aislamiento y suponen un mayor riesgo eléctrico.

Además, su bajo costo, la facilidad de calibración mediante potenciómetro y su compatibilidad con plataformas como ESP32 y LoRaWAN justifican su elección como componente principal para la medición de voltaje en este proyecto.

2.2.3 Internet de las cosas (IoT)

El Internet de las cosas (IoT, por sus siglas en inglés) es un enfoque tecnológico que facilita la conexión de elementos físicos con el entorno digital, lo que permite la recopilación, el procesamiento y la transmisión de datos a través de redes de comunicación. Según IBM (2024), el IoT se define como una red integrada de dispositivos físicos dotados de sensores, software y capacidades de conectividad que les permiten interactuar entre sí y con otros sistemas a través de internet [15].

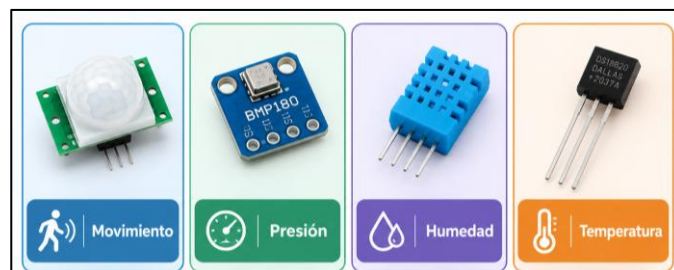


Figura 1. Sensores de IoT en todos sus ámbitos

El desarrollo progresivo de esta tecnología ha permitido transformar elementos informáticos desechados en sistemas interconectados con capacidad para comunicarse, ajustarse y reaccionar de manera autónoma, sin necesidad de intervención humana directa[16].

El principio operativo del Internet de las cosas (IoT) se estructura en cuatro etapas fundamentales:

- Sensado: recolección de datos provenientes del entorno físico mediante sensores o actuadores.
- Transmisión: transferencia de la información a través de protocolos de comunicación, ya sean alámbricos o inalámbricos.
- Procesamiento: interpretación y análisis de los datos en servidores locales o plataformas en la nube.
- Aplicación: representación visual de la información y ejecución de medidas de control o corrección.

En el contexto de los sistemas eléctricos, el IoT permite supervisar de manera continua parámetros como la corriente, el voltaje y la potencia, lo que permite identificar fallos de manera anticipada, optimizar el uso energético y fortalecer la seguridad operativa. Esta capacidad de monitorización y gestión remota es un recurso estratégico para la modernización de las instalaciones eléctricas de las redes de baja tensión.

2.2.4 Arquitectura de una red IoT

La arquitectura del Internet de las cosas representa la estructura fundamental que determina la forma en que los dispositivos se conectan entre sí, cómo se transmite la información y cómo se procesa para obtener datos significativos. Su objetivo principal es garantizar una comunicación eficaz entre los componentes del sistema, respetando los principios de interoperabilidad, escalabilidad y seguridad.

En términos generales, la arquitectura del Internet de las cosas (IoT) se estructura en tres niveles funcionales: percepción, red y aplicación [17].

- Nivel de percepción: corresponde al proceso de captura de datos del entorno físico mediante sensores, transductores y actuadores. Estos dispositivos transforman variables analógicas, como la corriente, el voltaje, la temperatura o la humedad, en señales digitales que pueden ser interpretadas por microcontroladores o sistemas embebidos.
- Nivel de red: cumple la función de interconectar la capa de percepción con la capa de aplicación. En esta etapa intervienen protocolos de comunicación y dispositivos de enlace, como gateways y routers, que garantizan una transmisión de datos eficiente y segura mediante tecnologías como Wi-Fi, Zigbee, LoRaWAN o NB-IoT.
- Nivel de aplicación: se encarga de la representación, el análisis y la gestión de la información recolectada. En este nivel se emplean plataformas de procesamiento y visualización, como Node-RED o The Things Network (TTN), que permiten controlar y administrar los datos generados por el sistema.

2.2.5 Tecnología de comunicación LPWAN

Las tecnologías de comunicación inalámbrica conocidas como redes de área amplia de baja potencia (LPWAN, por sus siglas en inglés) están orientadas a la conexión de dispositivos distribuidos que requieren un consumo energético mínimo, una cobertura extensa y el envío de datos en volúmenes reducidos. Este tipo de redes es especialmente adecuado en escenarios donde las soluciones convencionales, como el Wi-Fi o las redes celulares, presentan restricciones en términos de alcance o eficiencia energética [18].

Entre las tecnologías LPWAN más relevantes se encuentran LoRaWAN, Sigfox, NB-IoT y LTE-M. Cada una de ellas tiene características particulares que las hacen adecuadas para distintos tipos de aplicaciones, en función del nivel de consumo energético tolerado y de los requisitos de infraestructura de comunicación.

2.2.5.1 *Sigfox*

Sigfox es una compañía francesa que ha desarrollado una tecnología LPWAN homónima basada en un sistema de comunicación tipo celular que emplea banda ultraestrecha (UNB) para conectar dispositivos remotos [18].

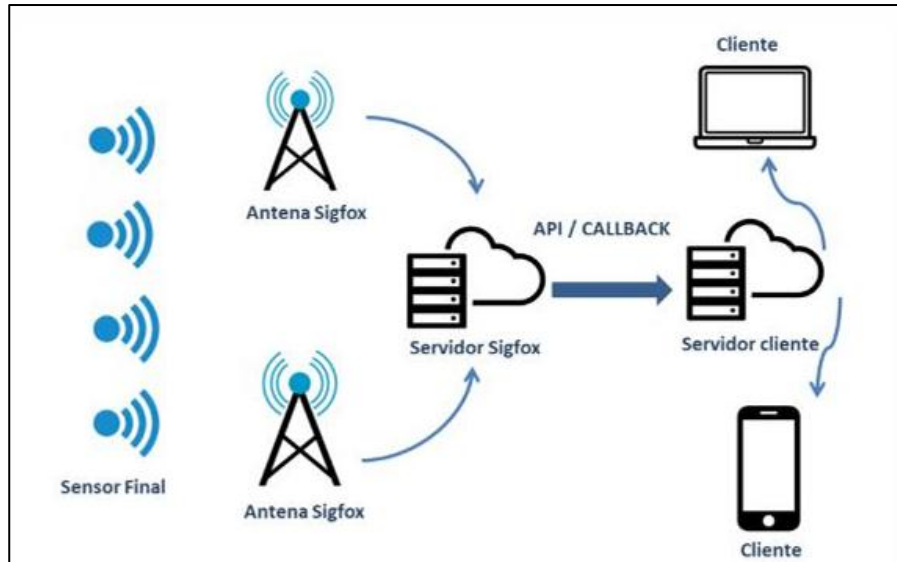


Figura 2. Arquitectura de Sigfox

2.2.5.2 LoRa

LoRa (Long Range) es el esquema de modulación en radiofrecuencia que opera como la capa física dentro de las redes LPWAN identificadas como LoRaWAN. Esta tecnología es propiedad de la empresa Semtech.

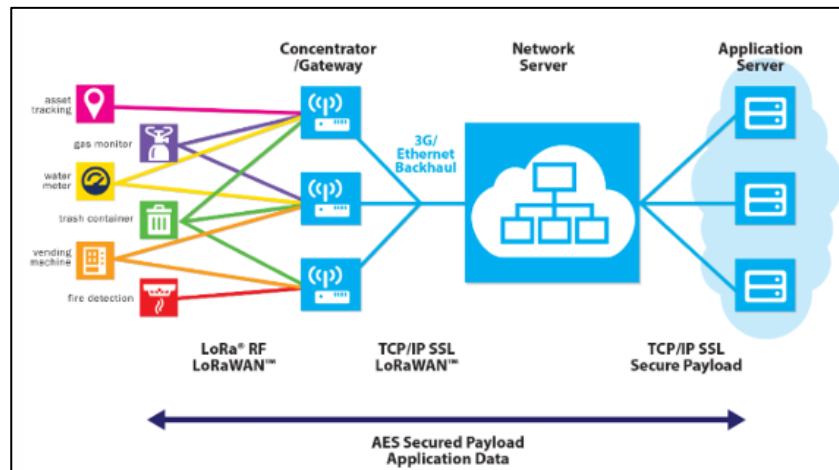


Figura 3. Arquitectura de una red LoRa

2.2.5.3 NB-IoT

NB-IoT (Narrowband Internet of Things) es una tecnología de comunicación de banda estrecha que opera en frecuencias licenciadas y puede coexistir con redes GSM y LTE. Fue

introducida por el 3GPP en la versión Release 13, lo que le permite soportar conexiones masivas sin competir por el espectro ni sufrir restricciones de ciclo de trabajo.

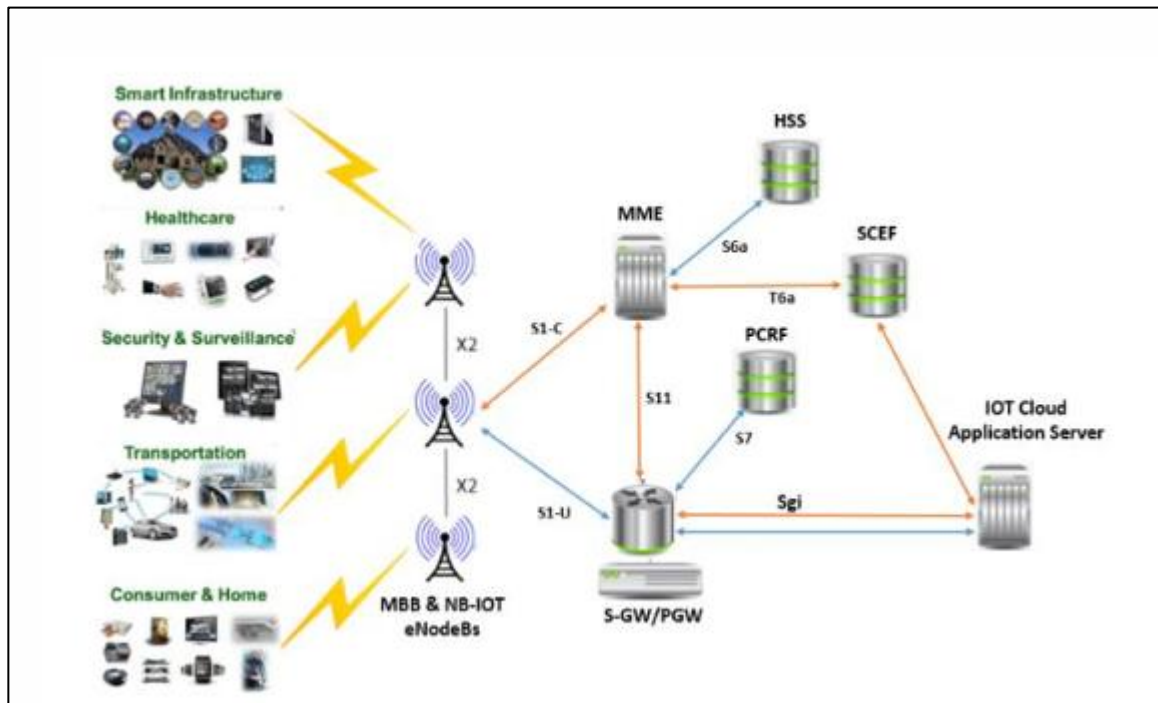


Figura 4. Arquitectura de red NB-IoT

2.2.5.4 LTE-M

LTE-M (Long Term Evolution for Machines) es una tecnología de comunicación móvil que utiliza la infraestructura existente de redes LTE, es decir, 4G. Su diseño facilita la integración en dispositivos previamente compatibles con las redes 2G, 3G y 4G, por lo que su implementación es más sencilla. Actualmente, LTE-M permite la transmisión de datos en diversos formatos, como imágenes, texto y voz.

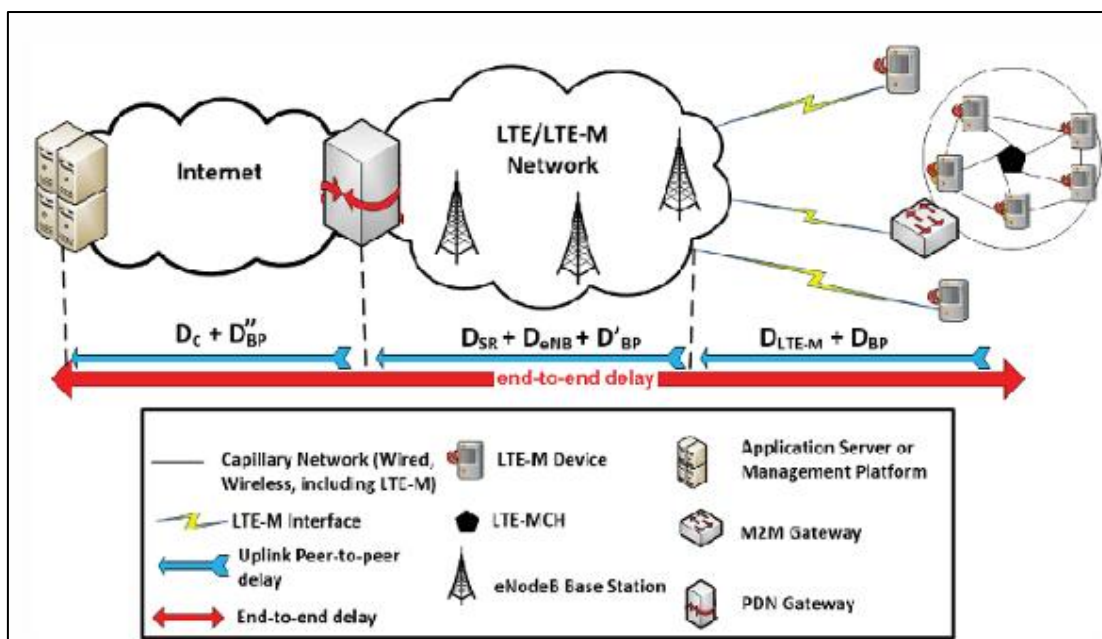


Figura 5. Arquitectura de una red LTE-M

Tabla 6

Tipos de tecnologías LPWAN

Parámetro	LoRaWAN	Sigfox	NB-IoT	LTE-M
Tipo de red	Privada o pública, basada en espectro sin licencia (ISM).	Red pública, propiedad del operador Sigfox.	Red celular de banda licenciada LTE.	Red celular LTE-M (Machine Type Communication).
Banda de frecuencia	433(Asia) / 868(EUROPA) / 915 (EEUU) MHz.	868(EUROPA) / 902(EEUU) MHz.	Bandas licenciadas del espectro LTE (700 – 2100 MHz).	Bandas LTE licenciadas (700 – 2100 MHz).
Velocidad de transmisión	0.3 – 50 kbps.	100 bps.	26 – 250 kbps.	200 – 1000 kbps.
Alcance aproximado	2 – 15 km en entorno urbano o rural.	10 – 50 km según topografía.	1 – 10 km.	1 – 10 km.
Consumo energético	Muy bajo, ideal para dispositivos alimentados por batería.	Muy bajo, duración de batería de hasta 10 años.	Bajo, pero superior a LoRaWAN.	Medio, mayor consumo por mayor tasa de datos.
Costo de implementación	Bajo; permite redes privadas sin dependencia de operadores.	Medio; requiere suscripción a red Sigfox.	Alto; depende de operadores celulares.	Alto; requiere infraestructura LTE existente.

Ventajas principales	Largo alcance, bajo costo, fácil escalabilidad, soporte en TTN.	Cobertura extensa, bajo consumo, pública disponible.	Alta confiabilidad, cobertura garantizada, soporte de operadores.	Mayor velocidad, movilidad y soporte de voz.
Desventajas	Baja tasa de datos, posible interferencia en bandas ISM.	Muy baja capacidad de datos, dependencia del operador.	Mayor costo e infraestructura compleja.	Alto consumo y complejidad técnica.
Aplicaciones típicas	Monitoreo eléctrico, agricultura inteligente, ciudades inteligentes.	Sensores ambientales, localización, telemetría básica.	Medición industrial, servicios públicos, infraestructura crítica.	Vehículos conectados, telemetría móvil, logística avanzada.

Notas. Fuente: Elaboración propia.

Entre las distintas tecnologías LPWAN, LoRaWAN destaca por ofrecer una combinación óptima de bajo consumo energético, amplia cobertura y simplicidad en su implementación. Estas características la convierten en una solución especialmente adecuada para aplicaciones de IoT orientadas al monitoreo de sistemas eléctricos en entornos de baja tensión.

2.2.6 Tecnología LoRa y protocolo LoRaWAN

La tecnología LoRa (Long Range) emplea una técnica de modulación de espectro ensanchado denominada Chirp Spread Spectrum (CSS), que permite transmisiones resistentes a las interferencias y con un consumo energético reducido. Además, el protocolo LoRaWAN (Long Range Wide Area Network), promovido por la LoRa Alliance, establece la estructura de red y las reglas de comunicación necesarias para conectar dispositivos IoT distribuidos en amplias zonas geográficas [18].

En la figura 4 se muestra la arquitectura de la red LoRaWAN, que se compone de cuatro elementos principales:

- Dispositivos terminales (end devices): sensores o microcontroladores encargados de captar información del entorno y transmitirla a la red.
- Gateways: equipos intermediarios que reciben las señales LoRa provenientes de los nodos finales y las redirigen hacia la infraestructura de Internet.
- Servidor de red: se encarga de gestionar procesos como la autenticación de dispositivos, el enrutamiento de paquetes y la aplicación de medidas de seguridad.
- Servidor de aplicación: interpreta los datos recibidos y los presenta de forma comprensible al usuario final para facilitar la toma de decisiones.

Entre las ventajas de esta tecnología, destacan su funcionamiento en bandas de frecuencia no licenciadas, su capacidad de cobertura en zonas urbanas y rurales, su eficiencia energética y su seguridad de extremo a extremo. Estas características la convierten en una solución idónea para aplicaciones de monitorización en sectores como el energético, el agrícola y el industrial.

2.2.7 Protocolos de comunicación IoT

En el ámbito del Internet de las cosas (IoT), los protocolos de comunicación desempeñan un papel fundamental, ya que definen el mecanismo mediante el cual los dispositivos intercambian datos. La elección del protocolo más adecuado depende de factores como el consumo energético, el tipo de red utilizada y la cantidad de información que se transmite.

Uno de los protocolos más ampliamente adoptados es MQTT (Message Queuing Telemetry Transport), reconocido por su estructura ligera y su alto rendimiento. Este protocolo se basa en un esquema de publicación y suscripción, en el que los dispositivos emisores envían mensajes asociados a determinados temas (topics), que son gestionados por un intermediario (broker) encargado de distribuirlos a los receptores suscritos. Esta arquitectura simplifica el proceso de

comunicación y permite la conexión simultánea de numerosos dispositivos dentro de una red IoT [19].

Tabla 7

Tabla comparativa de los protocolos de comunicación IoT

Parámetro	MQTT	HTTP / HTTPS	CoAP	AMQP
Modelo de comunicación	Publicación / Suscripción (Bróker central).	Cliente / Servidor	Cliente / Servidor (UDP)	Publicación / Suscripción
Transporte base	TCP/IP	TCP/IP	UDP	TCP/IP
Tamaño del mensaje	Muy pequeño (2 bytes de encabezado).	Grande (encabezado extenso).	Pequeño	Mediano
Consumo energético	Muy bajo	Medio a alto	Muy bajo	Medio
Seguridad	TLS / SSL	SSL / TLS (en HTTPS)	DTLS	TLS / SASL
Fiabilidad	Alta (QoS configurable)	Media	Media	Muy alta
Complejidad de implementación	Baja	Baja	Media	Alta
Ventajas principales	Ideal para IoT, bajo consumo, persistencia de sesión.	Fácil integración, seguridad en HTTPS, compatible con web.	Ligero, eficiente, diseñado para sensores pequeños.	Alta seguridad, entrega garantizada, colas persistentes.
Desventajas	No apto para transmisión de archivos grandes.	Alto consumo, latencia mayor.	Menor fiabilidad por UDP.	Mayor carga computacional.
Aplicaciones típicas	Monitoreo remoto, telemetría, IoT industrial.	Servidores web, APIs IoT, dashboards.	Sensores de bajo consumo, redes LPWAN.	Mensajería empresarial, integración con bróker potentes.

Notas. Fuente: Elaboración propia.

El protocolo MQTT se ha seleccionado para este proyecto debido a su eficiencia en sistemas IoT que requieren transmitir datos de manera ligera, fiable y con bajo consumo

energético. A diferencia de HTTP/HTTPS, cuyos encabezados son pesados y requieren conexiones cliente-servidor constantes, MQTT opera con paquetes muy pequeños y bajo un modelo de publicación-suscripción, lo que reduce significativamente el uso de ancho de banda y la carga sobre el microcontrolador.

En comparación con CoAP, que utiliza UDP y puede presentar pérdidas de paquetes, MQTT ofrece mecanismos de calidad de servicio (QoS) que garantizan la entrega de los mensajes, un aspecto crucial para el monitoreo de variables eléctricas. Por último, aunque AMQP es robusto, su complejidad y carga computacional lo hacen poco adecuado para dispositivos de baja potencia, como el ESP32.

Por estas razones, MQTT es la opción más adecuada, ya que equilibra bajo consumo, fiabilidad y facilidad de integración con plataformas como TTN y Node-RED, y cumple los requisitos del sistema de monitorización eléctrica desarrollado.

2.2.8 Gateway

En una red LoRaWAN, el gateway actúa como punto de enlace entre los nodos sensores (dispositivos finales) y la plataforma o servidor de red. Su función principal es recibir los paquetes de radio difundidos por los nodos (modulación LoRa) y reenviarlos, mediante una conexión de red (Ethernet, Wi-Fi, 4G, etc.), al servidor de red, donde se procesan y gestionan. Desde el punto de vista de la arquitectura del sistema, el gateway pertenece a la capa de red, ya que se encarga de transportar los datos desde la capa de percepción hasta la capa de aplicación.

Funciones de un gateway LoRaWAN

1. **Recepción de señales LoRa:** escucha múltiples canales según el concentrador de radio (por ejemplo, 8 o más canales) para captar transmisiones de diferentes factores de difusión (SF) y anchuras de banda.
2. **Procesamiento básico:** demodulación, filtrado, generación de paquetes según protocolo Semtech Packet Forwarder o LoRa Basics Station.
3. **Backhaul (conexión hacia la red):** ofrece conectividad mediante ethernet, Wi-Fi, 4G/LTE, PoE, etc., para llevar los datos al servidor de red o al “network server”.
4. **Compatibilidad regional de frecuencia:** los gateways deben soportar los planes de frecuencia utilizados en la región (por ejemplo, EU868, US915, AS923, etc.).
5. **Gestión y seguridad:** suelen integrar interfaz de configuración, soporte para gestión remota, firmware actualizable, y mecanismos de seguridad para asegurar integridad y autenticación de datos.

El gateway es el nexo esencial que permite que los sensores de corriente y voltaje distribuidos envíen sus mediciones a la plataforma de visualización y manipulación de datos. La selección adecuada del gateway garantizará un alcance suficiente, compatibilidad de frecuencia con el país (en Ecuador, generalmente 868 MHz o 915 MHz, según la autorización local), baja latencia y fiabilidad en el transporte de datos.

Tabla 8

Equipos gateway LoRaWAN

Equipo	Frecuencia	Características
Kerlink Wirnet iBTS	868 MHz	Monitoreo remoto, Backhaul mono o dual-WAN
Heltec Wi-Fi LoRa 32	868 MHz	Fácil de instalar, amplia compatibilidad

SenseCAP M2	915 MHz	Robusto, fácil de instalar, amplia compatibilidad
-------------	---------	---

Notas. Fuente: Elaboración propia.

El gateway SenseCAP M2 se ha seleccionado como dispositivo central de comunicación debido a su alta estabilidad operativa, su facilidad de configuración y su compatibilidad nativa con redes LoRaWAN en la banda de 915 MHz, que se utiliza ampliamente en proyectos de IoT en la región. A diferencia de otros gateways, como los modelos Kerlink o Heltec, que requieren configuraciones más avanzadas o están orientados a redes privadas específicas, el SenseCAP M2 ofrece una instalación más intuitiva, soporte directo para The Things Network y una arquitectura robusta diseñada para funcionar de manera continua. Su capacidad para manejar múltiples canales LoRa simultáneamente y mantener una comunicación fiable con plataformas en la nube lo convierte en una opción ideal para sistemas de monitorización eléctrica, en los que la estabilidad del enlace y la integridad de los datos son factores críticos para el funcionamiento del sistema.

2.2.9 Plataforma IoT y visualización de datos

Las plataformas IoT son el componente esencial para la gestión, el procesamiento y la visualización de la información procedente de los dispositivos conectados. Estas plataformas actúan como intermediarias entre el hardware físico (sensores, microcontroladores y pasarelas) y el usuario final, y permiten que los datos recopilados se almacenen, procesen y representen de manera comprensible [20].

En una arquitectura IoT, la plataforma se ubica en la capa de aplicación y es responsable de administrar la información y facilitar la interacción con los dispositivos. Su función es garantizar que los datos obtenidos de la red se puedan transformar en información útil para la toma de decisiones en tiempo real.

2.2.9.1 *The Things Network (TTN)*

The Things Network es una plataforma abierta y colaborativa basada en el protocolo LoRaWAN que permite desplegar y gestionar redes de sensores de forma gratuita. Su objetivo principal es proporcionar una infraestructura global para el Internet de las cosas, de modo que cualquier usuario pueda conectar sus dispositivos y compartir cobertura con otros miembros de la comunidad.

TTN proporciona una arquitectura compuesta por pasarelas, servidores de red y servidores de aplicaciones que garantizan la transmisión segura de los datos desde los nodos sensores hasta la nube. Entre sus características más destacadas se encuentran:

- Compatibilidad con múltiples regiones de frecuencia (EU868, US915, AS923, etc.).
- Integración con los protocolos MQTT, HTTP y webhooks, que facilitan el envío de datos a aplicaciones externas.
- Gestión de dispositivos mediante un panel intuitivo y autenticación por claves (appEUI, devEUI y appkey).
- Acceso gratuito y comunidad activa que contribuye al mantenimiento y expansión de la red.

