

CARRERA DE ELECTRÓNICA

DESARROLLO DE UNA PLATAFORMA WEB PARA GESTIONAR DATOS DE CONSUMO ELÉCTRICO PARA HOGARES EN LA ZONA URBANA.

Trabajo de Integración Curricular previo a la obtención del Título de Tecnólogo Superior en Electrónica

Autores: AMAGUA TITUAÑA FAUSTO DARIO
GAVILANES JIJÓN BYRON FERNANDO

Tutor: PHD. ZAMBRANO VIZUETE OSCAR MARCELO

**FEBRERO, 2025
SANGOLQUI – ECUADOR**

Autor: AMAGUA TITUAÑA FAUSTO DARIO	Título a obtener: Tecnólogo Superior en Electrónica Matriz: Sangolquí -Ecuador Correo electrónico: fausto.amagua@ister.edu.ec
	Título a obtener: Tecnólogo Superior en Electrónica Matriz: Sangolquí -Ecuador Correo electrónico: byron.gavilanes@ister.edu.ec
Autor: GAVILANES JIJÓN BYRON FERNANDO	Título a obtener: Tecnólogo Superior en Electrónica Matriz: Sangolquí -Ecuador Correo electrónico: byron.gavilanes@ister.edu.ec
	Dirigido por: PHD. ZAMBRANO VIZUETE OSCAR MARCELO Matriz: Sangolquí -Ecuador Correo electrónico: marcelo.zambrano@ister.edu.ec
	

TODOS LOS DERECHOS RESERVADOS.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2024 Tecnológico Universitario Rumiñahui

SANGOLQUÍ – ECUADOR

AMAGUA TITUAÑA FAUSTO DARIO

GAVILANES JIJÓN BYRON FERNANDO

**DESARROLLO DE UNA PLATAFORMA WEB PARA GESTIONAR DATOS DE
CONSUMO ELÉCTRICO PARA HOGARES EN LA ZONA URBANA.**

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.2

Sangolquí, 24 de marzo del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, **Byron Fernando Gavilanes Jijón** declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado **DESARROLLO DE UNA PLATAFORMA WEB PARA GESTIONAR DATOS DE CONSUMO ELÉCTRICO PARA HOGARES EN LA ZONA URBANA**, de la Tecnología Superior en Electrónica; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



GAVILANES JIJÓN BYRON FERNANDO

C.I.: 1714562970

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.2

Sangolquí, 24 de marzo del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO

Presente

Por medio de la presente, yo, **Fausto Darío Amagua Tituaña** declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado **Desarrollo de una plataforma web para gestionar datos de consumo eléctrico para hogares en la zona urbana**, de la **Tecnología Superior en Electrónica**; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



Fausto Darío Amagua Tituaña

C.I.: 1724183635

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA:

TECNOLOGÍA SUPERIOR EN ELECTRÓNICA

AUTORES:

AMAGUA TITUAÑA FAUSTO DARIO

GAVILANES JIJÓN BYRON FERNANDO

TUTOR (METODOLÓGICO Y ACADÉMICO):

RICARDO ELICIO ROSERO YUGSI

ZAMBRANO VIZUETE OSCAR MARCELO

CONTACTO ESTUDIANTE: DARIO AMAGUA

0968551765

CORREO ELECTRÓNICO:

darioamagua8@gmail.com

CONTACTO ESTUDIANTE: BYRON GAVILANES

0962302420

CORREO ELECTRÓNICO:

feru1083@gmail.com

TEMA:

DESARROLLO DE UNA PLATAFORMA WEB PARA GESTIONAR DATOS DE
CONSUMO ELÉCTRICO PARA HOGARES EN LA ZONA URBANA.