Su uso es especialmente ventajoso en proyectos académicos o prototipos, ya que no requiere infraestructura privada, permite la interconexión de gateways personales y posibilita el almacenamiento temporal de datos para pruebas y demostraciones.

2.2.9.2 *Node - RED*

Node-RED es una herramienta de desarrollo de código abierto creada por IBM que se basa en programación por flujos de datos (Flow-based programming). Su entorno gráfico permite conectar nodos con funciones, entradas o salidas, lo que facilita la integración entre dispositivos IoT, bases de datos y servicios web.

Node-RED funciona con JavaScript (Node.js) y cuenta con una interfaz gráfica en el navegador que permite crear flujos sin necesidad de realizar una codificación compleja. Entre sus principales características, destacan:

- Interfaz visual que permite diseñar flujos de datos mediante nodos conectados.
- Compatibilidad con MQTT, HTTP, TCP/IP y APIs REST, lo que lo convierte en una herramienta versátil.
- Creación de paneles de control personalizados con indicadores, medidores, gráficas y paneles de control.
- Gestión en tiempo real de variables medidas, como corriente, voltaje y potencia.
- Capacidad de automatización que permite ejecutar acciones ante eventos definidos (por ejemplo, activar una alerta cuando el voltaje cae por debajo de un valor umbral).

Node-RED se utiliza ampliamente en la implementación de sistemas eléctricos inteligentes, ya que permite visualizar información en tiempo real, generar historiales de datos y automatizar respuestas ante condiciones críticas, todo ello sin necesidad de una infraestructura costosa.

Capítulo 3

Este capítulo describe la evolución de la propuesta presentada, cuyo propósito es implementar un sistema de Internet de las cosas (IoT) para supervisar las variables eléctricas en un sistema de distribución de baja tensión. En esta sección se describen los componentes de la solución, tanto de hardware como de software, así como su arquitectura y diseño.

También se explica cómo se integran los diversos elementos que permiten la adquisición, transmisión, procesamiento y visualización de los datos. Para ello, se emplea la tecnología LoRaWAN para la comunicación y plataformas IoT para la gestión de la información. Asimismo, se aborda la arquitectura del sistema y su funcionamiento, y se deben tener en cuenta las consideraciones necesarias para su implementación.

El propósito de este capítulo es presentar de manera organizada el proceso de desarrollo del prototipo, destacando la relación entre los componentes que se van a utilizar y el cumplimiento de los objetivos planteados en la investigación.

3.1 Arquitectura y conexión de los componentes del sistema IoT propuesto

3.1.1 Arquitectura del sistema

La arquitectura del sistema de monitorización de variables eléctricas, que se muestra en la figura 6, se estructura en niveles funcionales de flujo de información. Se distinguen cuatro niveles principales: percepción, transmisión, middleware y presentación.

3.1.1.1 Capa de percepción

El sistema se inicia cuando el microcontrolador ESP32 recibe señales eléctricas del entorno, provenientes de los sensores. Este microcontrolador las procesa y prepara para su transmisión. Esta capa cumple la función de adquisición de datos dentro del sistema, ya que identifica las variables físicas necesarias para el análisis.

3.1.1.2 Capa de transmisión

El nodo CubeCell AB02 envía de inmediato los datos procesados al gateway SenseCAP M2 mediante comunicación LoRaWAN, lo que garantiza que la información se transmita de manera fiable y con un amplio alcance entre el punto de medición y la infraestructura IoT.

3.1.1.3 Capa de middleware

La plataforma TTN se encarga de recibir y gestionar los paquetes de datos y de dirigirlos hacia Node-RED. En este punto se implementan las normas para el procesamiento de datos y se prepara la información para su análisis y almacenamiento. Esta capa funciona como la unidad de procesamiento, es decir, como el cerebro del sistema, que analiza la señal recibida.

3.1.1.4 Capa de presentación

La información procesada se muestra en tiempo real mediante la aplicación Node-RED y se registra en una base de datos, lo que permite hacer un seguimiento constante, realizar un análisis histórico y detectar posibles irregularidades en el sistema eléctrico.

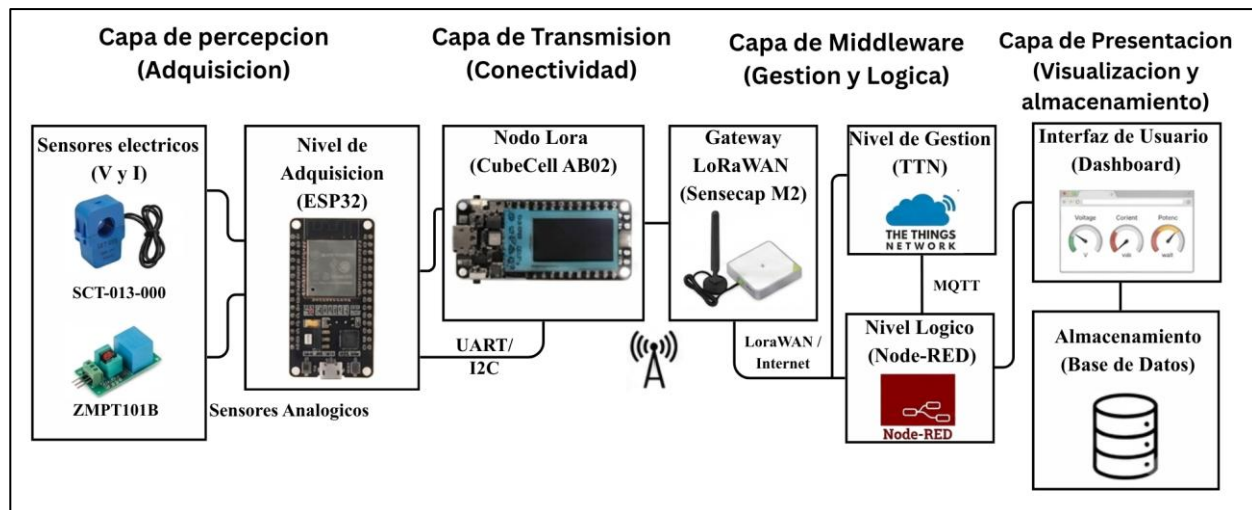


Figura 6. Diagrama del sistema de monitorización energética

La figura 7 muestra el escenario de implementación física y la distribución espacial de los componentes en el entorno de pruebas. A diferencia del modelo lógico, este diagrama representa la instrumentación del sistema.

- **Acoplamiento eléctrico:** muestra las conexiones directas al tablero de distribución.
- **Carga monitorizada:** vincula el prototipo de medición con el aparato de laboratorio objeto de estudio.
- **Infraestructura de hardware:** ubica los módulos físicos (CubeCell AB02 y gateway SenseCAP M2) dentro del espacio de la red local.

- **Puntos de consumo:** define los dispositivos finales (móvil y portátil) que utiliza el operador para recibir alertas y visualizar variables.

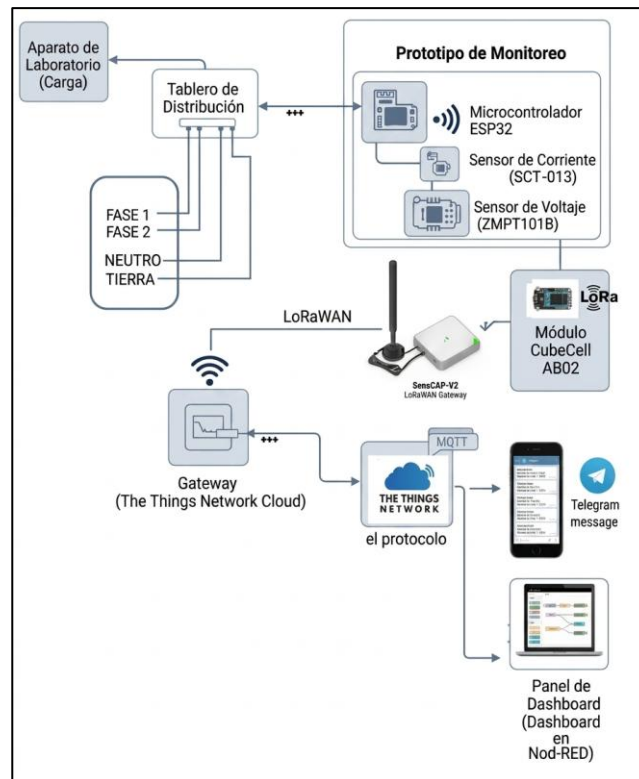


Figura 7. Escenario topológico del sistema

3.2 Elementos de hardware para la adquisición y transmisión de datos.

El sistema propuesto requiere distintos componentes de hardware que permiten adquirir, procesar y transmitir las variables eléctricas a la red IoT. A continuación, se detallan los elementos principales.

3.2.1 Microcontrolador ESP32

El ESP32, mostrado en la figura 8, es el microcontrolador que se encarga de controlar la adquisición y el procesamiento de las señales de corriente y de voltaje provenientes de los sensores del prototipo mediante algoritmos implementados en él. Esta gestión permite organizar los datos

de manera adecuada para que puedan enviarse correctamente al nodo LoRaWAN y, posteriormente, a la red IoT.

Además, el ESP32 se caracteriza por su bajo consumo energético, lo que lo convierte en una opción eficiente para sistemas de monitorización que requieren un funcionamiento continuo, y garantiza la estabilidad del prototipo durante su uso.

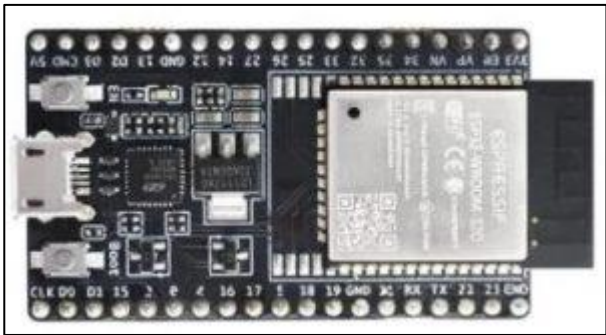


Figura 8. Microcontrolador ESP32

Tabla 9

Características del ESP32

Característica	Descripción
Wi-Fi Protocolos	802.11b/g/n, 2.4 GHz
SRAM	520 KB
ROM	448 KB
Pines GPIO	34
Bluetooth	Bluetooth v4.2 y BLE
Voltaje de operación	3.0 V 3.6 V

Notas. Fuente: Elaboración propia.

3.2.2 Sensor de voltaje AC ZMPT101B

El sensor de voltaje ZMPT101B, mostrado en la figura 9, mide el voltaje alterno del sistema eléctrico mediante un módulo basado en un transformador con salida analógica. Este componente incorpora un amplificador operacional que permite ajustar y estabilizar la señal a través de un potenciómetro [21].

En el prototipo, el sensor adquiere la señal de voltaje de baja tensión y la transmite al microcontrolador ESP32, donde se procesa para obtener lecturas de voltaje en tiempo real. Para garantizar una mayor exactitud en las mediciones, es necesario realizar una calibración con un voltímetro de referencia.

Además, este sensor está diseñado únicamente para medir señales de voltaje en corriente alterna (CA), por lo que es adecuado para aplicaciones de monitorización energética en entornos de baja tensión.

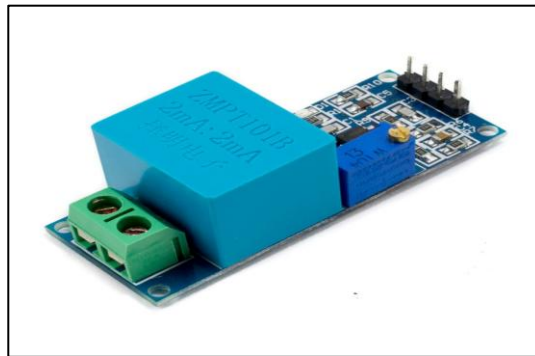


Figura 9. Sensor de voltaje

En la siguiente tabla 10 se muestran las características del sensor de voltaje ZMPT101B con todo detalle.

Tabla 10

Características del sensor de voltaje

Características	Detalles
Voltaje de alimentación	3.3V – 5VDC
Voltaje alterno de entrada	250VAC Max.
Voltaje alterno de salida	Onda senoidal de 5VAC Max
Señal de entrada	Analógica senoidal

Notas. Fuente: Elaboración propia.

3.2.3 Sensor de corriente SCT-013

El SCT-013, que se muestra en la figura 10, es un sensor de corriente no invasivo que opera como un transformador y permite medir la corriente alterna mediante inducción. Este proceso genera una señal proporcional a la corriente que circula por el conductor [22].

Durante el desarrollo del prototipo, este sensor capta la corriente del sistema eléctrico sin necesidad de intervenir físicamente en el circuito. La señal obtenida se envía al ESP32 para su procesamiento y posterior análisis. Implementar este componente es esencial para completar la medición y obtener una representación más completa y precisa de los datos del comportamiento eléctrico.

Su uso es adecuado debido a su seguridad, fácil instalación y plena compatibilidad con sistemas de monitorización energética, lo que lo convierte en una solución eficiente para aplicaciones IoT.



Figura 10. Sensor de corriente

En la siguiente tabla 11 se detallan las características importantes del sensor de corriente SCT-013.

Tabla 11

Características del sensor de corriente

Características	Descripción
Corriente de entrada	0 a 100 A
Largo del cable	1m
Resistencia dieléctrica	6000V AC/1min
Material del núcleo	Ferrita
Temperatura de trabajo	Entre -25 °C a 70 °C
Grado de resistencia	Grado B

Notas. Fuente: Elaboración propia.

3.2.4 Nodo LoRaWAN CubeCell AB02

El nodo CubeCell, presentado en la figura 11, transmite los datos eléctricos hacia la red IoT utilizando tecnología LoRaWAN, lo que permite la comunicación con el gateway a larga distancia.

El módulo HTCC-AB02 utilizado en este proyecto opera en la clase A de LoRaWAN, que se caracteriza por un modelo de comunicación en el que el dispositivo transmite datos y abre ventanas de percepción inmediatamente después de cada transmisión, lo que contribuye a un uso eficiente de la energía. Además, el modelo permite la conexión de distintos tipos de sensores para la recolección de información [23].

Para este proyecto se emplea la banda de frecuencia correspondiente a Norteamérica (Estados Unidos), de alrededor de 915 MHz, adecuada para la configuración interna del sistema.

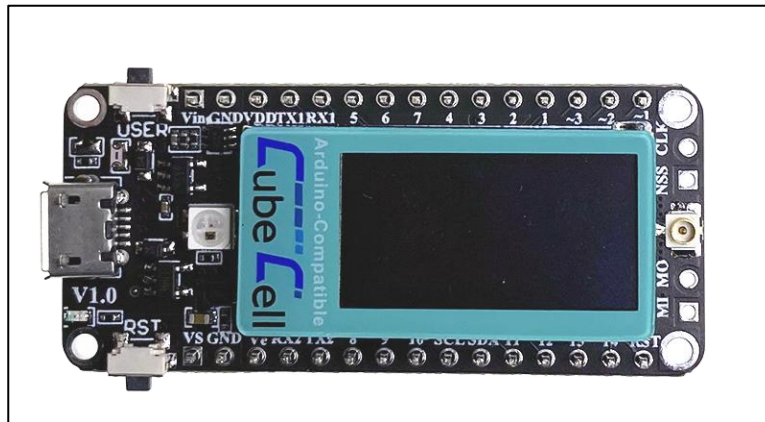


Figura 11. Módulo CubeCell AB02

En la tabla se presentan parámetros con descripciones importantes del módulo CubeCell AB02.

Tabla 12*Parámetros importantes del módulo CubeCell AB02*

Características	Detalles
Chip maestro	ASR6502
Chipset LoRa	SX1262
Alimentación	2.1 – 3.5V
Frecuencia con la que opera	470-510 MHz
	863-923 MHz
Memoria	128 kbits de memoria Flash
	16 kbits de memoria RAM
Trabaja a temperatura	Entre -20 a 70 grados Celsius
Batería	Opera con una batería de Litio de 3.7V
Interfaz LoRa	LoRaWAN 1.0.2
Interfaz de comunicación	UART, SPI, I2C, SWD
Pines	Pines GPIO: 21
	Pines PWM: 4
	Pines ADC: 3 (12bits)

Notas. Fuente: Elaboración propia.

3.2.5 Gateway LoRaWAN SenseCAP M2

El gateway SenseCAP M2, ilustrado en la figura 12, es un gateway LoRaWAN encargado de recibir la información enviada por los nodos y de enviarla posteriormente a una plataforma IoT mediante una conexión a internet. Su funcionamiento se basa en actuar como enlace entre la red LoRaWAN y la red IP, lo que facilita la comunicación con la plataforma The Things Network (TTN).

Este dispositivo ofrece una cobertura de gran alcance y una capacidad de salida de señal fuerte, por lo que es apropiado para el despliegue de redes LoRaWAN en aplicaciones de monitorización. También es adecuado para planes de frecuencia LoRaWAN a nivel global, desde 865 MHz hasta 923 MHz [24].



Figura 12. Gateway SenseCAP M2

La tabla 13 muestra una descripción de las características principales del gateway SenseCAP M2.

Tabla 13*Descripción del gateway SenseCAP M2*

Características	Detalles
Memoria RAM	DDR2 128MB
Memoria Flash	32MB
Procesador	MT7628(MIPS24KEc@580MHz)
Chip integrado del Gateway de largo alcance	SX1302 / SX1303
Frecuencia	IN865/EU868/RU/US915/AU915/KR920/AS923
Wi-Fi	Soporta 2.4 GHz 802.11 b/g/n
LoRaWAN	Soporta Clase A, C
Canales	8
Bluetooth	Módulo Dual, Bluetooth V2.1 + EDR y V5.2
Ethernet	RJ45 (10M / 100M)
Antena	Largo alcance: Antena externa de 3dBi con base. Wi-Fi: antena interna BLE: Antena cerámica
Voltaje de entrada	DC 12V – 2A

Notas. Fuente: Elaboración propia.

3.2.6 Batería de litio de 3,7 V

La batería de litio de 3,7 V y 1100 mAh, representada en la figura 13, proporciona la alimentación necesaria al prototipo. Este tipo de batería destaca por su alta densidad energética y su capacidad de recarga, lo que la convierte en una opción adecuada para dispositivos electrónicos [25].

En este proyecto, la batería suministra alimentación a los componentes principales, como el microcontrolador ESP32 y el módulo CubeCell AB02, y garantiza su funcionamiento continuo. El uso de esta batería es indispensable debido a su reducido tamaño, eficiencia energética y facilidad de integración con aplicaciones en entornos IoT.



Figura 13. Batería de litio

3.3 Herramientas de software para la gestión y visualización de datos

El sistema propuesto integra varias herramientas de software que permiten gestionar, manipular, visualizar y almacenar los datos eléctricos adquiridos. Estas plataformas o aplicaciones facilitan la comunicación entre los dispositivos del sistema y el usuario final en la arquitectura IoT implementada.

3.3.1 Plataforma IoT - The Things Network (TTN)

The Things Network, plataforma IoT mostrada en la figura 14, hace posible la gestión de dispositivos LoRaWAN y, sobre todo, la recepción de la información enviada al gateway. Su funcionamiento consiste en actuar como servidor de red, asumir el procesamiento de los mensajes procedentes de los nodos y ponerlos a disposición para que se integren con otras aplicaciones [26].

En el proyecto, la red LoRaWAN se llevó a cabo utilizando TTN como servidor, cuyo papel es recibir la información del nodo CubeCell AB02 mediante un gateway. De esta forma, se permite su decodificación y se envían los datos a las plataformas de visualización, que en este caso es Node Red.



Figura 14. Plataforma TTN

3.3.2 Node - RED: plataforma de visualización

La herramienta o plataforma Node-RED se proyecta en el sistema propuesto como el entorno encargado de la integración, el procesamiento y la visualización de los datos eléctricos obtenidos. En este proyecto, Node-RED participa como un servidor que recibe los datos obtenidos de la plataforma TTN mediante el protocolo MQTT, lo que permite administrar los flujos de datos generados por el sistema.

Además, mediante su entorno basado en nodos, se establecen y configuran flujos que posibilitan el procesamiento de los datos de voltaje y corriente, convirtiéndolos en información comprensible que se representa mediante una visualización o interfaz gráfica en tiempo real.

De esta manera, la herramienta Node-RED destaca por su funcionalidad en el sistema, ya que permite el monitoreo o análisis continuo de las variables eléctricas y mejora la interacción del usuario con los datos obtenidos por el prototipo.

3.3.3 Arduino IDE

En el sistema sugerido, se emplea el entorno de desarrollo Arduino IDE para programar y configurar tanto el microcontrolador ESP32 como el nodo CubeCell AB02. En el contexto del proyecto, este escenario facilita la puesta en marcha de los algoritmos necesarios para adquirir datos de los sensores, procesarlos y preparar la información para enviarla a través de la tecnología LoRaWAN [27].

Se implementan y cargan las rutinas de lectura de los sensores de voltaje y de corriente, así como la lógica para el intercambio de información entre los dispositivos mediante el software Arduino IDE. Esto garantiza el correcto funcionamiento del sistema desde la fase de medición hasta el envío de datos. También facilita la carga y la depuración del código, lo que permite comprobar el funcionamiento del prototipo durante su desarrollo.

Arduino IDE influye directamente en la implementación del sistema al ofrecer herramientas precisas para programar y controlar los dispositivos que conforman la arquitectura IoT propuesta.

Tabla 14*Tabla de librerías para Arduino*

Librerías/Biblioteca	Descripción	Edición
	Habilita la vinculación del microcontrolador	
Wifi.h	ESP32 con redes Wi-Fi, necesariamente indispensable para que exista una comunicación con plataformas IoT.	Incluida en ESP32 Core
Arduino JSON	Ayuda a codificar y decodificar los datos en formato JSON, utilizado para estructurar la información transmitida.	6.x
PubSubClient	Implementa el protocolo MQTT, lo que posibilita la comunicación entre el sistema y las plataformas como TTN y Node-RED.	2.8
LoRawan_APP	Librería empleada para la configuración y gestión de la comunicación LoRaWAN en el módulo CubeCell AB02.	Incluida en (SDK CubeCell)
Wire.h	Posibilita la comunicación mediante protocolos I2C mediante el microcontrolador y dispositivos externos.	Estándar
Emonlib	Permite calcular los valores de corriente RMS a partir de señales analógicas del sensor SCT-013.	1.1.0

Notas. Fuente: Elaboración propia.

3.3.4 Bróker Mosquito (Eclipse Mosquito)

Se añade el bróker Eclipse Mosquito a la arquitectura del sistema propuesto como componente que se encarga de gestionar la comunicación basada en el protocolo MQTT. Su función principal es actuar como intermediario en la transmisión de mensajes entre los diferentes elementos del sistema, lo que facilita que los datos eléctricos adquiridos se transmitan de manera eficaz.

Durante la operación del sistema, los datos creados y administrados en la plataforma TTN se publican en temas MQTT. Estos también son suscritos por Node-RED mediante el bróker Mosquito. De este modo, cada mensaje transmitido se recibe en tiempo real, lo que facilita su procesamiento, visualización y almacenamiento dentro del sistema.

Mosquito, al estar dentro de la arquitectura, permite que los componentes se desacoplen, ya que los dispositivos no necesitan comunicarse directamente entre ellos, sino que interactúan a través de protocolos establecidos. Esto mejora la estabilidad del sistema y asegura una comunicación liviana, eficaz y apropiada para las condiciones de IoT.

3.4 Elaboración y diseño de la propuesta del sistema

El diseño plantea un sistema de supervisión de variables eléctricas que emplea plataformas IoT y tecnología LoRaWAN para enviar y visualizar datos a distancia. El sistema está compuesto por módulos autónomos, lo que simplifica su implementación y mantenimiento. Cada uno de los módulos cumple una función específica: captura de señales, procesamiento de datos, transmisión inalámbrica, visualización y almacenamiento.

Los sensores de voltaje y de corriente inician el flujo de operación y se encargan de enviar datos al microcontrolador. Los datos se transmiten a través del nodo LoRaWAN hacia el gateway

y, posteriormente, se gestionan en la plataforma The Things Network (TTN). En resumen, una herramienta lógica como Node-RED procesa la información y permite la visualización en tiempo real, así como el registro para el análisis del sistema eléctrico.

3.4.1 Diseño electrónico de conexiones

El diseño electrónico del prototipo se desarrolló con el software Fritzing (como se observa en la figura 15), conocido por su capacidad y fiabilidad a la hora de elaborar circuitos y conexiones eléctricas. Esta aplicación facilita la planificación y el montaje eficiente del prototipo, lo que reduce la posibilidad de cometer errores durante el proceso de elaboración o fabricación.

A continuación, se muestra un procedimiento paso a paso para la creación del prototipo.

La figura 15 muestra el diagrama esquemático de las conexiones electrónicas del prototipo de medición. La arquitectura de hardware se clasifica en tres subsistemas principales, de la siguiente manera:

- **Sistema de alimentación y gestión energética:** formado por una pila de litio de 3,7 V acoplada a un módulo cargador TP4056 y un elevador de voltaje regulado a 5 V para garantizar el suministro constante al microprocesador y los sensores.
- **Bloque de procesamiento y comunicación:** El núcleo del sistema es el microcontrolador ESP32, que está conectado mediante un bus I2C (pines d21 y d22) al módulo Heltec AB02 para la transmisión inalámbrica de datos.
- **Bloque de instrumentación y acondicionamiento:** integra el sensor de voltaje ZMPT101B en el pin digital d34 y el sensor de corriente SCT-013. Este último sensor tiene conectado al pin D34 un circuito de acondicionamiento analógico pasivo (divisor de voltaje con resistencias de 10 k Ω y un condensador de filtrado

de 10 μ F) para adaptar las señales antes de su lectura.

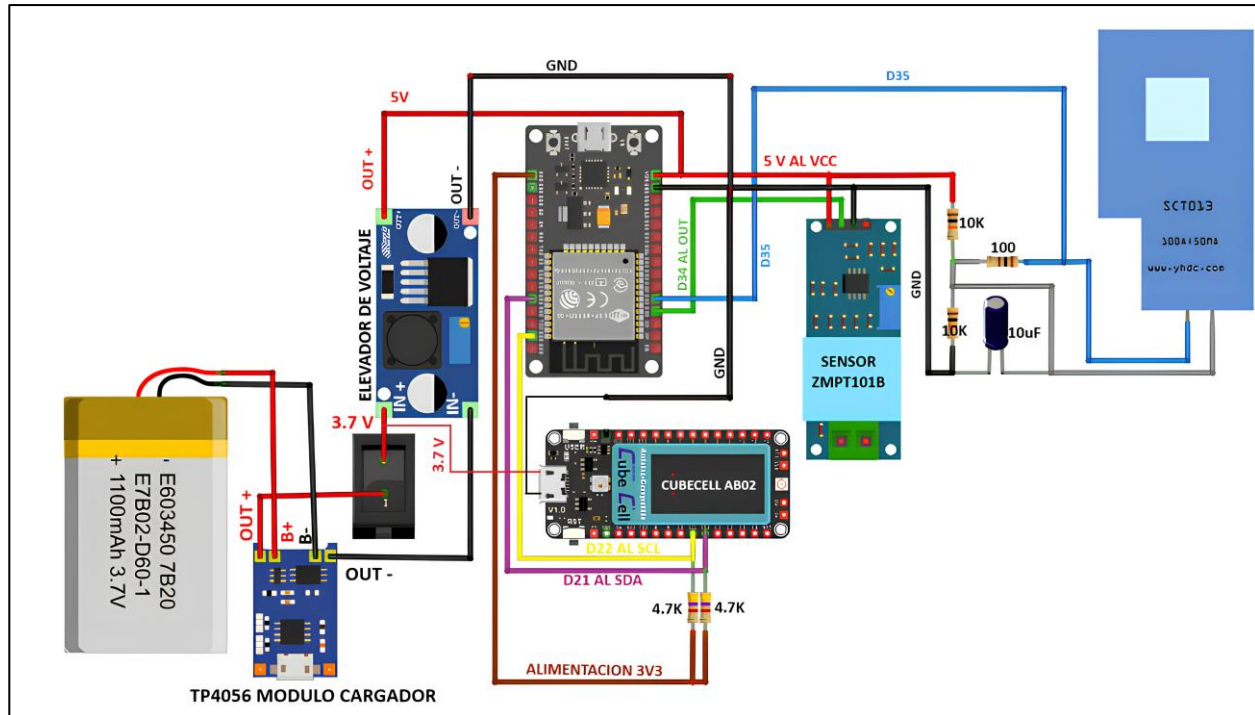


Figura 15. Diseño de conexiones del sistema

3.4.1.1 Dirección de librerías usadas para diseño en Fritzing

Librería para el componente sensor de corriente SCT-013:

<https://forum.fritzing.org/t/obter-os-componentes/3916/10>

Librería para el componente sensor de voltaje ZMPT101B:

<https://forum.fritzing.org/t/zmpt101b-voltage-sensor/8235>

Librería para el componente módulo Heltec CubeCell AB02:

<https://forum.fritzing.org/t/heltec-dev-boards/8112/36>

Librería para el componente ESP32:

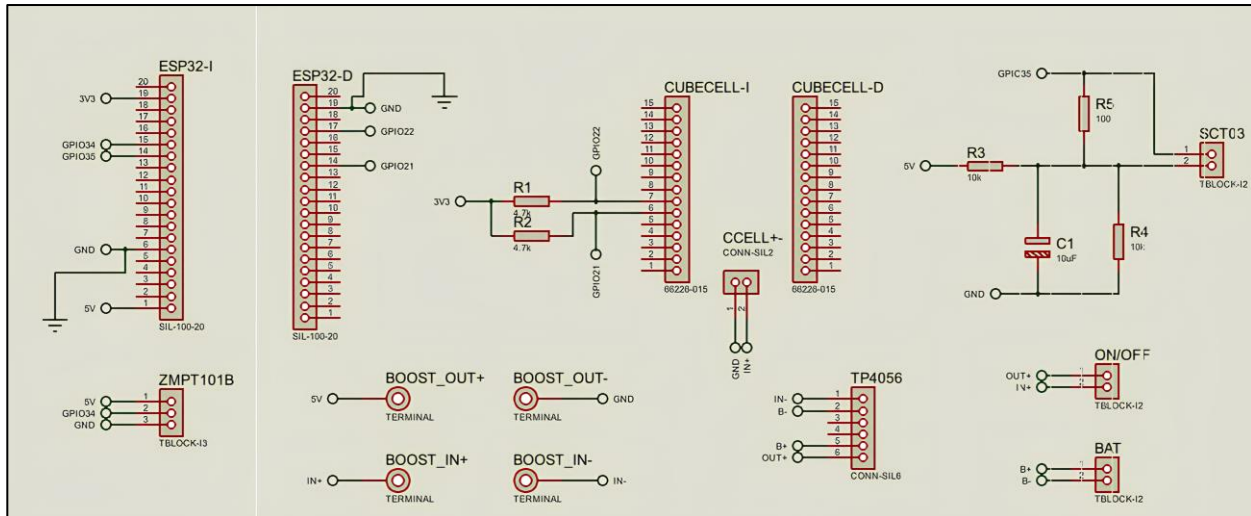
<https://forum.fritzing.org/t/esp-32-de-30-pines/22637>

3.4.2 Diseño esquemático del circuito en Proteus

Una vez establecidas las conexiones generales entre los componentes del sistema, se elaboró el diseño esquemático del circuito mediante el software Proteus, especializado en el diseño y desarrollo de circuitos electrónicos. En esta etapa se logró representar de manera detallada la interconexión eléctrica entre los diferentes componentes del prototipo, lo que facilitó la verificación de las conexiones antes de su implementación física.

El diseño esquemático incluye el microcontrolador ESP32 como núcleo principal de adquisición y procesamiento de datos, el módulo CubeCell AB02, encargado de la transmisión inalámbrica mediante tecnología LoRaWAN, y los sensores de voltaje y corriente, junto con el resto de componentes necesarios para el funcionamiento del sistema. La gráfica del circuito permitió verificar la compatibilidad entre los dispositivos y garantizar una adecuada distribución de conexiones, lo que disminuyó la probabilidad de errores durante la etapa de implementación.

La figura 16 muestra el diseño esquemático completo del sistema desarrollado en Proteus, que sirve de base para el diseño de la tarjeta de circuito impreso (PCB) y garantiza una integración adecuada de los componentes electrónicos.



3.4.3 Diseño del circuito impreso (PCB layout)

Se diseñó la PCB teniendo en cuenta criterios de organización, accesibilidad y facilidad de ensamblaje, logrando integrar los principales módulos electrónicos en una sola tarjeta de baquelita. Esta disposición ayuda a reducir el uso de conexiones externas por cables, a mejorar la estabilidad mecánica del circuito y a minimizar las posibles interferencias eléctricas en el funcionamiento del prototipo.

La figura 17 muestra el diseño final del circuito impreso desarrollado para el prototipo de monitorización de variables eléctricas.

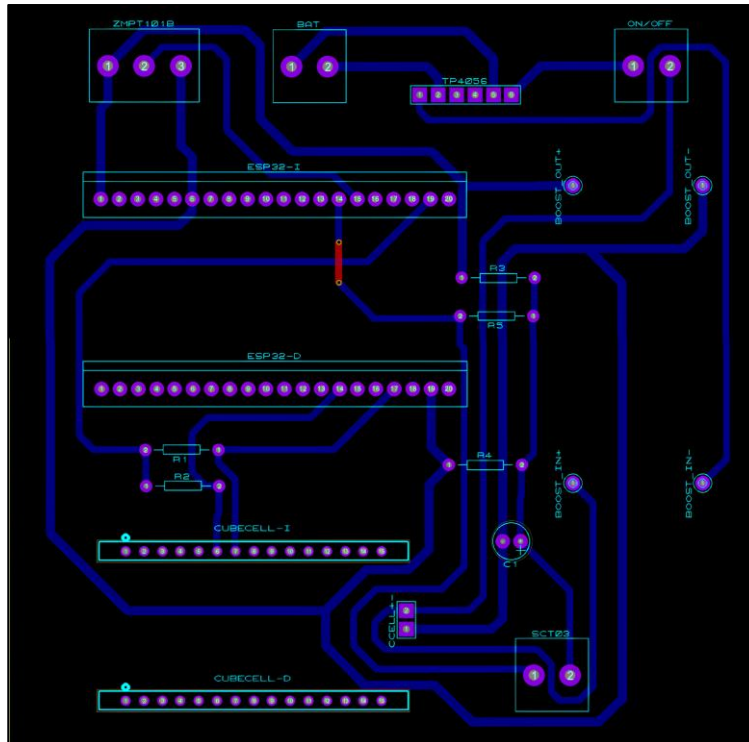


Figura 17. Diseño de PCB para el prototipo

3.4.4 Diseño tridimensional del prototipo

El modelo establece una base sobre la que se instalan los principales componentes electrónicos del sistema. En el centro se encuentra la placa de desarrollo ESP32, que procesa los datos de los sensores de corriente y voltaje. También se tiene en cuenta el espacio que ocuparán el módulo de alimentación, el sistema de comunicación y los elementos auxiliares necesarios para el correcto funcionamiento del prototipo.

El diseño tridimensional no solo mejoró la presentación del proyecto, sino que también ayudó a identificar posibles problemas de espacio, lo que permitió realizar ajustes antes de fabricar el prototipo físico. Esta fase optimizó la disposición interna de los componentes, minimizó los errores en el ensamblaje y logró una estructura ordenada que facilita la disipación de calor y el

acceso a los dispositivos para futuras revisiones o modificaciones. En la figura 18 se observa el diseño.

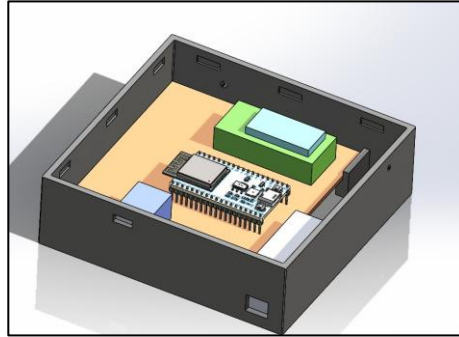


Figura 18. Diseño 3D (caja)

3.5 Configuración e integración del sistema IoT

3.5.1 Configuración del gateway SenseCAP M2

Para acceder a la configuración del gateway SenseCAP M2, es necesario utilizar las credenciales de autenticación proporcionadas por el fabricante. Estas credenciales permiten acceder a la interfaz de administración del aparato a través de un navegador web, como se muestra en la figura 19. Una vez establecida la conexión a la red local, se accede mediante la dirección IP asignada al gateway (192.168.168.1). A continuación, se solicita un usuario y una contraseña para garantizar el acceso seguro. Este procedimiento de autenticación es esencial para habilitar la configuración de los parámetros de la red y la vinculación con la plataforma TTN, lo que permite supervisar el estado del dispositivo dentro de la arquitectura del sistema.



Figura 19. Información y credenciales del gateway

En la interfaz del gateway, hay que acceder a la sección correspondiente a LoRa y, posteriormente, ubicar la opción llamada Channel Plan, como se muestra en la figura 20.

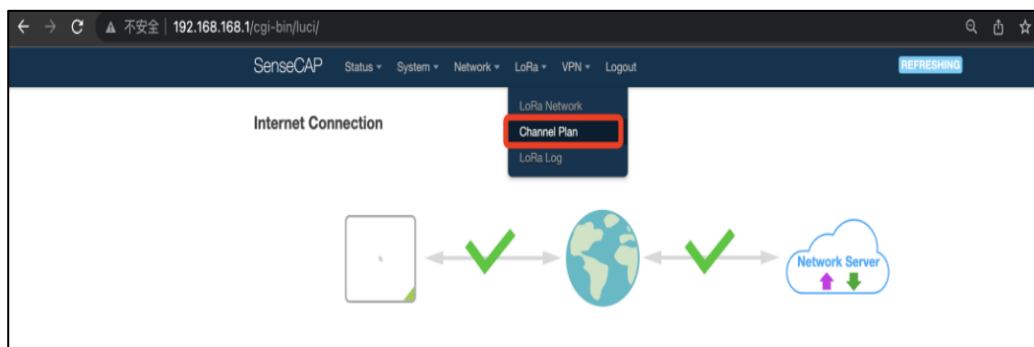


Figura 20. Interfaz del gateway SenseCAP M2

Como se muestra en la figura 21. A continuación, se marca la región US902-928 en la casilla correspondiente. Después, en el canal 0, se configura la frecuencia del plan en FSB1. Esta configuración permite ajustar una sincronización exacta con el servidor indicado para visualizar el flujo de datos.



Figura 21. Elección del plan de canal

Como se muestra en la figura 22. Para configurar el dispositivo en una red inalámbrica, primero hay que acceder a la opción Network y, a continuación, seleccionar Wireless. A continuación, hay que conectar el dispositivo a una red Wi-Fi disponible.

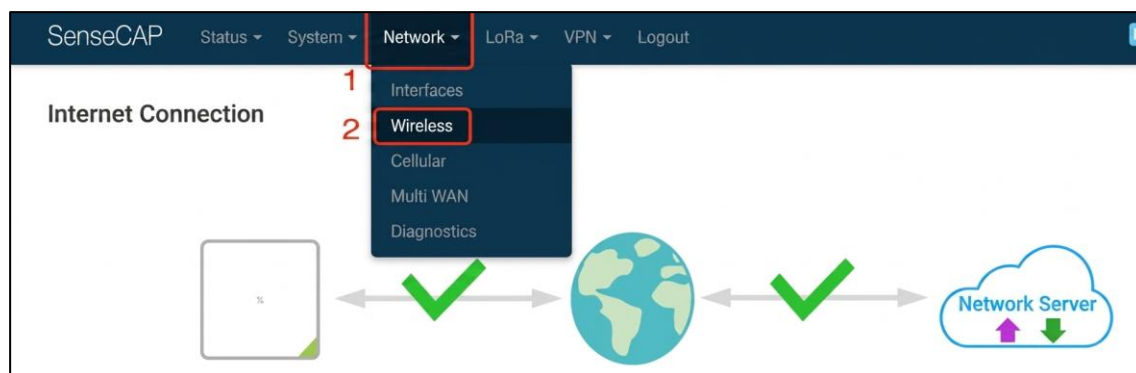


Figura 22. Configuración de red en el gateway

Al hacer clic en la pestaña Scan, se escanean las redes disponibles. A continuación, se selecciona la red Wi-Fi a la que se desea conectar el gateway, tal y como se observa en la figura 23.

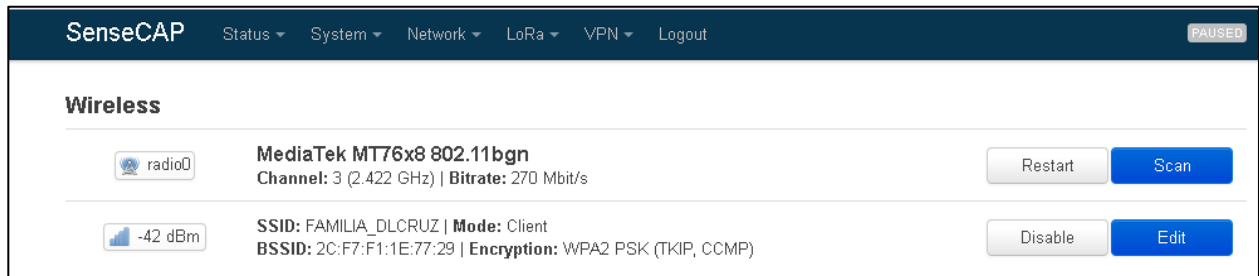


Figura 23. Interfaz para escanear redes inalámbricas

Al terminar el escaneo, aparece una lista con las redes Wi-Fi disponibles. De este listado, hay que elegir la red que se quiere y pulsar en «Join network» para iniciar la conexión (véase la figura 24).

Join Network: Wireless Scan						
Signal	SSID	Channel	Mode	BSSID	Encryption	
-30 dBm	FAMILIA_DLCRUZ	3	Master	10:62:EB:87:88:D1	mixed WPA/WPA2 PSK (CCMP)	Join Network
-55 dBm	TP-Link_59C0	3	Master	40:3F:8C:8B:59:C0	WPA2 PSK (CCMP)	Join Network
-70 dBm	XTRIM_PATUCHIN	6	Master	60:A4:B7:7D:5E:5A	WPA2 PSK (CCMP)	Join Network
-87 dBm	Maestro_Piloso_Antonio	10	Master	28:87:BA:06:8D:A2	WPA2 PSK (CCMP)	Join Network
-88 dBm	ISABEL_26	5	Master	F0:09:0D:39:48:E0	WPA2 PSK (CCMP)	Join Network
-90 dBm	hidden	8	Master	7A:EB:46:83:C2:DB	mixed WPA/WPA2 PSK (TKIP, CCMP)	Join Network
-93 dBm	XTRIM_FAMILIA_FIGUEROA	10	Master	64:6E:60:09:89:06	WPA2 PSK (CCMP)	Join Network
						Stop refresh Dismiss

Figura 24. Establecer una red inalámbrica

Para finalizar la configuración, como primer paso (1), es indispensable introducir las credenciales que coinciden en la parte indicada y, posteriormente, para completar el proceso de conexión a la red Wi-Fi se hace clic en Submit como segundo paso (2). Como se observa en la figura 25.

Joining Network: "FAMILIA_DLCRUZ"

Replace wireless configuration ☒

Name of the new network

WPA passphrase 1

Specify the secret encryption key here.

Lock to BSSID ☐

Instead of joining any network with a matching SSID, only connect to the BSSID **10:62:EB:87:88:D1**.

Cancel Submit 2

Figura 25. Ingresar datos para nueva red inalámbrica

El gateway permite configurar diferentes tipos de funcionamiento para establecer la comunicación mediante LoRa, como base, reenvío de paquetes o servidor de la red local. En este caso, se utiliza el modo de reenvío de paquetes, para el cual es necesario configurar el servicio en la dirección “nam1.cloud.thethings.network” para asegurar la comunicación con TTN (véase la figura 26).

SenseCAP Status System Network LoRa VPN Logout

LoRaWAN Network Settings

Gateway EUI: 2CF7F1162600012

Mode: Packet Forwarder

Packet Forwarder Settings

General Settings Intervals Settings Beacon Settings GPS Information Forward Rules Packet Filter

Gateway EUI: 2CF7F1162600012

Server Address: nam1.cloud.thethings.network

Server Port (Up): 1700

Server Port (Down): 1700

Save & Apply

Powered by LuCI / OpenWrt 21.02.0 r1-20220615 r16279-5cc0535800

Figura 26. Configuración para asignar LoRaWAN

3.5.2 Configuración de la plataforma The Things Network

Para poder enviar datos, es necesario crear una cuenta en la plataforma TTN. Una vez realizado el registro, hay que iniciar sesión e ir a la pestaña “Consola” para empezar con la configuración correspondiente. Como se observa en la figura 27.

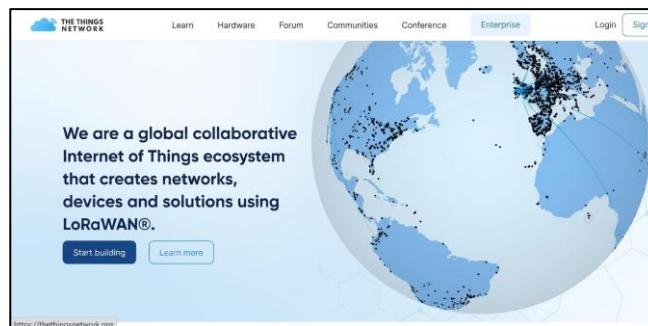


Figura 27. Crear una cuenta en la plataforma TTN

Después de iniciar sesión en la aplicación TTN, se selecciona la región (Estados Unidos), tal y como se muestra en la figura 28. En este caso, se selecciona la opción North America 1.

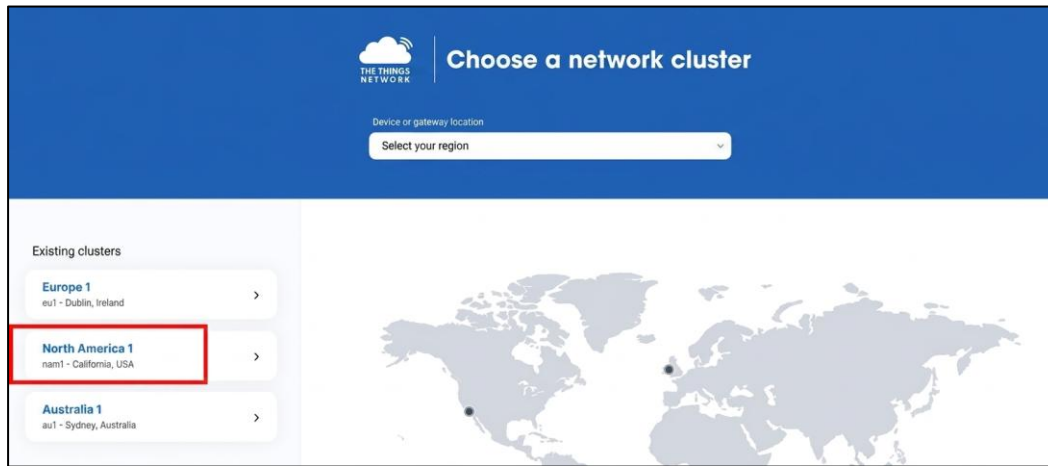


Figura 28. Selección de región en la plataforma TTN

3.5.3 Configuración del gateway SenseCAP M2 en la plataforma TTN

Se muestra la interfaz de la plataforma TTN, donde se accede a la pestaña de gateway. Esta sección sirve para registrar el dispositivo gateway SenseCAP M2 en TTN. Tal y como se muestra en la figura 29.

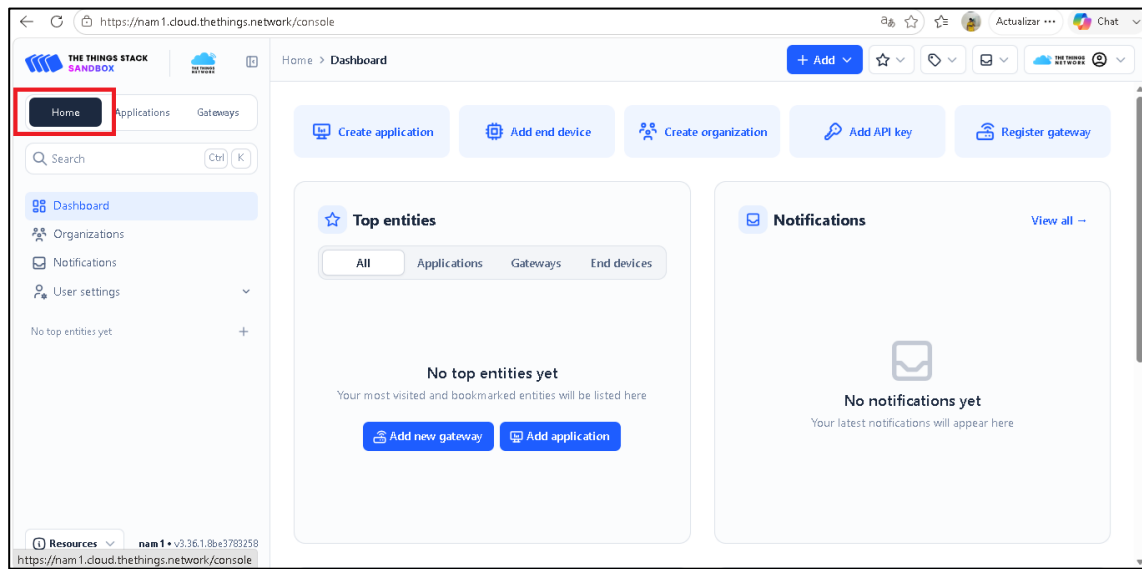


Figura 29. Interfaz inicial de TTN

Al hacer clic en la opción Gateway, se accede a la sección Register Gateway, como se muestra en la figura 30.

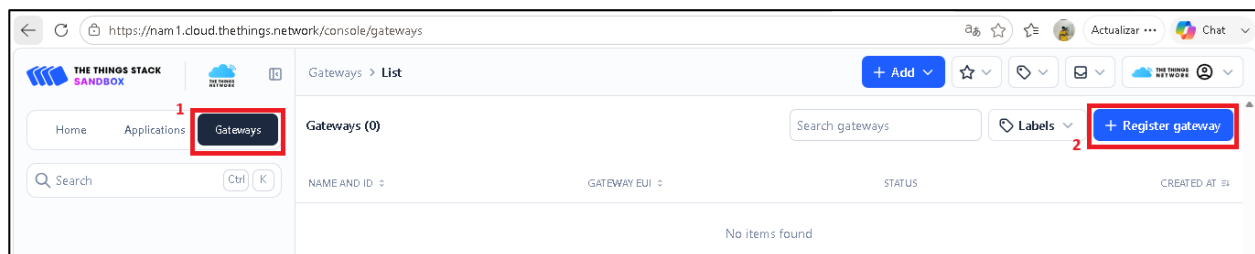


Figura 30. Selección del registro del gateway

La plataforma TTN ofrece dos métodos para registrar el dispositivo gateway SenseCAP M2: escaneando el código QR del dispositivo o de forma manual. Como se muestra en la figura 31.

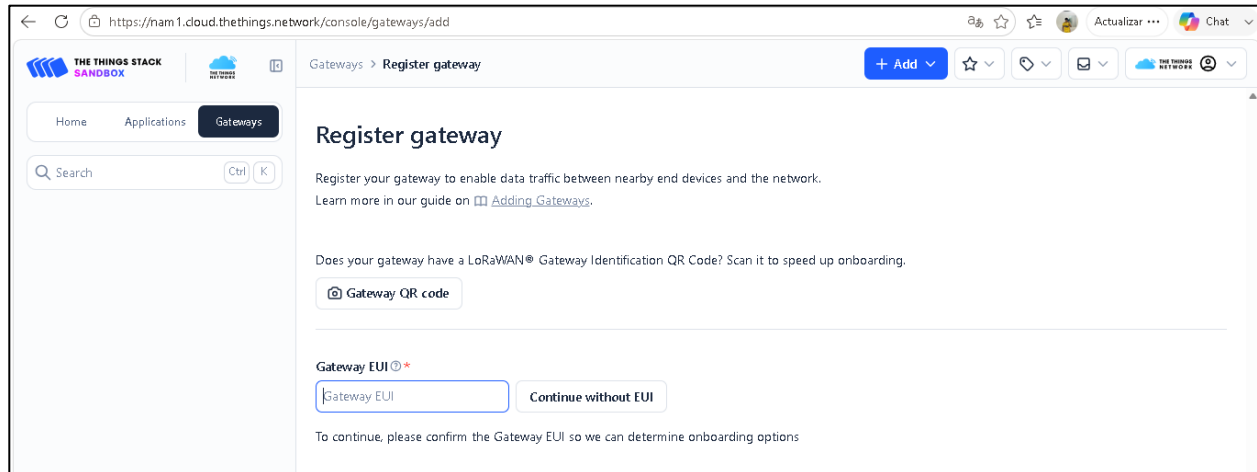


Figura 31. Métodos de registro

Todos los apartados que aparecen en la interfaz de la figura 32 deben rellenarse con la información correspondiente a la puerta de enlace. De esta manera, se puede finalizar dándole clic en “Registrar gateway”.

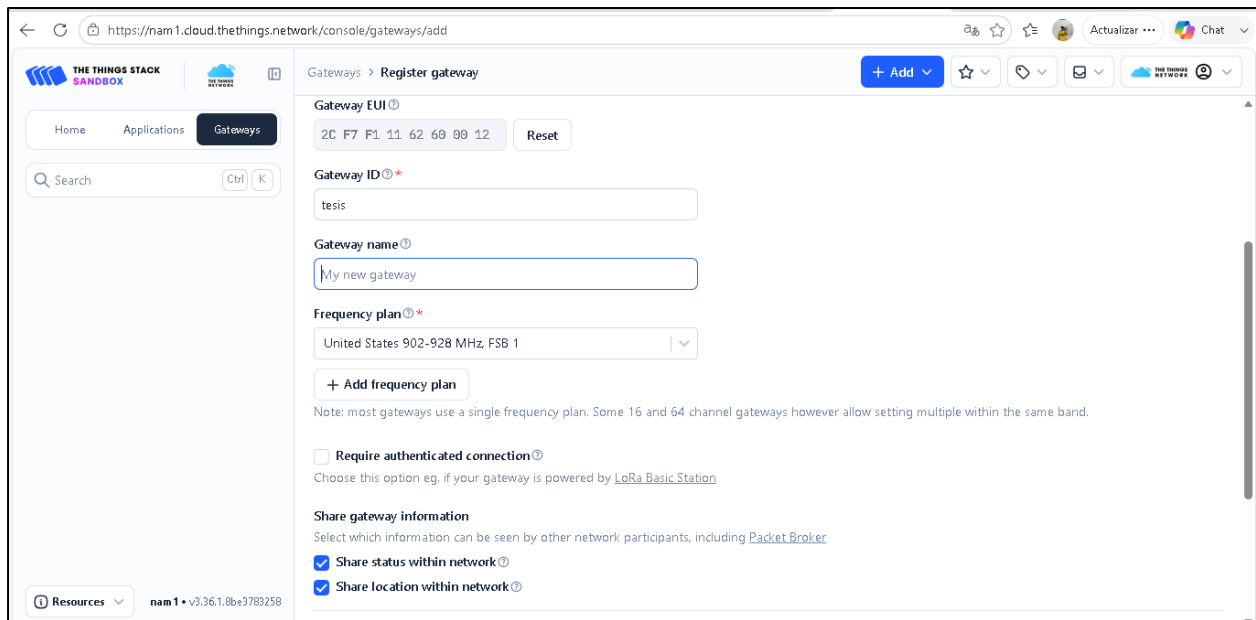


Figura 32. Interfaz de registro del gateway

En la plataforma, es necesario que el dispositivo se muestre con el ID correspondiente y que la conexión creada se visualice como se muestra en la figura 33.

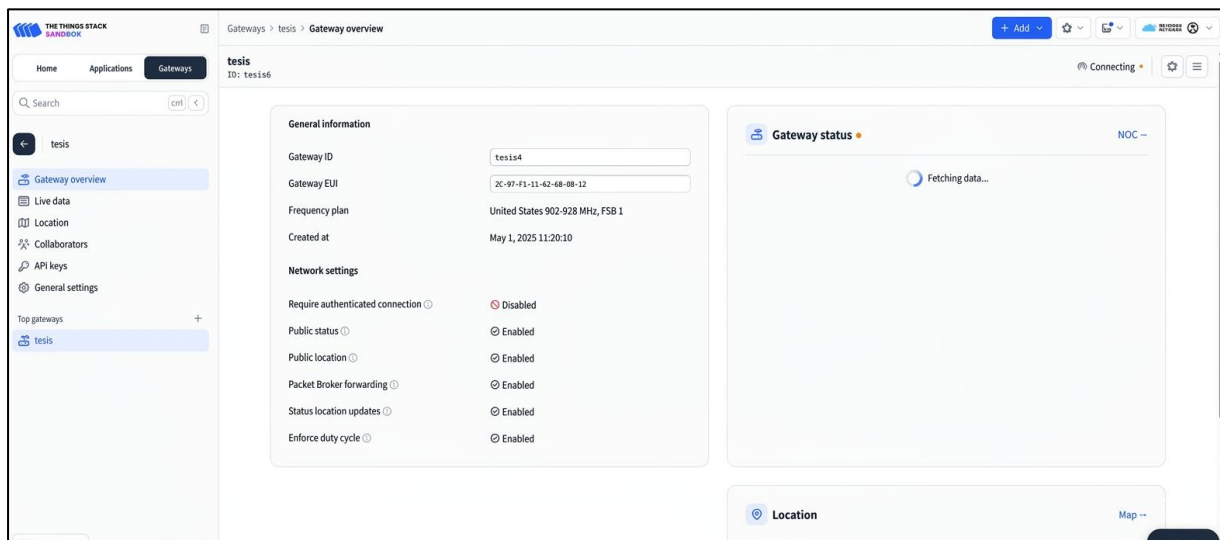


Figura 33. Verificación del estado del gateway

3.5.4 Configuración de nodos finales en la aplicación TTN

Después de registrar el gateway en la aplicación de TTN, también hay que registrar el nodo o incorporarlo. El primer paso es hacer clic en la sección Applications. El segundo paso es hacer clic en la sección Create application, que se encuentra en la parte derecha de la figura 34.

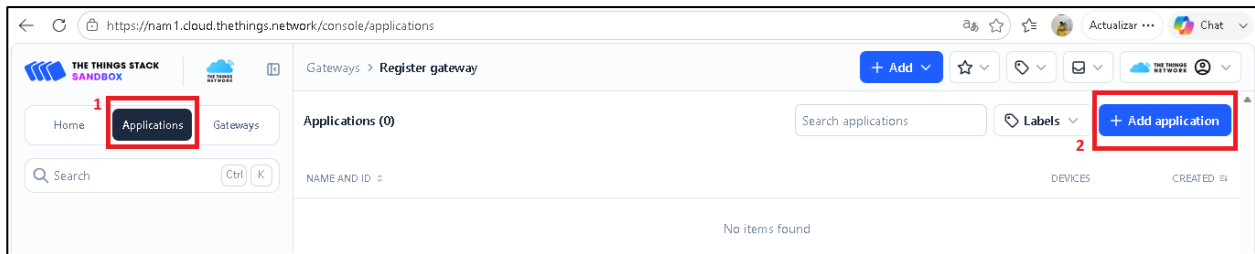


Figura 34. Crear aplicación para nodos

Se completan los campos obligatorios que se requieren para crear la aplicación. Una vez rellenados los campos, se hace clic en “Crear aplicación”, como se observa en la figura 35.

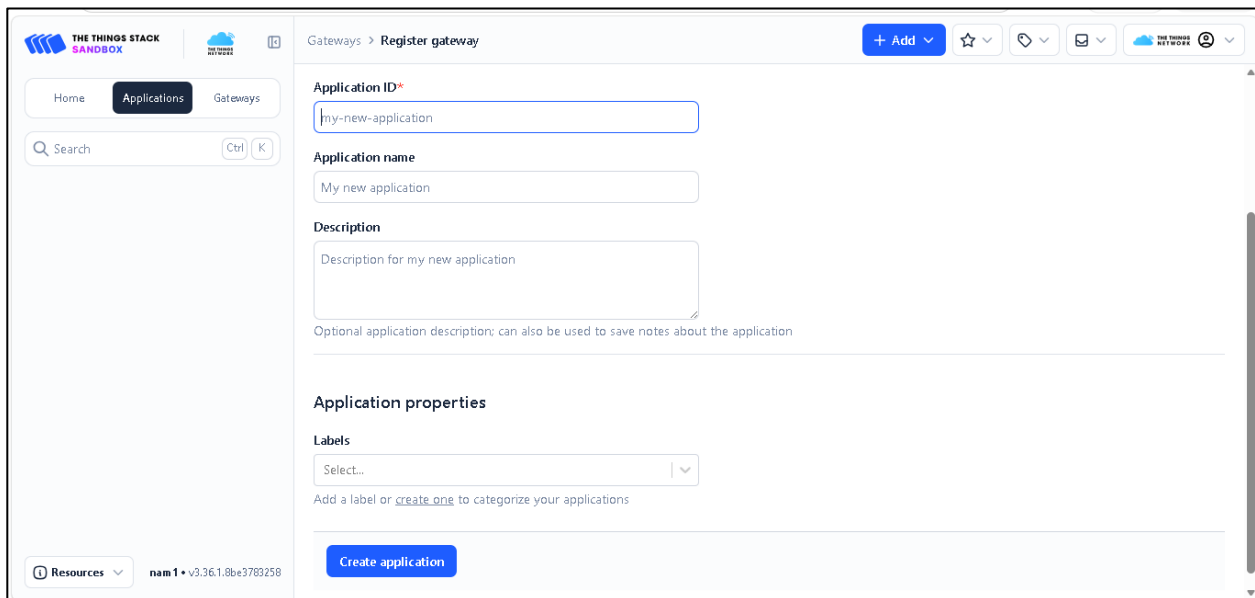


Figura 35. Campos para aplicación en TTN

Una vez creada la aplicación, haga clic en End device en la parte izquierda de la interfaz para agregar el módulo CubeCell AB02 a la aplicación TTN. A continuación, haga clic en Register end device, en la parte derecha, como se muestra en la figura 36.

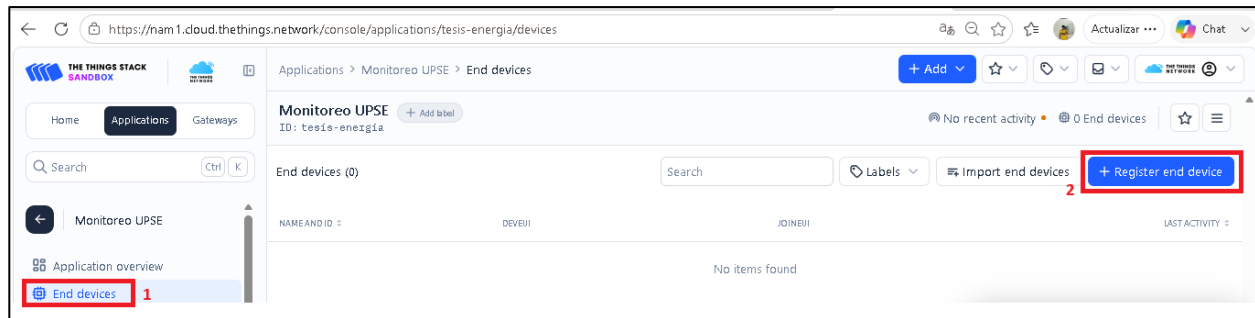


Figura 36. Crear aplicación

En la nueva sección que aparecerá, se rellenan los campos correspondientes al módulo que se va a registrar, se selecciona el modelo HTCC-AB02 y también la clase que usa el módulo (clase A ABP), como se puede ver en la figura 37.

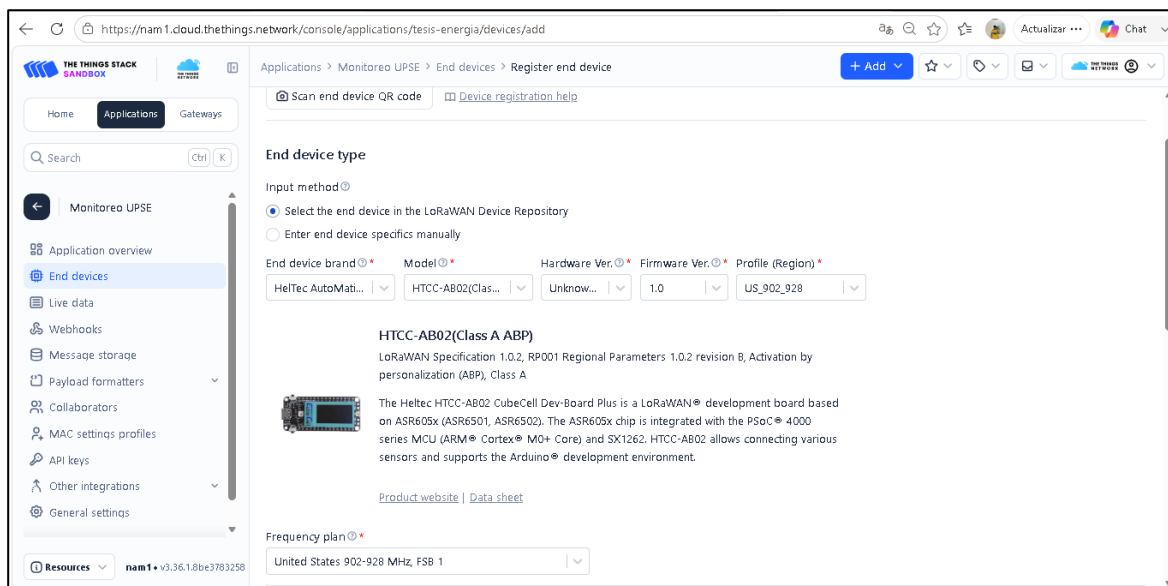


Figura 37. Registro del CubeCell en TTN

The screenshot shows the 'Add end device' page in the Tethys Things Stack console. The left sidebar contains navigation links for 'Home', 'Applications', 'Gateways', 'End devices', 'Live data', 'Webhooks', 'Message storage', 'Payload formatters', 'Collaborators', 'MAC settings profiles', 'API keys', 'Other integrations', and 'General settings'. The main content area is titled 'Applications > Monitoreo UPSE > End devices > Register end device'. It includes a 'JoinEUI' field with a 'Reset' button, a 'DevEUI' field with a 'Generate' button (highlighted with a red box and a red '1'), an 'AppKey' field with a 'Generate' button, and an 'End device ID' field. Below these is the 'Device properties' section with a 'Labels' dropdown and a text input. The 'After registration' section has two radio buttons: 'View registered end device' (selected) and 'Register another end device of this type'. At the bottom, a 'Resources' section shows 'nam1' with a 'Register end device' button (highlighted with a red box and a red '2').

Para verificar que el dispositivo está sincronizado correctamente con la aplicación TTN, debe aparecer “connected”. Tal y como se muestra en la figura 39.



3.5.5 El decodificador de los mensajes en TTN

Para optimizar el uso del espectro electromagnético y cumplir con las restricciones de tiempo de transmisión (duty cycle) que generalmente impone el estándar LoRaWAN, los datos analógicos correspondientes al voltaje, la corriente, la potencia y la energía se empaquetan en el nodo CubeCell AB02 en formato binario (payload de 10 bytes).

Por ello, se implementó un decodificador de carga útil (payload formatter) en la aplicación de The Things Network mediante un algoritmo programado en JavaScript. Este módulo de software en la nube intercepta la trama binaria en formato hexadecimal que llega desde el canal de radiofrecuencia, la decodifica y la reestructura en un formato legible tipo JSON. Este proceso garantiza la integridad y el adecuado procesamiento de los datos energéticos antes de enviarlos a la plataforma de almacenamiento y visualización superior.

Como se observa en la figura 40, el primer paso es seleccionar la opción que se usará en este caso, el payload formatter, y después el formato (JavaScript).

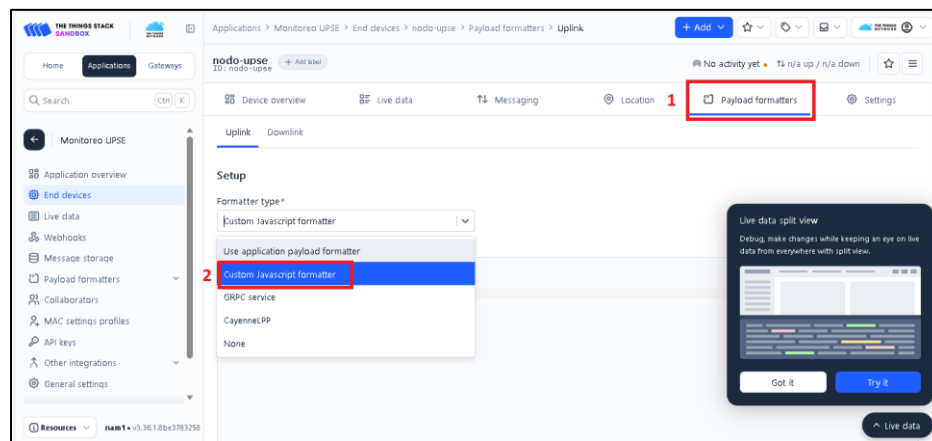


Figura 40. Configuración uplink en TTN

Después de completar la configuración para determinar el formato que se usará en la sección Payload formatters, se debe hacer clic en Save changes o Guardar cambios, tal y como se aprecia en la figura 41.

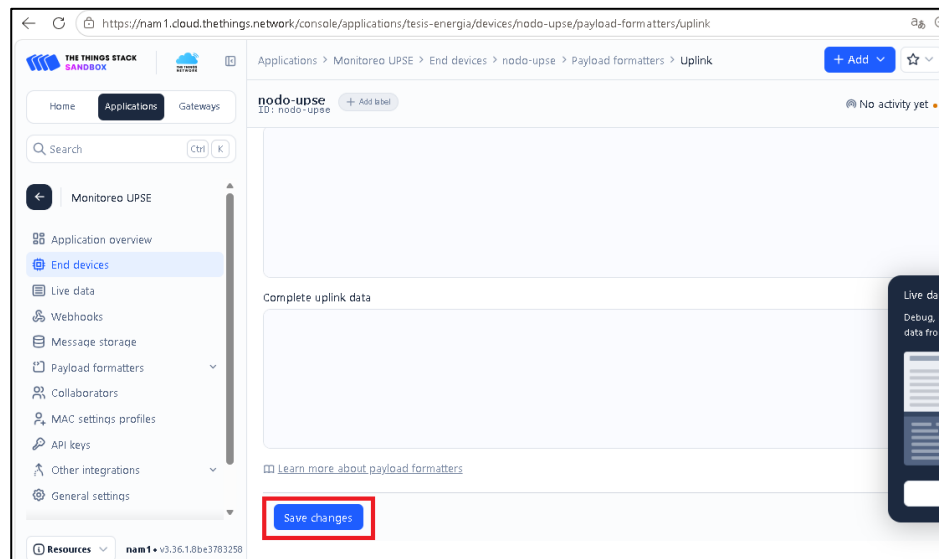
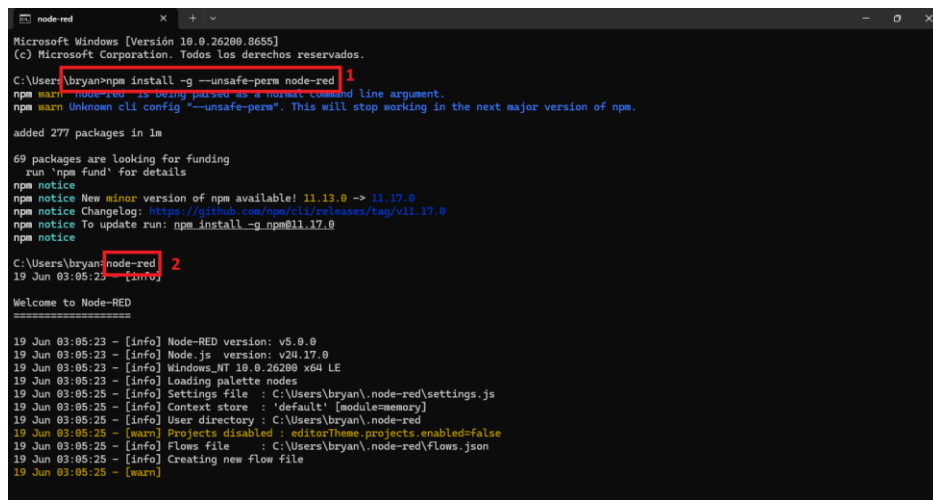


Figura 41. Configuración de los formatters de carga útil en TTN

3.5.6 Configuración del servidor Node-RED

Para instalar el servidor Node-RED, primero hay que descargar Node.js. A continuación, cuando se muestre la interfaz de cmd, se introduce el siguiente comando: `npm install -g --unsafe-perm node-red` (1), que sirve para instalar Node-RED en el sistema. Una vez culminada esta parte, se introduce el siguiente comando: “`node-red`” (2) para iniciar o ejecutar. Como se muestra en la figura 42.



```
Microsoft Windows [Versión 10.0.26200.8655]
(c) Microsoft Corporation. Todos los derechos reservados.

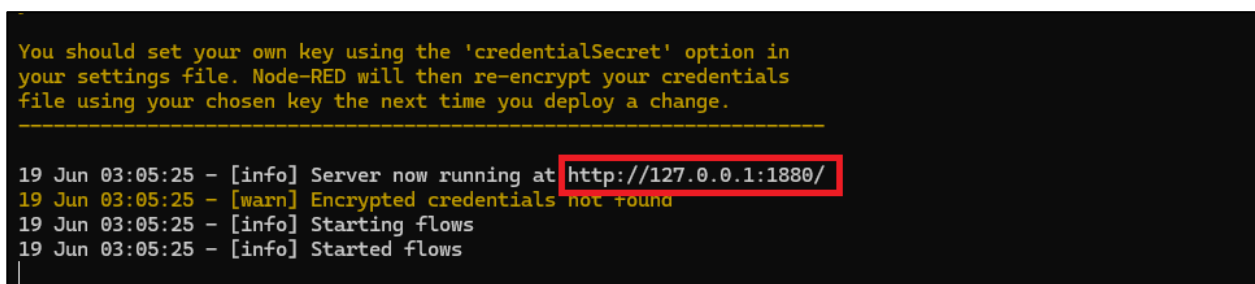
C:\Users\bryan>npm install -g --unsafe-perm node-red 1
npm warn node-red is being pursued as a normal command line argument.
npm warn Unknown cli config "--unsafe-perm". This will stop working in the next major version of npm.

added 277 packages in 1m
69 packages are looking for funding
  run 'npm fund' for details
npm notice New minor version of npm available! 11.13.0 -> 11.17.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.17.0
npm notice To update run: npm install -g npm@11.17.0
npm notice

C:\Users\bryan>node-red 2
19 Jun 03:05:23 - [info]
Welcome to Node-RED
=====
19 Jun 03:05:23 - [info] Node-RED version: v5.9.0
19 Jun 03:05:23 - [info] Node.js version: v24.17.0
19 Jun 03:05:23 - [info] Windows_NT 10.0.26200 x64 LE
19 Jun 03:05:23 - [info] Loading palette nodes
19 Jun 03:05:25 - [info] Settings file   : C:\Users\bryan\.node-red\settings.js
19 Jun 03:05:25 - [info] Context store  : 'default' (module=memory)
19 Jun 03:05:25 - [info] User directory : C:\Users\bryan\.node-red
19 Jun 03:05:25 - [warn] Projects disabled : editorTheme.projects.enabled=false
19 Jun 03:05:25 - [info] Flows file    : C:\Users\bryan\.node-red\flows.json
19 Jun 03:05:25 - [info] Creating new flow file
19 Jun 03:05:25 - [warn]
```

Figura 42. Acceso al servidor Node-RED

Para acceder a la interfaz, se utiliza una dirección IP proporcionada por Node-RED, como se muestra en la figura 43.



```
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
19 Jun 03:05:25 - [info] Server now running at http://127.0.0.1:1880/
19 Jun 03:05:25 - [warn] Encrypted credentials not found
19 Jun 03:05:25 - [info] Starting flows
19 Jun 03:05:25 - [info] Started flows
```

Figura 43. Dirección IP para ejecutar Node-RED

Una vez realizados estos pasos, se puede visualizar su interfaz inicial mediante la dirección IP que asigna Node-RED. Como se muestra en la figura 44.

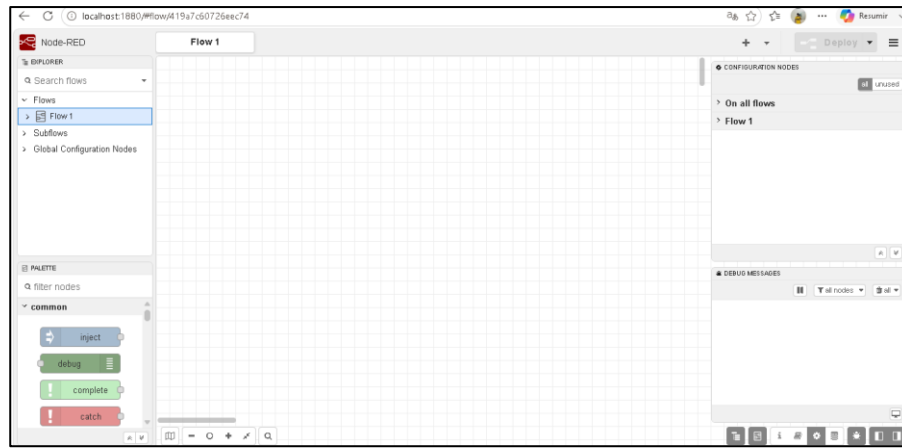


Figura 44. Interfaz del servidor Node-RED

Dentro del servidor Node-RED, el siguiente paso es descargar las librerías necesarias. Para ello, hay que hacer clic en el icono de las tres barritas, que en este caso está ubicado en la parte superior derecha de la interfaz y que corresponde al menú de opciones (1), como se muestra en la figura 45. A continuación, se accede haciendo clic en Manage palette (2).

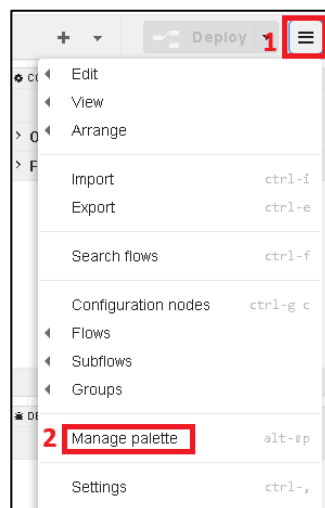


Figura 45. Sección de administración de paletas

Una vez dentro del apartado que brinda Manage Palette, se hace clic en Install para poder instalar las librerías necesarias que se buscarán en el apartado de Search. Como se muestra en la figura 46.

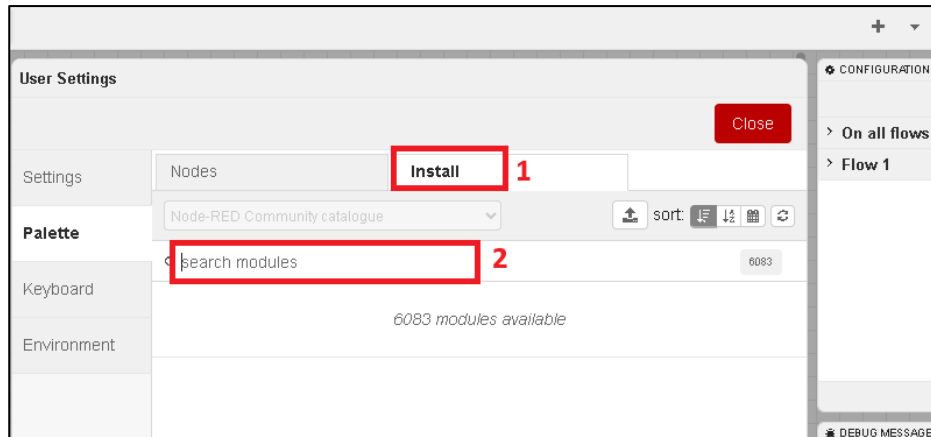


Figura 46. Instalación de las librerías necesarias para Node-RED

Lista de librerías usadas en Node-RED:

- **@flowfuse/node-red-dashboard:** es la versión más actualizada (dashboard 2.0) de la interfaz gráfica de Node-RED.
- **Node-red-contrib-telegrambot:** este es el nodo oficial para la conexión entre Node-RED y la aplicación de mensajería Telegram.
- **Node-red-MQTT:** este es el nodo que hace posible conectarse a internet de manera inalámbrica con la aplicación The Things Network.
- **Node-red-dashboard:** Es la versión antigua (Dashboard 1.0) de la interfaz gráfica de Node-RED.

Después de instalar las librerías requeridas, se accede al apartado de la paleta de nodos para localizar los nodos que se van a usar, tal y como se observa en la figura 47.

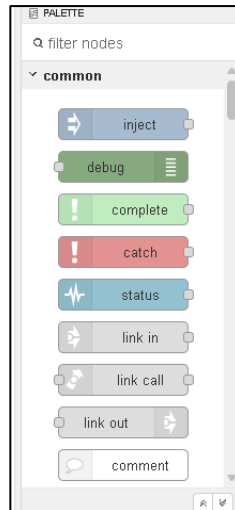


Figura 47. Sección de la paleta de nodos

Los nodos que se deben usar son:

MQTT in: nodo que se conecta a internet para recibir las tramas de los datos desde la aplicación de The Things Network.

JSON: es el nodo que traduce estas tramas de datos y las convierte en variables independientes.

Switch o function: es el cerebro del sistema, donde se ejecuta la programación de las reglas lógicas para detectar anomalías (sobrecorriente, voltaje no deseado y fugas).

Ui-gauge y Ui-chart: corresponden a los nodos gráficos para el panel de control.

Plantilla: este es el nodo que escribe el texto de la alerta combinando las palabras con los datos reales de las anomalías adquiridas.

Telegram-Bot-output: es el nodo que se encarga de recibir el mensaje que brinda el template y de enviarlo como notificación inmediata al teléfono móvil por Telegram.

Definición de nodos

Para detectar anomalías, es indispensable establecer una configuración eficiente de los nodos que se van a usar en el servidor Node-RED.

3.5.7 Definición de los nodos por MQTT

Nodo MQTT in

La función principal de este nodo es escuchar el canal (topic) del dispositivo inalámbrico y recibir en tiempo real las tramas de datos transmitidas por el nodo CubeCell AB02 a través del gateway. También es el nodo que se conecta a la aplicación The Things Network. Se encuentra en la parte izquierda de la interfaz, como se muestra en la figura 48.

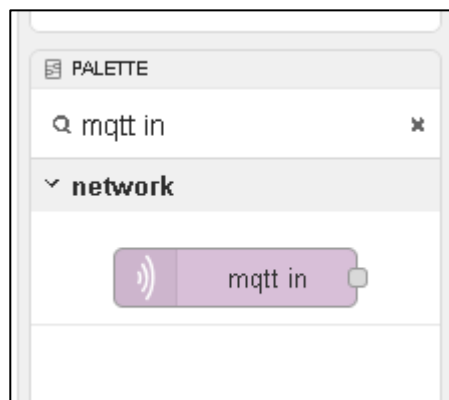


Figura 48. Búsqueda del nodo MQTT in

Para configurar este nodo, haga doble clic sobre él para abrir una ventana donde se determinarán los parámetros del nodo MQTT in. Tal y como se muestra en la figura 49.

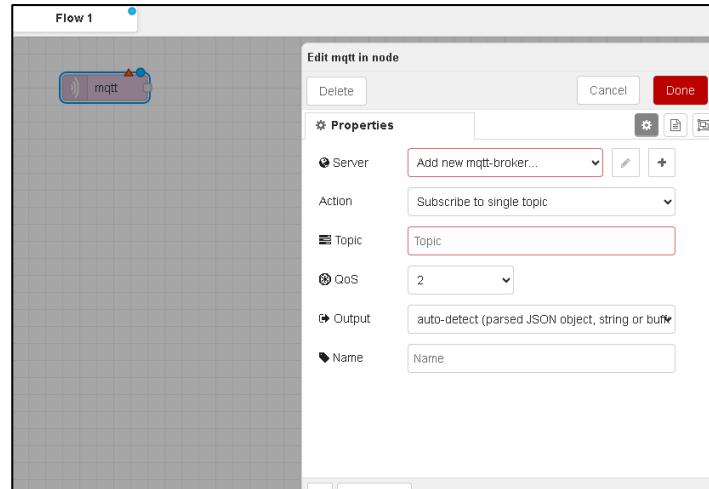


Figura 49. Entorno de configuración de MQTT in

Para establecer el servidor adecuado y garantizar la conexión entre Node-RED y TTN, es importante buscar la casilla Server para configurar el nodo. Se da clic en el símbolo “+”, donde aparecerá una ventana flotante. En la casilla Server se coloca “nam1.cloud.thethings.network”, que es la dirección oficial para la región de las Américas. En la casilla Port, hay que asegurarse de que esté asignado el número de puerto abierto 1883, como se muestra en la figura 50.

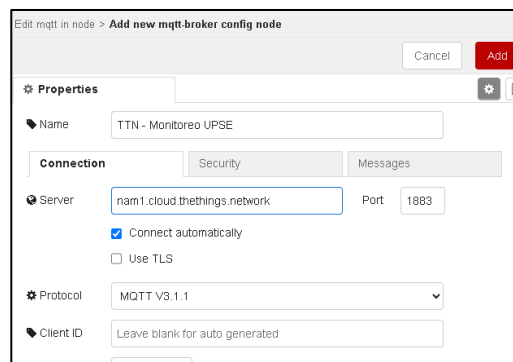


Figura 50. Configuración del servidor y el puerto

En la casilla Security del nodo MQTT in también se debe establecer la configuración correspondiente con las credenciales oficiales que genera la plataforma The Things Network, tal y como se muestra en la figura 51.

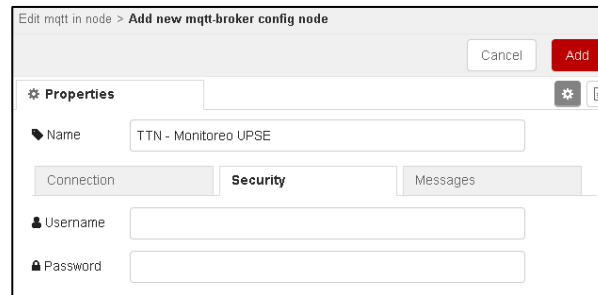


Figura 51. Configuración de seguridad del nodo

Las credenciales del nodo MQTT se adquieren al acceder a The Things Network y entrar en la sección de aplicaciones, donde se visualizará una opción llamada “MQTT”, como se observa en la figura 52.

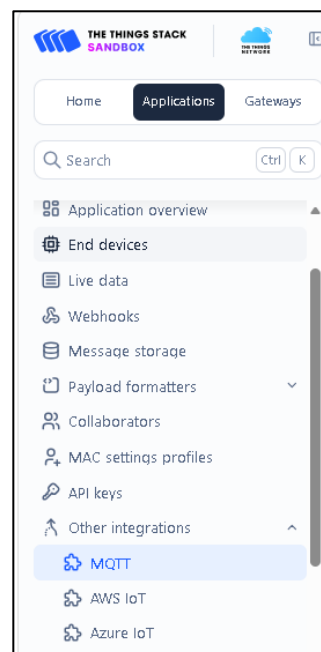


Figura 52. Opción MQTT en TTN

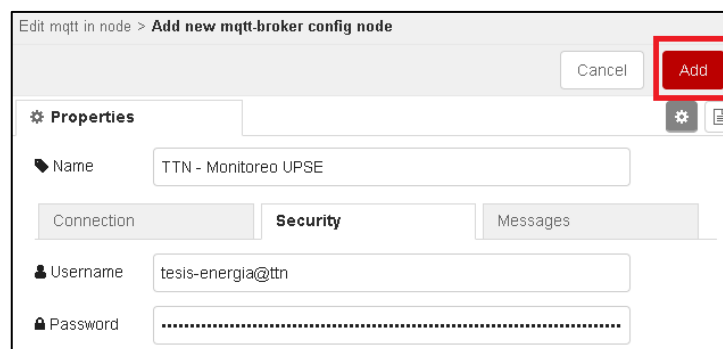
Al hacer clic en la opción MQTT, se abrirá una nueva sección donde, en el apartado “Connection information”, se encuentran las credenciales requeridas, que corresponden al usuario y la contraseña, como se muestra en la figura 53.



The screenshot shows the 'Monitoreo UPSE' MQTT configuration page in the TTN portal. The page title is 'Monitoreo UPSE' with a sub-label 'ID: tesis-energia'. Below the title, there is a brief explanation of MQTT and links for further resources. The 'Connection information' section contains fields for 'MQTT server host', 'Public address', 'Public TLS address', 'Username', and 'Password'. The 'Username' field is filled with 'tesis-energia@ttn' and the 'Password' field is filled with a masked password. A red rectangle highlights the 'Username' and 'Password' fields.

Figura 53. Credenciales de MQTT en TTN

Después, se añaden los datos de las credenciales adquiridas por TTN a los campos “Usuario” y “Contraseña” del servidor Node-RED y se hace clic en “Add”, tal y como se muestra en la figura 54.



The screenshot shows the 'Edit mqtt in node > Add new mqtt-broker config node' dialog in Node-RED. The 'Properties' tab is active, showing the 'Name' field filled with 'TTN - Monitoreo UPSE'. Below the 'Properties' tab, there are three sub-tabs: 'Connection', 'Security', and 'Messages'. The 'Security' sub-tab is active, showing the 'Username' field filled with 'tesis-energia@ttn' and the 'Password' field filled with a masked password. A red rectangle highlights the 'Add' button in the top right corner.

Figura 54. Credenciales en MQTT in

El siguiente paso es asignarle un tema, pero para ello hay que tener en cuenta primero las directrices de la plataforma TTN. La imagen 55 muestra cómo se escoge el segundo modelo, ya que se recogerán datos de uplink o subida.

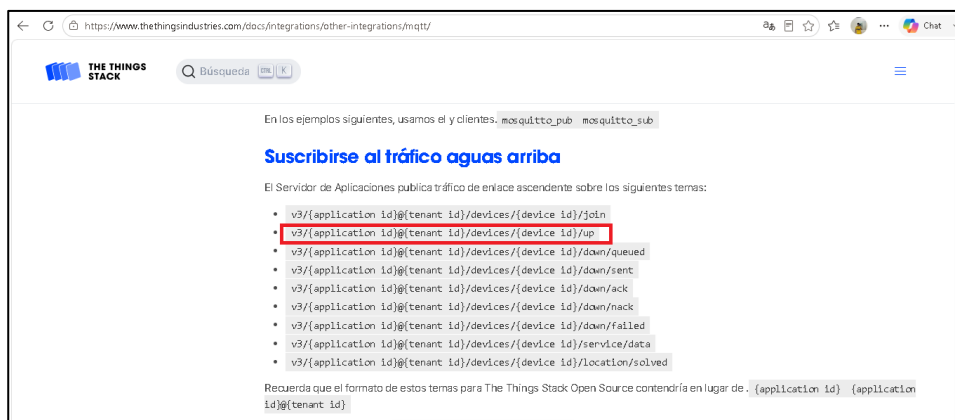


Figura 55. Directrices para el modelo Topic en TTN

Una vez asignado el tema que se va a usar, se reemplazan nuestros datos. En este caso, el modelo que se va a usar es (v3/ {application id} @ {tenant id} /devices/ {device id} /up) y los datos de la aplicación y el nodo CubeCell AB02 son v3/tesis-energia@ttn/devices/nodo-upse/up, tal y como se muestra en la figura 56. De esta forma, se completa el último apartado que requiere el nodo MQTT in.

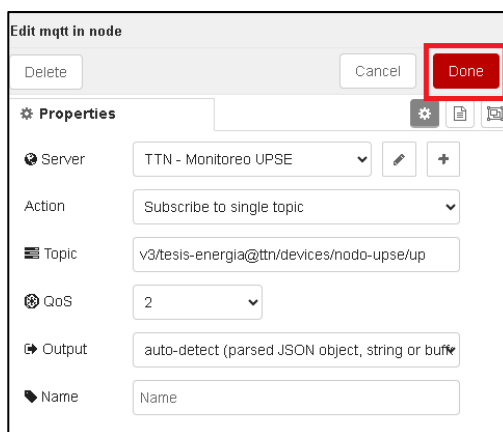


Figura 56. Nodo MQTT in en el apartado final

Nodo JSON o de traducción

Se requiere el uso de este nodo porque los datos del módulo CubeCell AB02 se transportan por Internet de forma encriptada y comprimida como una cadena de texto plano. La función del nodo JSON es convertir este texto plano en un objeto organizado, es decir, organizar estos datos y prepararlos para su representación gráfica.

Primero, hay que buscar el nodo en la parte izquierda, en la sección “Paleta”, y arrastrarlo a la ventana de Flow. A continuación, se hace clic en el cuadro gris de la salida del nodo MQTT in y en la entrada del nodo JSON, tal y como se muestra en la figura 57.

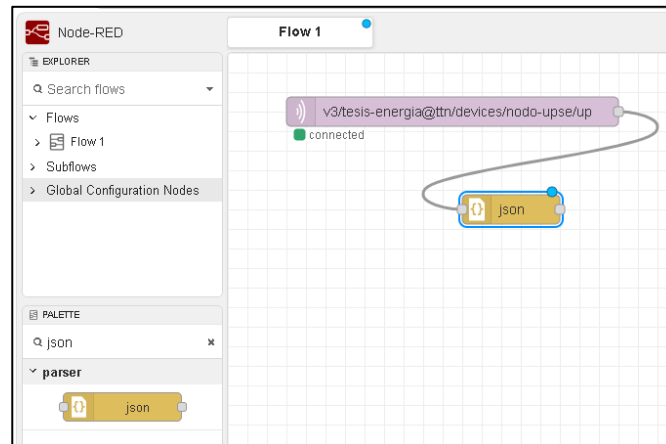


Figura 57. Nodo JSON

Nodo debug

Para verificar que la comunicación se está llevando a cabo correctamente y que los datos están llegando de TTN, se utilizará el nodo Debug. Busque en la sección “Paleta” el nodo que necesita (1) y conecte la salida del nodo JSON con la entrada del nodo Debug mediante el cable virtual. Finalmente, haga clic en Deploy, que se encuentra en la parte superior derecha (2), tal y como se muestra en la figura 58.

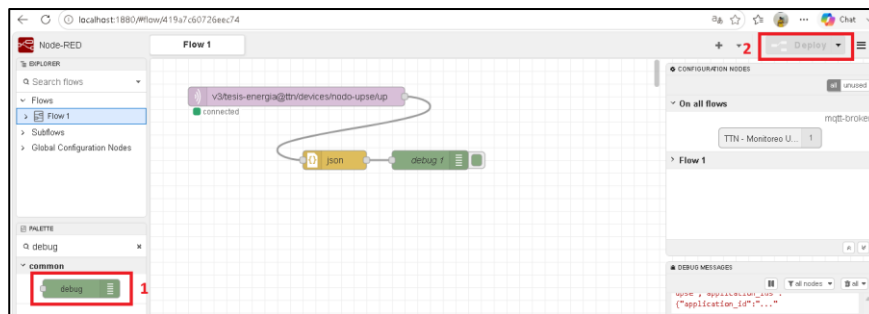


Figura 58. Nodo debug

Para verificar los mensajes de depuración o abrir la consola de visualización, haga clic en el icono con forma de insecto, tal y como se observa en la figura 59. De esta manera, se confirma que los datos binarios que el CubeCell transmite por ondas de radio están viajando hacia el gateway y cruzando la nube The Things Network, y que están entrando en tiempo real al Node-RED local.

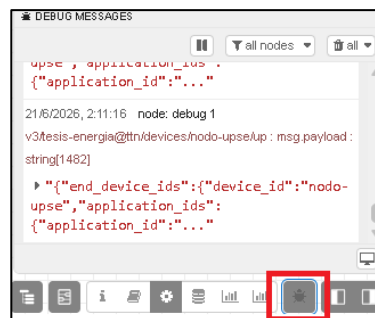


Figura 59. Ventana de mensajes de depuración

Nodo change

El siguiente paso es incluir un nodo llamado change para filtrar los datos, ya que el mensaje contiene muchos metadatos de red. Para conectar los relojes o indicadores del panel de control y configurar las alertas de anomalías eléctricas (Telegram), es necesario utilizar un decodificador.

Una vez arrastrado el nodo change a la ventana de Flow, se coloca entre el nodo JSON y el nodo debug, como se observa en la figura 60.

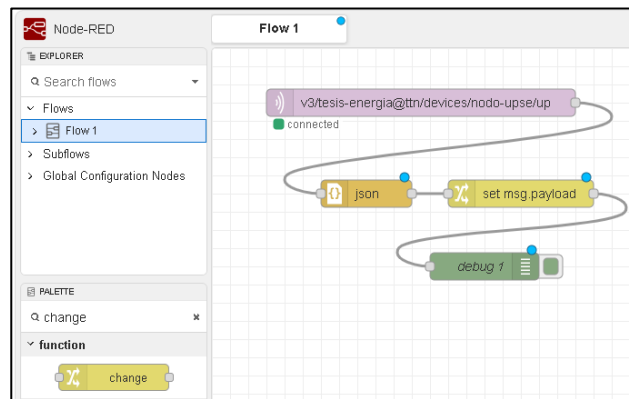
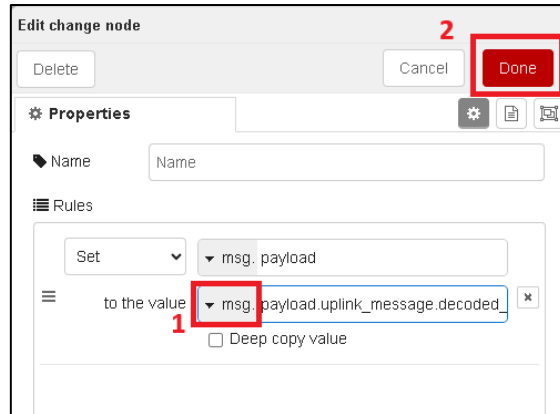
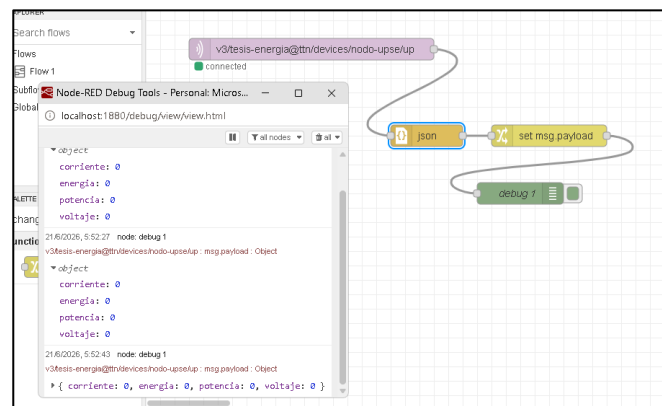


Figura 60. Nodo change

Después, haga doble clic en el nodo Change para configurarlo. Diríjase a la casilla o sección llamada “to the value”, situada al lado. En el menú desplegable, se deberá escoger el formato adecuado, tipo msg (1). También se asignará esta ruta en el apartado vacío, tal y como se muestra en la figura 61.



Una vez completados estos pasos, haga clic en Deploy (2). De esta forma, se eliminarán los mensajes de texto largo que se mostraban anteriormente y se verá únicamente una estructura limpia de las cuatro variables numéricas, como se muestra en la figura 62.



3.5.8 Diseño del panel de control gráfico con los nodos ui-gauge y ui-charge

Nodo ui-gauge

Es el nodo establecido para los relojes de agujas que representarán cada variable eléctrica.

Una vez arrastrado a la ventana Flow, se procede a configurarlo. Como se observa en la figura 63.

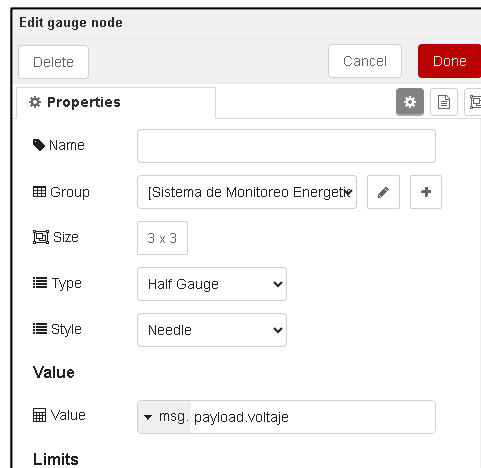


Figura 63. Ventana de configuración del nodo gauge

A continuación, se arrastra un cable virtual desde la salida del nodo Change hasta la entrada del nodo UI-Gauge. Es importante saber que el nodo Change entrega un paquete de cuatro variables (voltaje, corriente, potencia y energía), pero el nodo Gauge solo puede leer una variable a la vez.

Al hacer doble clic en el nodo ui-gauge, podemos configurar la asignación de una variable a cada gauge, de tal forma que nos ubicamos en la opción value, donde payload se modificará a “payload. Voltaje” (1). En el apartado Limits (2), se rellenan los campos de límite de escala: en Min., 0, y en Máx., 150. Estos son valores reales para una red eléctrica residencial de Ecuador. En el apartado Segments (3), se configuran los colores de alerta: verde (rango seguro), con un valor de 0, que se mantendrá en 0 V hasta el siguiente umbral; amarillo (subtensión/alerta), con un valor de 108, que se pondrá amarillo si el voltaje baja de 108 V, y rojo (sobretensión/crítico), con un

valor de 127; es decir, si el voltaje sube de 127 V, la aguja cambiará a rojo como indicador visual de fallo. Tal y como se muestra en la figura 64.

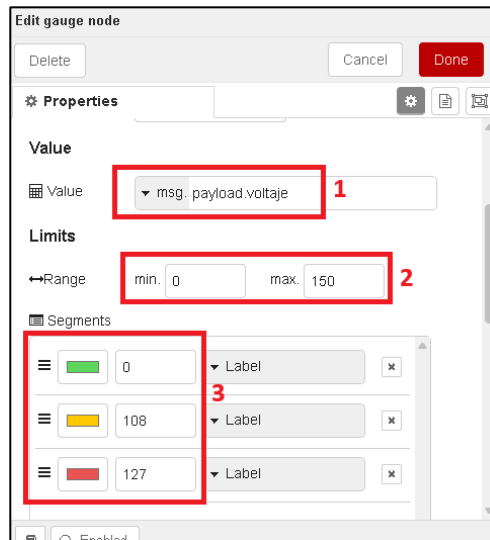


Figura 64. Nodo gauge asignado para voltaje

En el apartado de etiquetado, se rellenan sus campos. En la casilla Label se coloca el voltaje local y en Units se le asigna V, como se muestra en la figura 65.



Figura 65. Asignación de variable de voltaje

Antes de clonar todo este proceso para las dos variables restantes, hay que configurar el nombre de group y el de page del dashboard. Para ello, en la misma ventana del UI-Gauge se busca la casilla denominada Group, donde se da clic en el lápiz. Se abrirá otra pestaña, se busca la casilla

Page para dar clic nuevamente en el lápiz. Una vez que la nueva pestaña se despliega en la sección de Nombre, se coloca “Monitoreo energético UPSE”, como se observa en la figura 66. Por último, se da clic en Update.

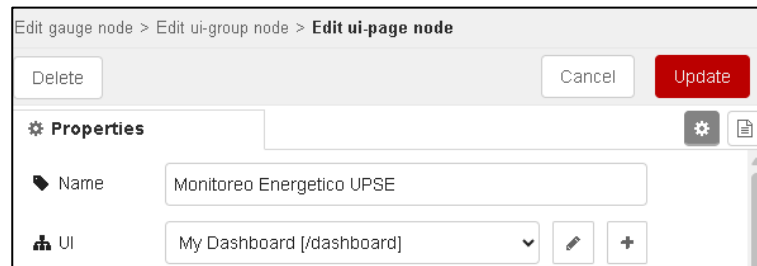


Figura 66. Asignación de nombre a nodo ui-page

Después de hacer clic en Update, se retrocederá a una pestaña llamada Edit UI-Group Node, donde se le asigna el nombre de “Variables en tiempo real”, tal y como se muestra en la figura 67. Por último, se hace clic en Update.

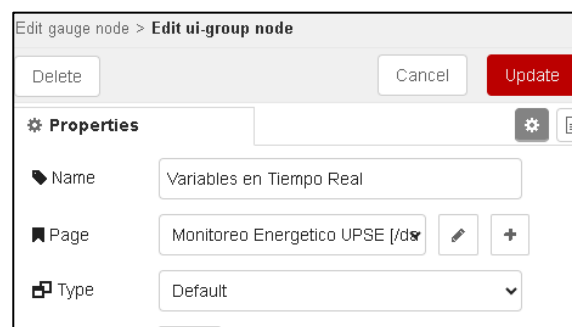


Figura 67. Asignación de nombre al nodo ui-group

Una vez completada esta configuración, se clona el nodo de voltaje para las dos variables restantes (corriente y potencia). Para ello, se pulsa Ctrl+C para copiar y se añaden dos nuevas secciones con los nombres “corriente” y “potencia”, como se muestra en la figura 68.

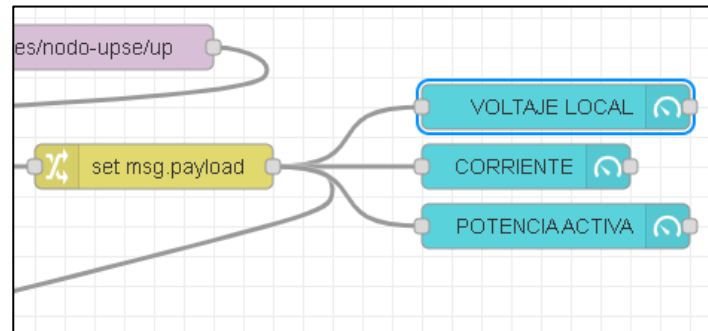


Figura 68. Añadir nodos para las variables de corriente y potencia

A continuación, se procede a establecer cada nodo con sus respectivas variables para que se presenten los datos en tiempo real en el panel de control. Se configura el nodo de corriente como se muestra en la figura 69. En la casilla value se añade “.corriente” (1), después se añaden los valores en la casilla limits, que son entre 0 y 15. Estos representan la escala excelente para consumo residencial. En la casilla segments (2) los valores que se añaden a los colores verde, amarillo y rojo representan el indicador visual de sobrecarga eléctrica. Como último paso, en la casilla labelling (3) se añaden las unidades y el nombre que llevará el nodo o medidor. En este caso, «corriente» y la unidad «A» de amperios. A continuación, se da clic en done.

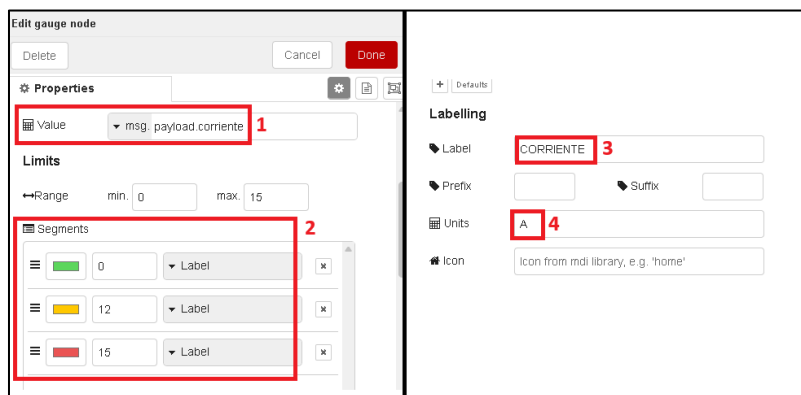


Figura 69. Configuración del nodo corriente

Continuando con la configuración de nodos, se rellenan los campos necesarios para mostrar los datos en el medidor que presenta los valores de potencia, como se muestra en la figura 70. En la casilla Value (1), se añade “Potencia”. En la casilla limits, se rellenan los campos con los valores de 0 a 2000, que representan una escala excelente para medir electrodomésticos. En la casilla segments (2), los colores indican operación eficiente, demanda elevada y sobredemanda térmica. Por último, en la casilla Labelling (3), se añade el nombre del medidor y la unidad, en este caso W, que significa watts. Para finalizar, se hace clic en “done”.

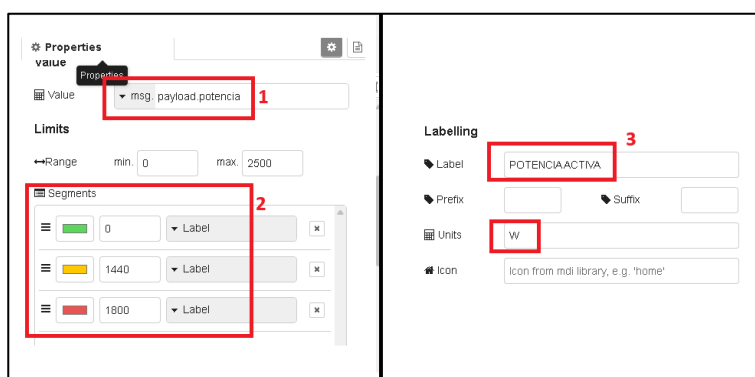


Figura 70. Configuración para nodo de potencia

Antes de incluir el nodo ui-chart en la interfaz de Flow, hay que utilizar cuatro nodos change, ya que así la gráfica que se desea presentar incluirá los datos lineales de las cuatro variables (voltaje, corriente, potencia y energía).

Esto se debe a que, si se conecta el nodo UI-chart con el nodo change ya existente llamado “set.msg. payload” en el diagrama de flujo, este nodo entrega las cuatro variables juntas y hace que la gráfica moderna de FlowUse intente dibujar todo automáticamente y mezcle las escalas. La solución es utilizar cuatro nodos Change independientes para cada variable y conectar su entrada con la salida del nodo “Change” ya existente. Como se observa en la figura 71.

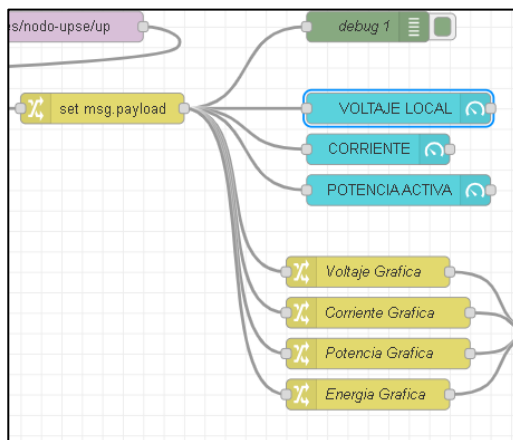


Figura 71. Nodos independientes

A continuación, hay que configurar cada nodo incluido para que tenga su variable asignada. Para ello, hay que hacer doble clic en el nodo y se abrirá una ventana flotante donde se rellenarán los campos correspondientes. En primer lugar, en la casilla To the value, se reemplaza msg. payload por “msg. payload. Voltaje” (1). A continuación, se hace clic en +Add (2). Después, aparecerá una nueva sección donde se definirá la etiqueta “msg. Topic” (3), para que la gráfica diferencie qué línea es voltaje, corriente, etc. Por último, hay que hacer clic en «Done». Como se

muestra en la figura 72. A este nodo le damos Ctrl+C para clonarlo y repetir los pasos, pero esta vez definiendo su respectiva variable.

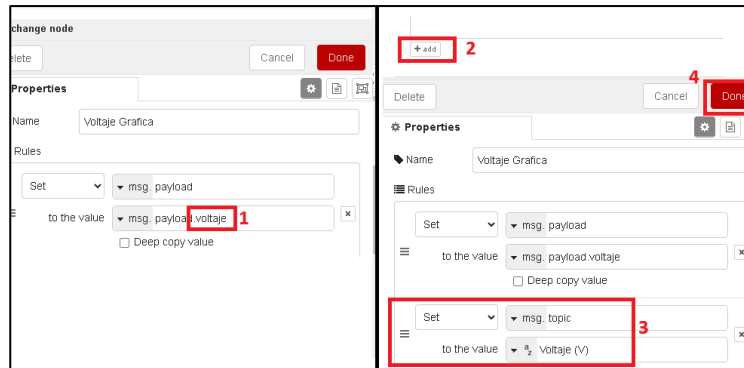


Figura 72. Configuración de la etiqueta msg. topic

Nodo ui-chart para representación gráfica

Posteriormente, se configuraron los nodos change para la gráfica y, mediante el cable virtual, se conectó la salida de estos con la entrada del nodo ui-chart. De esta forma, la gráfica podrá identificar cada variable por separado para su presentación. Como se aprecia en la figura 73.

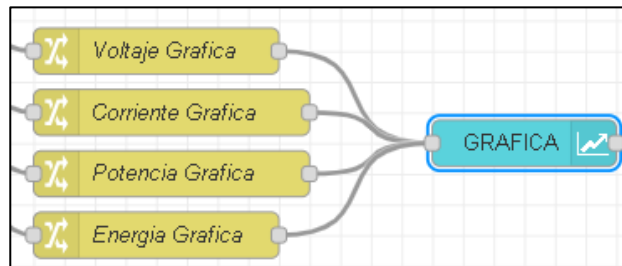


Figura 73. Conexión entre nodos

Después de realizar las conexiones, se debe configurar el nodo: hacer doble clic para abrir la ventana de configuración. Primero, hay que hacer clic en el icono “v”, donde se asignará el grupo ya creado (1). Segundo, hay que asignar el tiempo (2) para que la gráfica vaya reflejando los datos según este. Por último, hay que hacer clic en «Done», como se observa en la figura 74.

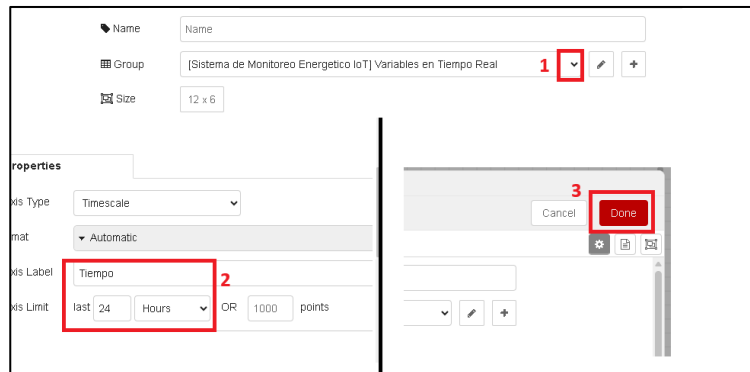


Figura 74. Configuración del nodo ui-chart

3.5.9 Aplicación Telegram

Chat bot – botfather

Para establecer comunicación entre Telegram y Node-RED, se requiere un bot que genere credenciales que sirvan como claves de seguridad e identidad del bot creado ante los servidores de Telegram.

El primer paso es escribir @botfather en la lupa de búsqueda de Telegram. Este bot es el encargado de crear y administrar el bot de alertas de la plataforma Telegram. Como se muestra en la figura 75.

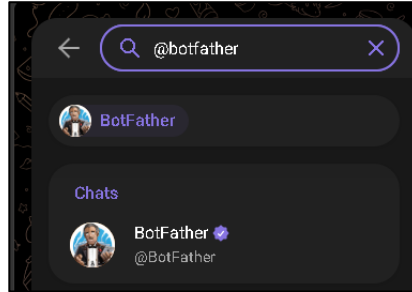


Figura 75. Chatbot father

Dentro del chat, se hace clic en Start o Iniciar, lo que hace que se despliegue un mensaje con múltiples opciones. La opción que se seleccionará es Newbot. En la sección Message, se escribe /newbot y se envía. Como se muestra en la figura 76.

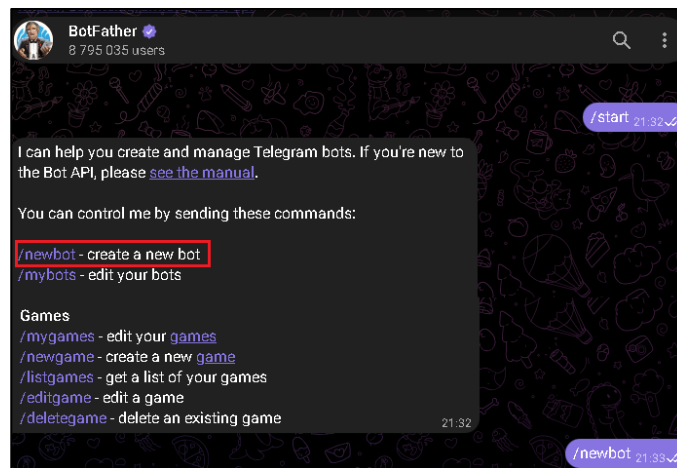


Figura 76. Creación del chatbot personal

A continuación, se deberán seguir los siguientes pasos para asignar un nombre de usuario al chatbot. Primero, se asigna el nombre del bot como “alertas energía upse” y, después, el usuario como “tesismedidorupse2026_bot”, donde es necesario colocar “_bot” al final del nombre. Además, Botfather responderá con un mensaje largo y se generará el token, que sirve para configurar Node-RED y establecer la comunicación entre las plataformas. En la figura 77 se observan cada uno de estos pasos y la credencial.

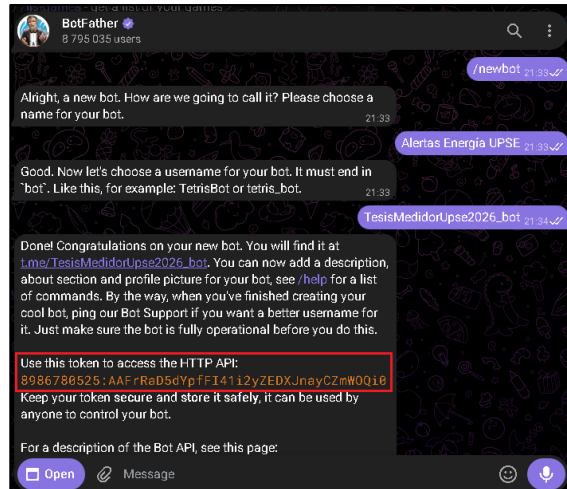


Figura 77. Creación de alertas de energía

En el mensaje largo anterior, nos asigna la dirección del chat del bot creado. Haz clic en ese enlace y, a continuación, haz clic en Iniciar para comenzar la conversación. Como se muestra en la figura 78.

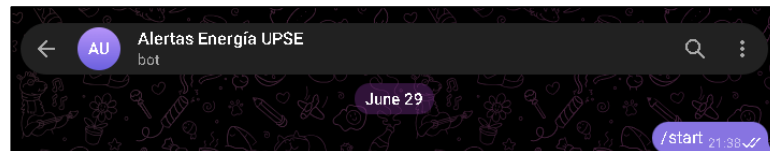


Figura 78. Chat bot creado

Posteriormente, también es necesario obtener el ID de usuario personal (chat ID). Para ello, hay que escribir @userinfobot en el buscador de Telegram. A continuación, se debe entrar en el chat y hacer clic en Iniciar. El mismo bot responderá de inmediato con los datos correspondientes. Tal y como se muestra en la figura 79.

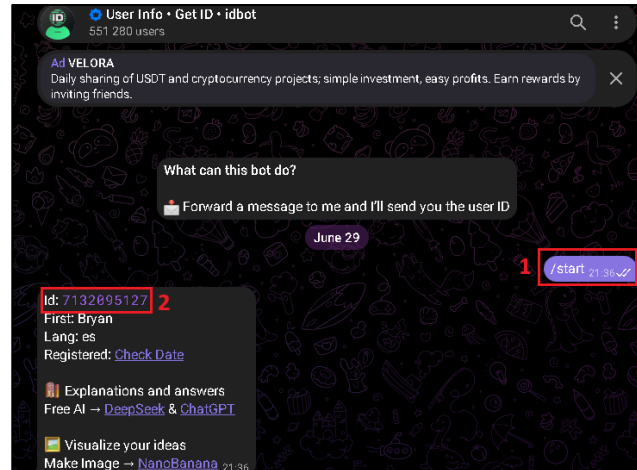


Figura 79. Credencial del ID del chat ID

Con estas credenciales se termina de realizar la configuración necesaria para enlazar los nodos en la estructura de flujo de Node-RED y poder comunicarse con Telegram.

Primero, en la parte izquierda, en la sección de paletas, buscar el nodo Switch y proceder a configurarlo. Tal y como se observa en la figura 80, el primer paso (1) es asignar un nombre al nodo “Filtros anomalías V”, el segundo paso (2) es añadir “.voltaje” en la casilla Property, el tercer paso (3) es reemplazar el signo “= =” de la casilla izquierda por el signo mayor que (>) y colocar “127” en la casilla derecha, el cuarto paso (4) es dar clic en Add, el quinto paso (5) es colocar el signo menor que (<) en la casilla izquierda y asignar el valor 105 en la casilla de al lado y, por último, el sexto paso (6) es dar clic en Done.

La conexión del nodo es la siguiente: la salida del nodo switch llamado “Set msg. Payload” se conecta con la entrada del switch recién creado.

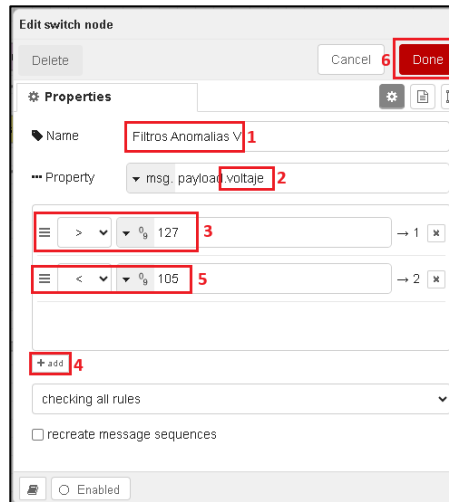


Figura 80. Configuración del nodo de filtro de anomalías V

Ahora, en el apartado Paleta, busca el nodo llamado “template” y arrastra dos nodos: el primero se establece para redactar las alertas altas y el segundo, para redactar las alertas bajas. Una vez culminado este paso, se configura cada nodo.

El primer nodo se define como “redactar alerta alta”. Este sirve para enviar mensajes de alerta. En primer lugar, en la caja de texto grande se borra el mensaje predeterminado y se coloca el código con el mensaje de alerta de sobretensión. En segundo lugar (2), en la casilla de salida se selecciona texto plano. En tercer lugar (3), se hace clic en Done para terminar la configuración del nodo. Como se observa en la figura 81.



Figura 81. Configuración del nodo de plantilla (sobretensión)

El segundo nodo de la plantilla se define como “redactar alerta baja” y se siguen los mismos pasos que en el nodo anterior, con la única diferencia de que el código indica que el mensaje es para alertas de caída de voltaje. Se puede ver en la figura 82.

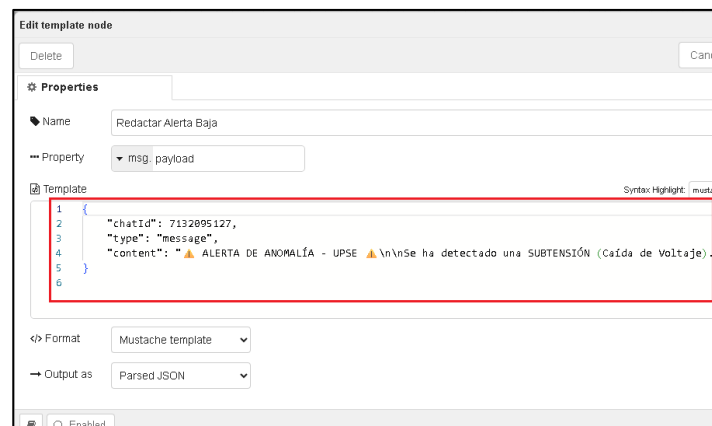


Figura 82. Configuración del nodo template subtensión

Posteriormente, una vez culminadas las configuraciones de los nodos Template, se usa el último nodo para la comunicación con Telegram. En la parte izquierda de la interfaz de Node-RED, en

la casilla Paleta, se busca el nodo Sender, que sirve para enviar datos, mensajes o archivos desde el flujo hacia un servidor externo (Telegram).

Configuración del nodo sender

Al hacer doble clic sobre el nodo, se abrirá una ventana flotante. Dar clic en la casilla donde está el signo “+”, como se observa en la figura 83.

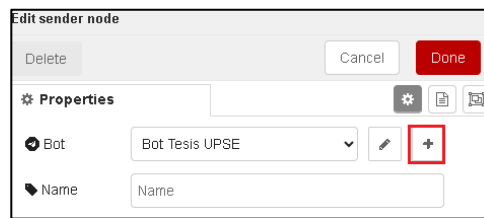


Figura 83. Node sender

Después de hacer clic en esa casilla, se abrirá otra ventana en la que se deben rellenar los campos principales con las credenciales oficiales obtenidas por Telegram, como se observa en la figura 84. Primer paso (1): definir un nombre, segundo paso: pegar la credencial que se obtuvo por el chat bot al crear el bot personal, tercer paso (3): dar clic en update para guardar cambios.

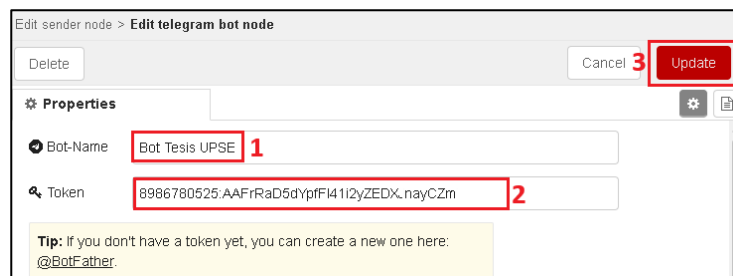


Figura 84. Credenciales en el nodo sender

3.5.10 Configuración del historial de registro de anomalías

Agregar un apartado donde se registre el historial de las anomalías, no solo en tiempo real, sino también para la visualización del panel de control. En primer lugar, en la sesión Paleta, buscar el nodo function, que en este apartado sirve para el procesamiento y la estructuración de la información.

Posteriormente, al hacer doble clic en el nodo function para configurarlo, se abrirá una pestaña o ventana flotante en la que se rellenarán las casillas vacías. Tal y como se muestra en la figura 85. El primer paso (1) es definir el nombre del nodo como “Formatear fila tabla”, el segundo paso (2) es pegar el código en la casilla On message, que corresponde al algoritmo de JavaScript para definir el historial de las anomalías, y el tercer paso (3) es hacer clic en Done para guardar los cambios.

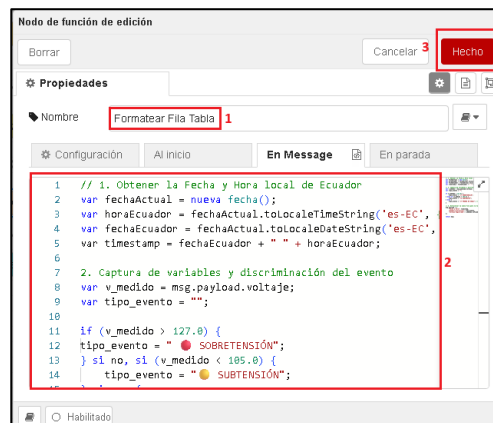


Figura 85. Configuración función para historial de anomalías

Como nodo final, se usa el nodo template de la sección correspondiente a la librería de dashboard 2.0 (nodo de color azul). Este nodo permite crear componentes visuales totalmente personalizados y programar elementos con HTML, CSS y JavaScript.

La configuración de este nodo se muestra en la figura 86. Primer paso (1) se define un nombre al nodo, paso (2) se selecciona el grupo donde están definidas las variables en tiempo real, paso (3) pegar el código y paso (4) dar clic en done para guardar cambios.

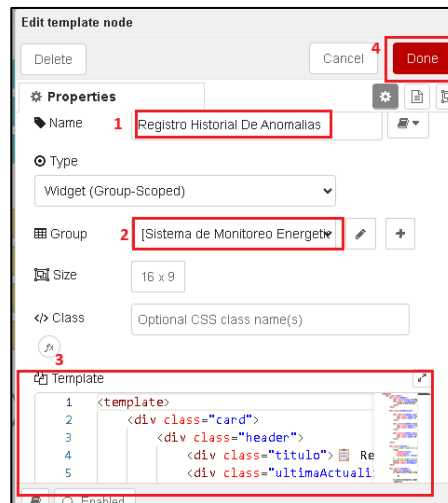


Figura 86. Configuración del historial

3.5.11 Diagrama de flujo final en Node-RED

La figura 87 muestra el diagrama de flujo de Node-RED, que implementa un sistema robusto de monitorización IoT para la gestión energética. Este sistema captura lecturas en tiempo real desde la aplicación The Things Network (TTN) mediante un nodo MQTT y las procesa a través de un nodo JSON para extraer métricas críticas, como el voltaje local, la corriente y la potencia activa. Esta información se muestra simultáneamente en una interfaz gráfica formada por indicadores de aguja y una gráfica lineal histórica, y alimenta un módulo lógico de supervisión (filtros de anomalías V), encargado de clasificar las desviaciones en el suministro. Ante cualquier anomalía, el sistema genera de manera autónoma notificaciones escritas que se distribuyen mediante un bot de Telegram. Paralelamente, formatea y almacena registros del incidente en una tabla interactiva personalizada con el nodo UI-template. De este modo, se logra una solución integral de monitorización, alerta temprana y auditoría de datos.

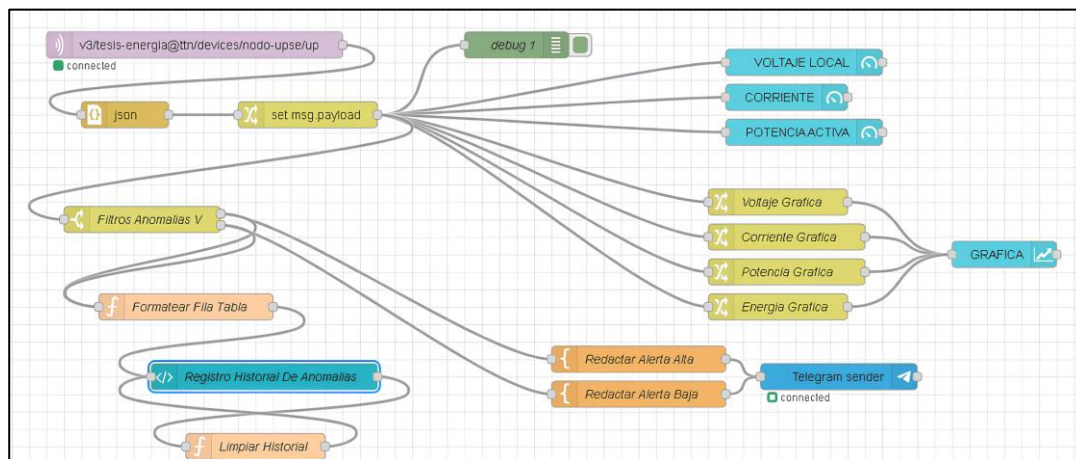


Figura 87. Diagrama de flujo final

Capítulo 4

4.1 Resultados

Con el propósito de verificar el funcionamiento del prototipo desarrollado, se realizaron distintas pruebas experimentales en cada uno de los niveles que conforman el sistema de monitorización de variables eléctricas. Estas evaluaciones permitieron verificar el funcionamiento de los componentes de hardware y software implementados, así como su integración, con el fin de comprobar el cumplimiento de los objetivos de la investigación. En los apartados siguientes se exponen los resultados alcanzados durante la validación de cada uno de los elementos que conforman el prototipo y su funcionamiento como sistema integral.

4.1.1 Componentes del prototipo

El prototipo desarrollado incorpora los elementos necesarios para la adquisición, el procesamiento y la transmisión de variables eléctricas, que permiten el monitoreo remoto del consumo energético mediante la tecnología LoRaWAN. El diseño está formado por módulos electrónicos que funcionan de manera coordinada para medir el voltaje, la corriente, la potencia y la energía consumida por una carga eléctrica, y enviar datos a una plataforma de monitorización en la nube.

Los sensores que conforman la etapa de adquisición de datos son el sensor de voltaje ZMPT101B y el sensor de corriente SCT-013, ambos conectados al microcontrolador ESP32. Este dispositivo procesa las señales de los sensores, realiza los cálculos correspondientes de las variables eléctricas y prepara la información para enviarla. Posteriormente, los datos se envían al módulo Heltec CubeCell AB02 mediante el protocolo de comunicación I2C, que incorpora tecnología LoRa y actúa como nodo transmisor de la red LoRaWAN.

Por último, el CubeCell AB02 envía los datos al gateway SenseCAP M2, que se encarga de recibir los paquetes de datos y reenviarlos a The Things Network (TTN) para que se guarden y se visualicen. Una vez integrados estos componentes, se dispone de un sistema capaz de supervisar de forma remota el consumo energético, haciendo uso de las ventajas de las redes LPWAN, que ofrecen comunicaciones de largo alcance y bajo consumo energético. La figura 88 muestra el prototipo implementado y la distribución de los componentes que forman el sistema.

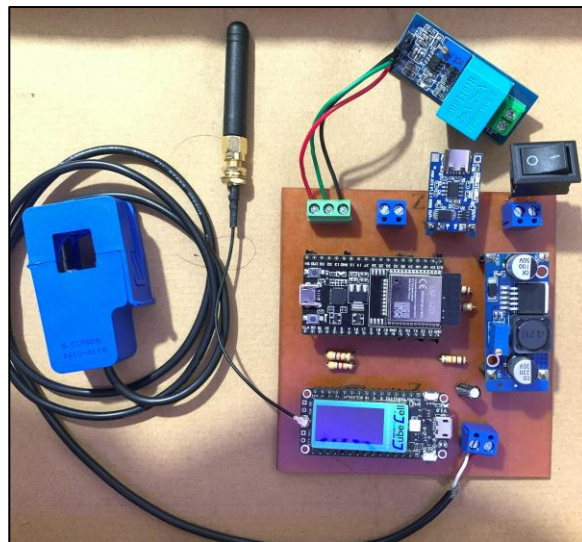


Figura 88. Implementación del prototipo

La figura 89 muestra el prototipo desarrollado para el monitoreo de variables eléctricas y su integración con la infraestructura de comunicación LoRaWAN. El prototipo está diseñado en una tarjeta de circuito impreso (PCB) para garantizar un diseño eficiente, ordenado y fiable, y así reducir el ruido eléctrico y mejorar la estabilidad de las mediciones. A un lado se encuentra el gateway SenseCAP, aparato encargado de recibir los datos enviados por el módulo CubeCell AB02 mediante la red LoRaWAN. Finalmente, al otro lado se encuentra la antena que, junto con el gateway, permite una comunicación eficiente y de largo alcance.



Figura 89. Prototipo implementado, gateway SenseCAP M2 y antena

4.2 Resultado de las pruebas de funcionamiento

4.2.1 Validación del rendimiento

La figura 90 muestra el registro de los paquetes LoRaWAN gestionados por el gateway SenseCAP M2 en las pruebas de funcionamiento del sistema de monitorización de variables eléctricas. En la gráfica se observan los distintos intervalos de transmisión en los que el gateway recibió la información enviada por el módulo CubeCell AB02 y procesó la comunicación dentro de la red LoRaWAN. Las barras de color verde representan los paquetes recibidos (RX), las barras azules los paquetes transmitidos (TX) y los valores en color rojo representan el total de paquetes procesados en cada intervalo de tiempo.

Los paquetes recibidos y transmitidos mantienen un comportamiento consistente a lo largo de las pruebas, lo que indica que la comunicación entre el nodo y el gateway es estable. Este resultado, que se observa en la gráfica, confirma que la información generada por el prototipo fue recibida y retransmitida correctamente, lo que garantiza un enlace fiable para el envío de las variables eléctricas a la plataforma de monitorización.

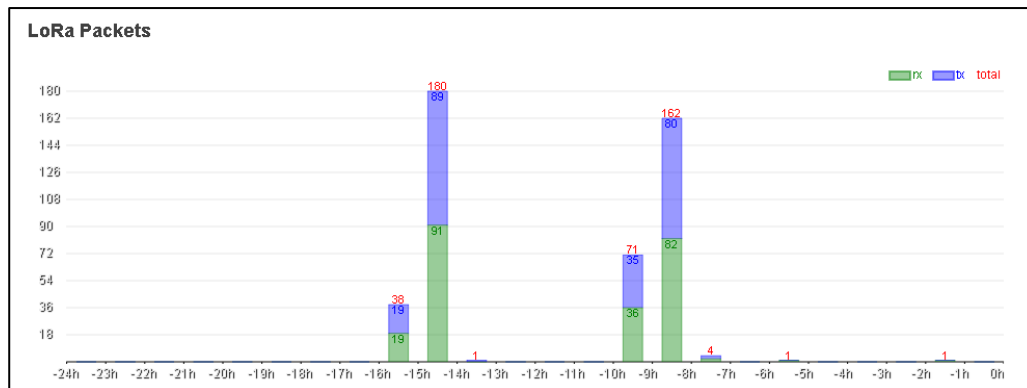


Figura 90. Registro de paquetes LoRaWAN

La figura 91 muestra el registro de mensajes intercambiados entre el gateway SenseCAP M2 y el nodo CubeCell AB02 durante las pruebas de funcionamiento del sistema en The Things Network (TTN). En el historial se observa que se han recibido mensajes uplink y que se han enviado mensajes downlink, lo que evidencia una comunicación bidireccional entre ambos dispositivos. Además, aparecen otros parámetros, como el contador de trama, el RSSI y el SNR, que permiten valorar la calidad del enlace LoRaWAN.

Los valores obtenidos durante las pruebas muestran una comunicación estable y continua, reflejada en el incremento secuencial del contador de tramas y en la recepción correcta de los mensajes transmitidos por el nodo. Además, los niveles de RSSI y SNR obtenidos se mantuvieron en rangos adecuados, lo que confirma un enlace confiable entre el prototipo y el gateway, que asegura la transmisión de las variables eléctricas a la plataforma TTN sin inconvenientes.

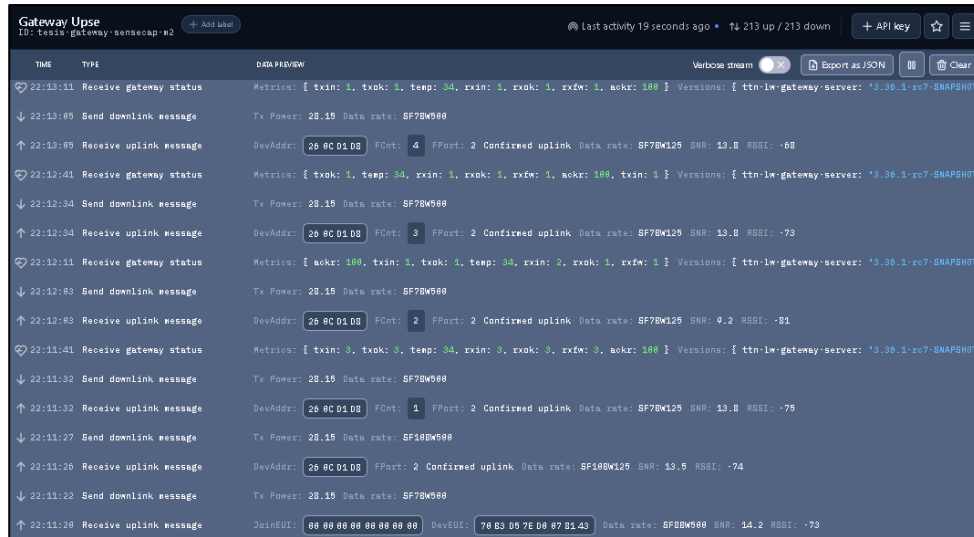


Figura 91. Registro de mensajes uplink y downlink en The Things Network

En la figura 92 se puede ver el panel de depuración de Node-RED, donde se visualizan en tiempo real los datos recibidos desde The Things Network mediante el protocolo MQTT. En el registro se puede observar que cada mensaje lleva las variables eléctricas medidas por el prototipo, como el voltaje, la corriente, la potencia y la energía, que son procesadas correctamente por el flujo desarrollado.

Los valores se actualizan de forma continua a medida que se reciben nuevos mensajes, lo que indica una comunicación estable entre TTN y Node-RED. De esta manera, se confirma que la información que envía el prototipo llega completa a la plataforma de procesamiento, lo que permite monitorizar de manera fiable las variables eléctricas en tiempo real.

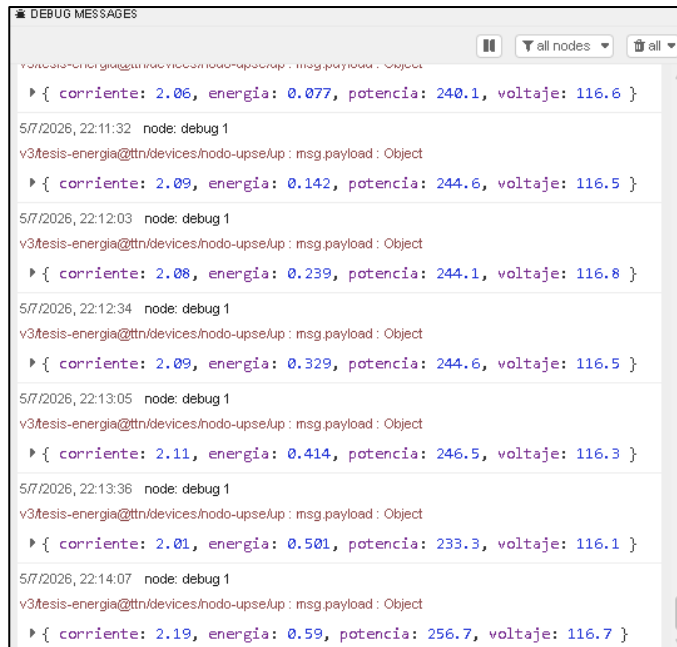


Figura 92. Panel de depuración de Node-RED con recepción de variables eléctricas desde TTN

En la figura 93 se pueden ver las notificaciones generadas por el bot de Telegram durante las pruebas de funcionamiento del sistema. Si el voltaje medido supera el nivel de sobretensión o desciende por debajo del límite de subtensión establecido, Node-RED procesa la información recibida de The Things Network y genera automáticamente un mensaje de alerta que se envía al dispositivo móvil del usuario a través del bot configurado.

En las notificaciones se exponen las variables eléctricas que originaron las anomalías, como el valor del voltaje y la corriente registrados en el momento de la detección. Para ello, el flujo de Node-RED utiliza un nodo switch que identifica la condición de sobretensión o subtensión, seguido de dos nodos template que se encargan de construir el mensaje correspondiente. Finalmente, el nodo Telegram Sender envía la alerta. Los resultados obtenidos muestran que las anomalías se notifican de forma automática y puntual, lo que permite al usuario conocer el estado de la red eléctrica en tiempo real.

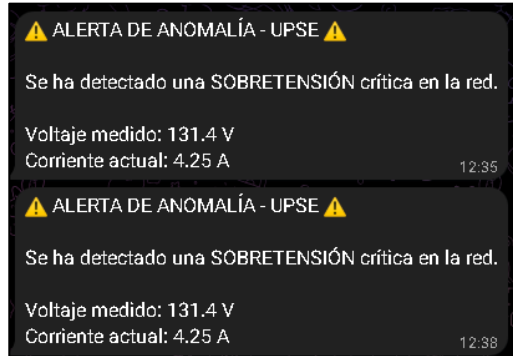


Figura 93. Notificaciones de anomalías enviadas mediante el bot de Telegram

La figura 94 muestra el panel de control desarrollado en Node-RED para el monitoreo de las variables eléctricas en tiempo real obtenidas por el prototipo. La interfaz permite observar mediante indicadores analógicos el voltaje, la corriente y la potencia, así como una gráfica que muestra el comportamiento de estas variables, incluida la energía consumida durante la operación del sistema.

El panel de control también incluye un módulo para el registro histórico de anomalías, donde se almacenan los eventos de sobretensión o subtenensión detectados, lo que permite una fácil consulta y gestión de los mismos. Los resultados obtenidos demuestran la correcta integración del sistema, ya que permite supervisar las variables eléctricas y las anomalías detectadas desde una única interfaz de monitorización.



Figura 94. Presentación del dashboard final

4.3 Consumo energético del prototipo

Con el fin de estimar el consumo energético del prototipo construido, se analizó el sistema de monitoreo de variables eléctricas, compuesto por los principales componentes electrónicos. Para ello, se consideró el consumo medio de cada dispositivo en su funcionamiento normal, incluyendo la adquisición de datos, el procesamiento de la información y la transmisión inalámbrica por LoRaWAN. Esta evaluación permitió determinar la autonomía aproximada del sistema con una batería recargable de 3,7 V y 1100 mAh (véase la tabla 15).

Tabla 15

Consumo energético del prototipo

Componentes	Consumo
ESP32	80 mA
CubeCell AB02 (incluye OLED integrada y transmisión LoRaWAN)	35 mA
Sensor ZMPT101B	3 mA
Componentes Auxiliares	7 mA

Total

125 mA

Notas. Fuente: Elaboración propia.

Teniendo en cuenta el consumo del prototipo y la capacidad de la batería de litio de 1100 mAh, la autonomía teórica del sistema se calcula mediante la siguiente expresión:

$$\textit{Autonomía} = \frac{\textit{Capacidad de la batería}}{\textit{Consumo total del sistema}}$$

$$\textit{Autonomía} = \frac{1100 \textit{ mAh}}{125 \textit{ mA}}$$

$$\textit{Autonomía} = 8.8 \textit{ h}$$

Expresado en horas y minutos

$$0.8 \times 60 = 48 \textit{ min}$$

Por lo tanto,

$$\textit{Autonomía} = 8 \textit{ h } 48 \textit{ min}$$

Los resultados obtenidos muestran que el prototipo puede operar durante aproximadamente 8 horas y 48 minutos en condiciones normales. Esta estimación tiene en cuenta el procesamiento continuo de las variables eléctricas, la visualización permanente de la información en la pantalla OLED del CubeCell y la transmisión de datos mediante LoRaWAN cada cinco segundos, lo que demuestra que la batería seleccionada tiene la capacidad necesaria para garantizar el funcionamiento del sistema durante las pruebas realizadas.

Conclusiones

Se desarrolló con éxito un sistema de adquisición de variables eléctricas mediante la integración de los sensores SCT-013-000 y ZMPT101B junto con el microcontrolador ESP32, que permite medir y procesar correctamente las variables de voltaje, corriente, potencia y energía eléctrica. Los resultados de las pruebas realizadas demostraron que el sistema ofrece mediciones estables y apropiadas para el monitoreo del consumo energético.

La integración del nodo CubeCell AB02 y el gateway SenseCAP M2 permitió implementar con éxito la transmisión inalámbrica de datos mediante la tecnología LoRaWAN. La configuración realizada permitió establecer una comunicación fiable entre el prototipo y la infraestructura de red, de modo que las variables eléctricas se transmitieron continuamente hacia la plataforma de monitorización sin pérdidas significativas de información.

La integración del protocolo MQTT con Node-RED permitió desarrollar una plataforma de supervisión en tiempo real que puede visualizar las variables eléctricas, registrar el historial de anomalías y generar notificaciones automáticas por Telegram en caso de condiciones de sobretensión y subtensión. Esta integración facilitó el monitoreo remoto y mejoró la capacidad de respuesta ante posibles eventos eléctricos.

Las pruebas realizadas evidenciaron de manera integral el rendimiento del sistema, verificando la precisión de la adquisición de datos, la estabilidad de la comunicación LoRaWAN y la correcta visualización de la información del panel de control desarrollado. Los resultados obtenidos demuestran que el prototipo opera de manera eficiente y fiable, cumpliendo los objetivos propuestos y constituyéndose como una alternativa para el monitoreo remoto de variables eléctricas.

Recomendaciones

Se recomienda ampliar las funcionalidades del sistema incorporando nuevos sensores que permitan analizar otras variables relacionadas con la calidad de la energía eléctrica, como la frecuencia, el factor de potencia o la distorsión armónica. Esto ayudaría a obtener un análisis más completo del comportamiento de la red eléctrica y aumentaría el alcance de la solución desarrollada.

Asimismo, se sugiere implementar mecanismos de optimización energética que aprovechen los modos de bajo consumo que ofrecen los microcontroladores empleados. La incorporación de estrategias de ahorro de energía permitiría aumentar la autonomía del prototipo cuando funcione con baterías, lo que facilitaría su uso en aplicaciones que no disponen de una fuente de alimentación permanente.

Por último, se sugiere crear una plataforma web o una aplicación móvil que complemente el panel de control implementado en Node-RED, para permitir el acceso remoto a las variables eléctricas y al historial de anomalías desde cualquier dispositivo con conexión a internet. Esta mejora facilitaría la supervisión del sistema, incrementaría la accesibilidad a la información y ampliaría las posibilidades de implementación en entornos residenciales, comerciales e industriales.

Bibliografía

- [1] S. I. Ricardo Plúas y . A. S. Olaves Rosales, «Implementación de un sistema de alarma comunitaria aplicando tecnología LPWAN e IoT para el barrio 5 de Junio del cantón La Libertad.,» 14 Agosto 2025. [En línea]. Available: <https://repositorio.upse.edu.ec/items/12f602a7-0078-4ad2-8532-987ad788368d>.
- [2] M. O. Ramos Gallegos , «Estudio de la tecnología LoRaWAN mediante la implementación de un prototipo de medición de humedad y temperatura.,» 12 Enero 2022. [En línea]. Available: <https://bibdigital.epn.edu.ec/handle/15000/22036>.
- [3] A. D. Espinoza Altamirano, «Desarrollo de un prototipo de IoT basado en el módulo wifi y servidor web local en la plataforma raspberry-pi enfocado para personas con capacidades especiales de movilidad.,» 8 Agosto 2021. [En línea]. Available: <https://bibdigital.epn.edu.ec/handle/15000/21774?>.
- [4] E. Bazan, «Implementación de prototipo basado en IOT y WSN para iluminación inteligente en carreteras de la Ciudad de Guayaquil,» 2020. [En línea]. Available: <https://es.scribd.com/document/615058998/B-CINT-PTG-N-570-Valencia-Gonzalez-Douglas-Enrique-Sanchez-Soledispa-Selena-Thalia>.
- [5] E. J. Meza Mora y R. F. Del Rosario Litardo, «Diseño e implementación de una red IOT basado en lorawan para el uso de proyectos desarrollados por estudiantes de la Universidad de Guayaquil, con un sistema de control De acceso a la RED

- IOT,» 2021. [En línea]. Available: <https://repositorio.ug.edu.ec/items/74c6c83c-e02e-4230-8282-3717e1fe0bd5?>.
- [6] C. Delgado Tapia y W. Valencia Astudillo, «Diseño e implementación de prototipo IOT para el monitoreo remoto de la calidad del agua para la crianza de tilapias en estanques,» 2021. [En línea]. Available: <https://dspace.ups.edu.ec/handle/123456789/21427?>.
- [7] B. R. Castillo Yaguana, «Desarrollo de un sistema IoT para el monitoreo de variables eléctricas requeridas en Sistema de Gestión Energética.,» 4 Marzo 2024. [En línea]. Available: <https://dspace.unl.edu.ec/items/aa44487c-ddcf-4b5d-ab4d-9d6b131626ff>.
- [8] E. J. Yagual Solano, «Implementación de un sistema de recolección de residuos mediante la tecnología LPWAN e IoT para el sector de Ballenita.,» 2025 Agosto 14. [En línea]. Available: <https://repositorio.upse.edu.ec/items/e17c7b26-435a-4918-8809-aad4deac5592?q>.
- [9] E. F. Zambrano Cajias, «Desarrollo de un prototipo para el monitoreo y control de parámetros de un variador de frecuencia en tiempo real usando Gateway IOT.,» Agosto 2023. [En línea]. Available: <https://repositorio.utc.edu.ec/items/e1023a6b-1cf3-4512-bb14-9120026ae257>.
- [10] L. A. Villacís Morán, «Diseño e implementación de un sistema de monitoreo IoT de tensión y corriente para medir consumo eléctrico en tableros industriales,» 2023. [En línea]. Available: <https://dspace.espol.edu.ec/handle/123456789/58258?>.

- [11] D. A. López Freire, «Diseño e implementación de un prototipo IOT para monitoreo y control del consumo energético en redes de baja tensión para uso residencial.,» 2023. [En línea]. Available: <https://dspace.esPOCH.edu.ec/items/72c6bca7-c29a-4849-8990-172d7f23579f?>.
- [12] R. E. Sanches Macias y D. I. Baldeon Barrios, «desarrollo de un sistema de supervision de energia basado en IoT para la gestion de operacion y mantenimiento de un planta industrial,» 2022. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/22753/1/UPS-GT003784.pdf?>.
- [13] C. X. Garcia Alvarez y R. A. Lopez Solano , «Implementacion de Prototipo IoT para la monitorizacion ambiental y radiacion solar utilizando hardware de bajo costo y energia solar,» 2024. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/27419/1/UPS-GT004974.pdf?>.
- [14] M. J. Ramirez Prado, «Diseño y prototipo de un sistema de monitoreo de consumo eléctrico residencial utilizando una plataforma virtual de internet de las cosas,» 2020. [En línea]. Available: <https://www.dspace.espol.edu.ec/handle/123456789/56404?>.
- [15] IBM, «What is the Internet of Things (IoT)?,» 2024. [En línea]. Available: <https://www.ibm.com/think/topics/internet-of-things?>.
- [16] J. Schulze, «What Is the Internet of Things (IoT)? With Examples,» 22 Octubre 2025. [En línea]. Available: <https://www.coursera.org/articles/internet-of-things?>.

- [17] B. Barcelona, «Capas de la Arquitectura IoT: Fundamentos y Casos Prácticos de Implementación,» [En línea]. Available: <https://alphaenginyeria.com/capas-arquitectura-iot>.
- [18] J. Lopez Vicario y X. Vilajosana Guillen, «Estudio en detalle de LoraWAN. Comparacion con otras tecnologias LPWAN considerando diversos patrones de trafico,» Enero 2020. [En línea]. Available: <https://openaccess.uoc.edu/server/api/core/bitstreams/a539d1db-2bf9-4c01-a827-0f5cbade47be/content>.
- [19] E. J. Guananga Narváez y W. R. Vivas Díaz, «Comparación de protocolos de comunicación para internet de las cosas (IoT),» Septiembre 2022. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/23315/1/UPS%20-%20TTS948.pdf>.
- [20] A. Bassi, «Introducción a Protocolos IoT,» 27 Febrero 2021. [En línea]. Available: https://www.gotoiot.com/pages/articles/iot_protocols_intro/index.html.
- [21] A. Electronics, «Sensor de Voltaje AC ZMPT101B,» [En línea]. Available: https://avelectronics.cc/producto/sensor-de-voltaje-zmpt101b/?srsltid=AfmBOooCXO0_0-vgCO4CzK_f6EPX_cNAnAUmjXncOMyjpWXW3CwkCnG0.
- [22] N. Mechatronics, «Tutorial sensor de corriente AC no invasivo SCT-013,» [En línea]. Available: https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-sct-013.html.

- [23] F. J. Mínguez Carbellido, «Arquitectura Software para el despliegue,» 2022. [En línea]. Available: <https://zaguan.unizar.es/record/155670/files/TAZ-TFG-2022-1479.pdf>.
- [24] S. Studio, «Descripción general del SenseCAP M2 Multi-Platform Gateway,» 28 Agosto 2023. [En línea]. Available: https://wiki.seeedstudio.com/es/Network/SenseCAP_Network/SenseCAP_M2_Multi_Platform/SenseCAP_M2_Multi_Platform_Overview/.
- [25] Motoma, «Batería LiPo de alta densidad de 3,7 V 1500 mAh,» [En línea]. Available: <https://motoma.com/spa/high-density-battery/3.7v-high-density-lipo-battery-1500mah.html>.
- [26] T. Rutte, «TTN: qué es y cómo funciona dentro de LoRaWAN y IoT,» 21 Agosto 2021. [En línea]. Available: <https://thingsdata.es/knowledge-base-es/ttn-explicacion/>.
- [27] L. Aljundi, «Uso del software Arduino (IDE),» 16 Enero 2024. [En línea]. Available: <https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide/>.

Anexos

Anexo 1 – Código de microcontrolador ESP32 en Arduino IDE

```
#include "EmonLib.h"
#include <Wire.h>

EnergyMonitor emon;

// =====
// CALIBRACIÓN (Ajustada para ESP32 a 12 bits)
// =====
// Si notas desfases, ajusta estos números en tus pruebas reales
#define vCalibration 180.3
#define currCalibration 115.0

// =====
// VARIABLES
// =====
float kWh = 0;
unsigned long lastmillis = 0;
unsigned long previousMillis = 0;
const long interval = 5000;

// =====
// ESTRUCTURA DE DATOS (16 Bytes exactos)
// =====
struct DatosEnergia {
    float v;
    float i;
    float p;
    float e;
};

// Variable volátil para evitar corrupción de datos durante la interrupción
// I2C
volatile DatosEnergia datos;

// =====
// ENVIAR DATOS AL CUBECCELL
...
```

```

// =====
void enviarDatos() {
    // Se envían los 16 bytes de la estructura de forma directa
    Wire.write((byte*)&datos, sizeof(DatosEnergia));
}

void setup() {
    // =====
    // SERIAL
    // =====
    Serial.begin(115200);
    delay(1000);

    // =====
    // I2C ESCLAVO (Dirección 0x55)
    // =====
    Wire.begin(0x55);
    Wire.onRequest(enviarDatos);

    // =====
    // ADC 12 BITS
    // =====
    analogReadResolution(12);

    // =====
    // CONFIGURAR SENSORES
    // =====
    // Pin 34: Voltaje, Pin 35: Corriente
    emon.voltage(34, vCalibration, 1.7);
    emon.current(35, currCalibration);

    // Descargar lecturas basura iniciales
    for (int x = 0; x < 3; x++) {
        emon.calcVI(20, 2000);
        delay(500);
    }

    lastmillis = millis();
    Serial.println("ESP32 INICIADO Y ESPERANDO MASTER");
}

```

```

void loop() {
    unsigned long currentMillis = millis();

    // =====
    // ACTUALIZAR CADA 5 SEGUNDOS
    // =====
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        // Calcular muestras (20 ciclos, timeout de 2000ms)
        emon.calcVI(20, 2000);

        // Calcular tiempo exacto para la integración de kWh
        unsigned long tiempoTranscurrido = millis() - lastmillis;
        lastmillis = millis();
        float horasTranscurridas = (float)tiempoTranscurrido / 3600000.0;

        // Reducción del factor de escala directo de EmonLib para ESP32 (12
        bits / 3.3V)
        float factorConversion = (3.3 / 4096.0) / (5.0 / 1024.0);
        float v_calculado = emon.Vrms * factorConversion;
        float i_calculado = emon.Irms * factorConversion;

        // =====
        // FILTROS ANTI-RUIDO
        // =====
        if (i_calculado < 0.08 || i_calculado > 15.0) {
            i_calculado = 0.0;
        }
        if (v_calculado < 50.0 || v_calculado > 300.0) {
            v_calculado = 0.0;
        }

        // Calcular Potencia Activa aparente limpia
        float p_calculada = (i_calculado == 0.0) ? 0.0 : (v_calculado *
        i_calculado);

        // Calcular Energía Acumulada
        kWh += (p_calculada / 1000.0) * horasTranscurridas;
    }
}

```

```

// =====
// GUARDAR EN ESTRUCTURA (Bloque seguro)
// =====
noInterrupts(); // Pausa temporal para evitar que el I2C lea a medias
datos.v = v_calculado;
datos.i = i_calculado;
datos.p = p_calculada;
datos.e = kWh;
interrupts(); // Reactiva interrupciones inmediatamente

// =====
// MONITOR SERIAL
// =====
Serial.print("Voltaje: "); Serial.print(v_calculado, 1); Serial.print("
V | ");
    Serial.print("Corriente: "); Serial.print(i_calculado, 3);
Serial.print(" A | ");
    Serial.print("Potencia: "); Serial.print(p_calculada, 1);
Serial.print(" W | ");
    Serial.print("Energia: "); Serial.print(kWh, 5); Serial.println("
kWh");
    }
}

```

Anexo 2 – Código del módulo CubeCell AB02 en Arduino IDE

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "HT_SH1107Wire.h"
#include <Wire.h>

// =====
// CONFIGURACIÓN LORAWAN (DATOS DE TTN)
// =====
/* Llaves oficiales generadas en tu consola The Things Network */
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x07, 0x81, 0x43 };
uint8_t appEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
uint8_t appKey[] = { 0x11, 0x78, 0xA0, 0x42, 0xE3, 0x9C, 0x57, 0xA3, 0xED,
0x14, 0xAE, 0xCA, 0x9E, 0x55, 0xC3, 0x27 };

/* Parámetros de red y sub-banda FSB1 de US915 */
uint8_t nwkSKey[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
uint8_t appSKey[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
uint32_t devAddr = ( uint32_t )0x00000000;

uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;
DeviceClass_t loraWanClass = LORAWAN_CLASS;
uint32_t appTxDutyCycle = 15000;
bool overTheAirActivation = LORAWAN_NETMODE;
bool loraWanAdr = LORAWAN_ADR;
bool keepNet = LORAWAN_NET_RESERVE;
bool isTxConfirmed = LORAWAN_UPLINKMODE;
uint8_t appPort = 2;
uint8_t confirmedNbTrials = 4;

// =====
// OLED & ESTRUCTURA DE DATOS SENSOR I2C
// =====
```

```
SH1107Wire oled(0x3c, 500000, SDA, SCL, GEOMETRY_128_64, GPIO10);
```

```
struct DatosEnergia {  
    float v;  
    float i;  
    float p;  
    float e;  
} datos;
```

```
bool pulso = false;  
bool datosRecibidos = false;
```

```
// =====  
// PROCESAMIENTO DEL PAYLOAD LORAWAN (TESIS)  
// =====
```

```
static void prepareTxFrame( uint8_t port )  
{  
    int16_t voltios = (int16_t)(datos.v * 10);  
    int16_t amperios = (int16_t)(datos.i * 100);  
    int16_t potencia = (int16_t)(datos.p * 10);  
    int32_t energia = (int32_t)(datos.e * 1000);
```

```
    appDataSize = 10;
```

```
    appData[0] = (voltios >> 8) & 0xFF;  
    appData[1] = voltios & 0xFF;  
    appData[2] = (amperios >> 8) & 0xFF;  
    appData[3] = amperios & 0xFF;  
    appData[4] = (potencia >> 8) & 0xFF;  
    appData[5] = potencia & 0xFF;  
    appData[6] = (energia >> 24) & 0xFF;  
    appData[7] = (energia >> 16) & 0xFF;  
    appData[8] = (energia >> 8) & 0xFF;  
    appData[9] = energia & 0xFF;
```

```
}
```

```

void setup() {
    Serial.begin(115200);
    #if(AT_SUPPORT)
        enableAt();
    #endif

    pinMode(Vext, OUTPUT);
    digitalWrite(Vext, LOW);
    delay(100);

    oled.init();

    // Pantalla de Presentación UPSE
    oled.clear();
    oled.setFont(ArialMT_Plain_16);
    oled.drawString(42, 23, "UPSE");
    oled.drawRect(0, 0, 127, 63);
    oled.display();
    delay(2000);

    // Animación de barra de carga
    for (int i = 0; i <= 100; i += 4) {
        oled.clear();
        oled.setFont(ArialMT_Plain_10);
        oled.drawString(18, 18, "Iniciando sistema...");
        oled.drawRect(0, 0, 127, 63);
        oled.drawRect(12, 40, 104, 10);
        oled.fillRect(14, 42, i, 6);
        oled.display();
        delay(40);
    }
    delay(500);

    Wire.begin();

```

```

    deviceState = DEVICE_STATE_INIT;
    LoRaWAN.ifskipjoin();
}

void loop()
{
    // =====
    // LÓGICA DE ACTUALIZACIÓN CONTINUA (Mantiene viva la pantalla e I2C
    // siempre)
    // =====
    datosRecibidos = false;
    Wire.requestFrom(0x55, sizeof(datos));

    if (Wire.available() >= sizeof(datos)) {
        byte* p = (byte*)&datos;
        for (int i = 0; i < sizeof(datos); i++) {
            *(p + i) = Wire.read();
        }
        datosRecibidos = true;
        pulso = !pulso;
    }

    // Renderizado constante de la interfaz gráfica avanzada
    oled.clear();
    oled.drawRect(0, 0, 127, 63);
    oled.drawLine(64, 0, 64, 63);
    oled.drawLine(0, 31, 127, 31);
    oled.setFont(ArialMT_Plain_10);

    oled.drawString(4, 2, "VOLTAJE");
    oled.drawString(4, 15, String(datos.v, 1) + " V");

    oled.drawString(68, 2, "CORRIENTE");
    oled.drawString(68, 15, String(datos.i, 2) + " A");

```

```

oled.drawString(4, 34, "POTENCIA");
oled.drawString(4, 47, String(datos.p, 1) + " W");

oled.drawString(68, 34, "ENERGIA");
oled.drawString(68, 47, String(datos.e, 3) + " kWh");

// Indicador de actividad o estado de red
if (!datosRecibidos) {
    oled.fillRect(49, 23, 30, 15);
    oled.setColor(BLACK);
    oled.drawString(53, 25, "ERR");
    oled.setColor(WHITE);
} else {
    if (pulso) oled.fillRect(62, 29, 5, 5);
}
oled.display();
delay(100);

// =====
// MÁQUINA DE ESTADOS DE LORAWAN (Se ejecuta en segundo plano)
// =====
switch( deviceState )
{
    case DEVICE_STATE_INIT:
    {
#ifdef LORAWAN_DEVEUI_AUTO
        LoRaWAN.generateDeveuiByChipID();
#endif
#ifdef AT_SUPPORT
        getDevParam();
#endif
        printDevParam();
        LoRaWAN.init(loraWanClass,loraWanRegion);
        deviceState = DEVICE_STATE_JOIN;

```

```

        break;
    }
    case DEVICE_STATE_JOIN:
    {
        LoRaWAN.join();
        break;
    }
    case DEVICE_STATE_SEND:
    {
        prepareTxFrame( appPort );
        LoRaWAN.send();
        deviceState = DEVICE_STATE_CYCLE;
        break;
    }
    case DEVICE_STATE_CYCLE:
    {
        txDutyCycleTime = appTxDutyCycle + randr( 0, APP_TX_DUTYCYCLE_RND );
        LoRaWAN.cycle(txDutyCycleTime);
        deviceState = DEVICE_STATE_SLEEP;
        break;
    }
    case DEVICE_STATE_SLEEP:
    {
        LoRaWAN.sleep();
        break;
    }
    default:
    {
        deviceState = DEVICE_STATE_INIT;
        break;
    }
}
}
}

```

Anexo 3 – JSON compact de Node-RED (flujo)

```
[{"id":"419a7c60726eec74","type":"tab","label":"Flow
1","disabled":false,"info":"","env":[],{"id":"ef85cc104688fa04","type":"mqtt
in","z":"419a7c60726eec74","name":"","topic":"v3/tesis-energia@ttn/devices/nodo-
upse/up","qos":"2","datatype":"json","broker":"111e714bbe771a4f","nl":false,"rap":true,"rh":
0,"inputs":0,"x":230,"y":60,"wires":[["faf937d342f76ef2"]]},{"id":"faf937d342f76ef2","type
":"json","z":"419a7c60726eec74","name":"","property":"payload","action":"obj","pretty":fals
e,"x":150,"y":140,"wires":[["7741ce28d19f1acd"]]},{"id":"ff2995383ce9edef","type":"debug
","z":"419a7c60726eec74","name":"debug
1","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","target
Type":"msg","statusVal":"","statusType":"auto","x":600,"y":60,"wires":[]},{"id":"7741ce28d
19f1acd","type":"change","z":"419a7c60726eec74","name":"","rules":[{"t":"set","p":"payloa
d","pt":"msg","to":"payload.uplink_message.decoded_payload","tot":"msg"}],"action":"","pr
operty":"","from":"","to":"","reg":false,"x":360,"y":140,"wires":[["ff2995383ce9edef","be017
6a3496d48b0","eb2c3663beedc77d","9a2b24b1d6708b36","99e0288035345bbf","d686ea5b9
77481e8","fad68565e60f2a2b","dc3174ef41c36477","b6f78507c79478ef"]]},{"id":"be0176a
3496d48b0","type":"ui-
gauge","z":"419a7c60726eec74","name":"","group":"c73dabd874aeb8e","order":1,"value":"
payload.voltaje","valueType":"msg","width":"4","height":"3","gtype":"gauge-
```

```

half","gstyle":"needle","title":"VOLTAJE
LOCAL","alwaysShowTitle":false,"floatingTitlePosition":"top-
left","units":"V","icon":"","prefix":"","suffix":"","segments":[{"from":"0","color":"#5cd65c",
"text":"","textType":"label"},{"from":"108","color":"#ffc800","text":"","textType":"label"},{
"from":"127","color":"#ea5353","text":"","textType":"label"}],"min":0,"max":"150","sizeThi
ckness":16,"sizeGap":4,"sizeKeyThickness":"8","styleRounded":true,"styleGlow":false,"clas
sName":"","x":930,"y":80,"wires":[]],{"id":"eb2c3663beedc77d","type":"ui-
gauge","z":"419a7c60726eec74","name":"","group":"c73dabd874aeb8e","order":2,"value":"
payload.corriente","valueType":"msg","width":"4","height":"3","gtype":"gauge-
half","gstyle":"needle","title":"CORRIENTE","alwaysShowTitle":false,"floatingTitlePosition
":"top-
left","units":"A","icon":"","prefix":"","suffix":"","segments":[{"from":"0","color":"#5cd65c",
"text":"","textType":"label"},{"from":"12","color":"#ffc800","text":"","textType":"label"},{"f
rom":"15","color":"#ea5353","text":"","textType":"label"}],"min":0,"max":"15","sizeThickne
ss":16,"sizeGap":4,"sizeKeyThickness":"8","styleRounded":true,"styleGlow":false,"classNa
me":"","x":910,"y":120,"wires":[]],{"id":"9a2b24b1d6708b36","type":"ui-
gauge","z":"419a7c60726eec74","name":"","group":"c73dabd874aeb8e","order":3,"value":"
payload.potencia","valueType":"msg","width":"4","height":"3","gtype":"gauge-
half","gstyle":"needle","title":"POTENCIA
ACTIVA","alwaysShowTitle":false,"floatingTitlePosition":"top-
left","units":"W","icon":"","prefix":"","suffix":"","segments":[{"from":"0","color":"#5cd65c"

```

```
t":"set","p":"topic","pt":"msg","to":"Corriente
(A)","tot":"str"}], "action":"","property":"","from":"","to":"","reg":false,"x":910,"y":280,"wire
s":[["412a2bfe11391c27"]]],{"id":"fad68565e60f2a2b","type":"change","z":"419a7c60726ee
c74","name":"Potencia
Grafica","rules":[{"t":"set","p":"payload","pt":"msg","to":"payload.potencia","tot":"msg"}, {"
t":"set","p":"topic","pt":"msg","to":"Potencia
(W)","tot":"str"}], "action":"","property":"","from":"","to":"","reg":false,"x":910,"y":320,"wir
es":[["412a2bfe11391c27"]]],{"id":"dc3174ef41c36477","type":"change","z":"419a7c60726e
ec74","name":"Energia
Grafica","rules":[{"t":"set","p":"payload","pt":"msg","to":"payload.energia","tot":"msg"}, {"t
":"set","p":"topic","pt":"msg","to":"Energia
(kWh)","tot":"str"}], "action":"","property":"","from":"","to":"","reg":false,"x":900,"y":360,"
wires":[["412a2bfe11391c27"]]],{"id":"b6f78507c79478ef","type":"switch","z":"419a7c6072
6eec74","name":"Filtros Anomalias
V","property":"payload.voltaje","propertyType":"msg","rules":[{"t":"gt","v":"127","vt":"num
"}, {"t":"lt","v":"105","vt":"num"}], "checkall":"true","repair":false,"outputs":2,"x":170,"y":26
0,"wires":[["cfd9ba9239f372ff","ddb4c77330f2629d"],["20c4dde3bae4366e","ddb4c77330f2
629d"]]],{"id":"cfd9ba9239f372ff","type":"template","z":"419a7c60726eec74","name":"Red
actar Alerta
Alta","field":"payload","fieldType":"msg","format":"handlebars","syntax":"mustache","templ
ate":"{\n  \chatId\": 7132095127,\n  \type\": \message\",\n  \content\": \" ⚠ ALERTA
```

```

text":"","textType":"label"},{"from":"1440","color":"#ffc800","text":"","textType":"label"},{
"from":"1800","color":"#ea5353","text":"","textType":"label"}],{"min":0,"max":"2500","sizeT
hickness":16,"sizeGap":4,"sizeKeyThickness":"8","styleRounded":true,"styleGlow":false,"cl
assName":"","x":930,"y":160,"wires":[]},{"id":"412a2bfe11391c27","type":"ui-
chart","z":"419a7c60726eec74","group":"c73dabd874aeb8e","name":"","label":"GRAFICA
","order":4,"chartType":"line","category":"topic","categoryType":"msg","xAxisLabel":"Tiem
po","xAxisProperty":"","xAxisPropertyType":"timestamp","xAxisType":"time","xAxisForma
t":"","xAxisFormatType":"auto","xmin":"","xmax":"","yAxisLabel":"","yAxisProperty":"payl
oad","yAxisPropertyType":"msg","ymin":"","ymax":"","bins":10,"action":"append","stackSer
ies":false,"pointShape":"circle","pointRadius":4,"showLegend":true,"removeOlder":"24","re
moveOlderUnit":"3600","removeOlderPoints":"","colors":["#0095ff","#ff0000","#ff7f0e","#2
ca02c","#a347e1","#d62728","#ff9896","#9467bd","#c5b0d5"],"textColor":["#252222"],"text
ColorDefault":false,"gridColor":["#cfcfcf"],"gridColorDefault":false,"width":"12","height":"
6","className":"","interpolation":"linear","x":1160,"y":300,"wires":[]},{"id":"99e0288035
345bbf","type":"change","z":"419a7c60726eec74","name":"Voltaje
Grafica","rules":[{"t":"set","p":"payload","pt":"msg","to":"payload.voltaje","tot":"msg"},{"t
":"set","p":"topic","pt":"msg","to":"Voltaje
(V)","tot":"str"}],{"action":"","property":"","from":"","to":"","reg":false,"x":900,"y":240,"wire
s":["412a2bfe11391c27"]},{"id":"d686ea5b977481e8","type":"change","z":"419a7c60726e
ec74","name":"Corriente
Grafica","rules":[{"t":"set","p":"payload","pt":"msg","to":"payload.corriente","tot":"msg"},{"

```

```

DE ANOMALÍA - UPSE ⚠️ \n\nSe ha detectado una SOBRETENSIÓN crítica en la
red.\n\nVoltaje medido: {{payload.voltaje}} V\nCorriente actual: {{payload.corriente}}
A\n\n", "output": "json", "x": 740, "y": 420, "wires": [[{"d9c036131bdb541"}]], {"id": "20c4dde
3bae4366e", "type": "template", "z": "419a7c60726eec74", "name": "Redactar Alerta
Baja", "field": "payload", "fieldType": "msg", "format": "handlebars", "syntax": "mustache", "temp
late": "{\n  \"chatId\": 7132095127,\n  \"type\": \"message\",\n  \"content\": \" ⚠️ ALERTA
DE ANOMALÍA - UPSE ⚠️ \n\nSe ha detectado una SUBTENSIÓN (Caída de
Voltaje).\n\nVoltaje medido: {{payload.voltaje}} V\nCorriente actual:
{{payload.corriente}}
A\n\n", "output": "json", "x": 740, "y": 460, "wires": [[{"d9c036131bdb541"}]], {"id": "d9c0361
31bdb541", "type": "telegram
sender", "z": "419a7c60726eec74", "name": "", "bot": "f3c8d4d680163d68", "haserroroutput": fals
e, "outputs": 1, "x": 970, "y": 440, "wires": [[]], {"id": "ddb4c77330f2629d", "type": "function", "z":
"419a7c60726eec74", "name": "Formatear Fila Tabla", "func": "// Obtener historial\nlet historial
= flow.get(\"historial\") || [];\n\n// Fecha y hora\nlet ahora = new Date();\nlet fecha =
ahora.toLocaleDateString(\"es-EC\");\nlet hora = ahora.toLocaleTimeString(\"es-EC\");\n\n//
Obtener voltaje\nlet voltaje = Number(msg.payload.voltaje);\n\nif (isNaN(voltaje)) return
null;\n\n// Variables\nlet evento = \"\";\nlet estado = \"\";\nlet color = \"\";\n\n//
Clasificación\nif (voltaje < 105) {\n\n  evento = \"Bajo Voltaje\";\n  estado = \" 🔴
Crítico\";\n  color = \"#E53935\";\n\n}\nelse if (voltaje > 127) {\n\n  evento =

```

```

Historial","func":"if (msg.payload === \"clear\") {\n\n  flow.set(\"historial\", []);\n\n  msg.payload = [];\n\n  return msg;\n\n}\n\nreturn

null;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":300,"y":520,"
wires":[["19c9d75547ad90d8"]]],{"id":"111e714bbe771a4f","type":"mqtt-
broker","name":"TTN - Monitoreo

UPSE","broker":"nam1.cloud.thethings.network","port":1883,"clientId":"","autoConnect":tru
e,"usetls":false,"protocolVersion":4,"keepalive":60,"cleansession":true,"autoUnsubscribe":tru
e,"birthTopic":"","birthQos":0,"birthRetain":false,"birthPayload":"","birthMsg":{"close
Topic":"","closeQos":0,"closeRetain":false,"closePayload":"","closeMsg":{"willTopic":
"",willQos":0,"willRetain":false,"willPayload":"","willMsg":{"userProps":"","session
Expiry":""}},{"id":"c73dabd874aeb8e","type":"ui-group","name":"Variables en Tiempo Real
","page":"4ab7f6444a91174f","width":16,"height":8,"order":1,"showTitle":true,"classNa
me":"","visible":true,"disabled":false,"groupType":"default"},"id":"f3c8d4d680163d68",
"type":"telegram bot","botname":"Bot Tesis

UPSE","usernames":"","chatids":"","baseapiurl":"","testenvironment":false,"updatemode":"p
olling","pollinterval":300,"usesocks":false,"sockshost":"","socksprotocol":"socks5","sockspo
rt":6667,"socksusername":"anonymous","sockspassword":"","bothost":"","botpath":"","localb
othost":"0.0.0.0","localbotport":8443,"publicbotport":8443,"privatekey":"","certificate":"","u
seselfsignedcertificate":false,"sslterminated":false,"verboselogging":false},"id":"4ab7f6444a
91174f","type":"ui-page","name":"Sistema de Monitoreo Energetico

IoT","ui":"e01a6b10d11738ec","path":"/page1","icon":"home","layout":"grid","theme":"a33e

```

Anexo 4 – Payload formatters para CubeCell AB02 en TTN

```
function decodeUplink(input) {  
  
  var data = {};  
  
  // 1. Reconstruir Voltaje (Bytes 0 y 1)  
  var voltiosRaw = (input.bytes[0] << 8) | input.bytes[1];  
  data.voltaje = voltiosRaw / 10.0;  
  
  // 2. Reconstruir Corriente (Bytes 2 y 3)  
  var amperiosRaw = (input.bytes[2] << 8) | input.bytes[3];  
  data.corriente = amperiosRaw / 100.0;  
  
  // 3. Reconstruir Potencia (Bytes 4 y 5)  
  var potenciaRaw = (input.bytes[4] << 8) | input.bytes[5];  
  data.potencia = potenciaRaw / 10.0;  
  
  // 4. Reconstruir Energía (Bytes 6, 7, 8 y 9)  
  var energiaRaw = (input.bytes[6] << 24) | (input.bytes[7] << 16) | (input.bytes[8] << 8) |  
  input.bytes[9];  
  data.energia = energiaRaw / 1000.0;  
  
  return {  
    data: data,  
    warnings: [],  
    errors: []  
  };  
}
```

```
return {  
  data: data,  
  warnings: [],  
  errors: []  
};  
}
```