OPCIÓN DE TITULACIÓN:

UNIDAD DE INTEGRACIÓN CURRICULAR

RESUMEN EN ESPAÑOL:

La investigación tuvo como objetivo desarrollar una plataforma web para la gestión y monitoreo del consumo eléctrico en hogares urbanos. Para ello, se implementó un sistema distribuido de recolección de datos en tiempo real, integrando sensores capaces de medir voltaje, corriente, potencia y energía consumida. Estos dispositivos transmiten la información a través de un ESP32, permitiendo su visualización local en una pantalla y su monitoreo remoto mediante Blynk. Además, se desarrollaron servicios web alojados en un Servidor Privado Virtual (VPS) en Google Cloud, donde la transmisión de datos se gestiona mediante un Broker MQTT y un servidor Node-RED. Los datos recopilados se almacenan en una base de datos MySQL y se visualizan en Grafana, lo que permite a los usuarios supervisar su consumo eléctrico y estimar costos según los kWh utilizados.

Para validar la precisión del sistema, se aplicó una metodología descriptiva y exploratoria basada en observación, encuestas y pruebas experimentales. Los resultados demuestran que la plataforma optimiza la gestión energética y tiene potencial escalable. Sin embargo, enfrenta desafíos como el costo de implementación y la necesidad de educar a los usuarios en la interpretación de datos. En conclusión, esta solución innovadora no solo mejora la eficiencia energética de cada hogar, sino que también puede integrarse en iniciativas de ciudades inteligentes, promoviendo el uso responsable de la electricidad y fomentando la sostenibilidad urbana

PALABRAS CLAVE:

Variables eléctricas
Plataforma web
IoT
MQTT
Grafana.

ABSTRACT:

The research aimed to develop a web platform for the management and monitoring of electrical consumption in urban households. To achieve this, a distributed real-time data collection system was implemented, integrating sensors capable of measuring voltage, current, power, and consumed energy. These devices transmit information via an ESP32 microcontroller, enabling local visualization on a screen and remote monitoring through Blynk. Additionally, web services hosted on a Virtual Private Server (VPS) in Google Cloud were developed, where data transmission is managed via an MQTT Broker and a Node-RED server. The collected data is stored in a MySQL database and visualized in Grafana, allowing users to monitor their electricity consumption and estimate costs based on kWh usage.

To validate the system's accuracy, a descriptive and exploratory methodology was applied, relying on observation, surveys, and experimental testing. The results demonstrate that the platform optimizes energy management and has scalable potential. However, it faces challenges such as implementation costs and the need to educate users in data interpretation. In conclusion, this innovative solution not only improves energy efficiency in individual households but could also integrate into smart city initiatives, promoting responsible electricity use and fostering urban sustainability.

KEYWORDS:

Electrical variables
Web Platform
IoT
MQTT
Grafana.



Firma del Estudiante (AUTOR)
AMAGUA TITUAÑA FAUSTO DARIO
C.I.: 1724183635



Firma del Estudiante (AUTOR)
GAVILANES JIJÓN BYRON FERNANDO
C.I.: 1714562970

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 24 de marzo del 2025

Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo, BYRON FERNANDO GAVILANES JIJÓN, con C.I.: 1714562970 alumno de la Carrera de ELECTRÓNICA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,



Firma del Estudiante

Byron Fernando Gavilanes Jijón

C.I.: 1714562970

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 24 de marzo del 2025

Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo, FAUSTO DARIO AMAGUA TITUAÑA, con C.I.: 1724183635 alumno de la Carrera ELECTRONICA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,



Firma del Estudiante

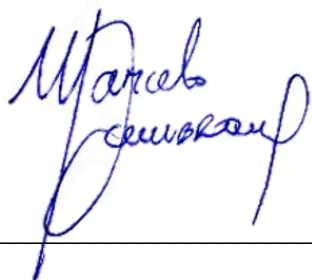
Fausto Dario Amagua Tituaña

C.I.: 1724183635

APROBACIÓN DEL TUTOR/A

En mi calidad de Tutor/a del Trabajo de Titulación, presentado por **AMAGUA TITUAÑA FAUSTO DARIO** con CC: 1724183635 y **GAVILANES JIJÓN BYRON FERNANDO** con CC: 1714562970 para optar por el Título de Tecnólogo/a Superior en Electrónica; cuyo título es: **DESARROLLO DE UNA PLATAFORMA WEB PARA GESTIONAR DATOS DE CONSUMO ELÉCTRICO PARA HOGARES EN LA ZONA URBANA**, se considera que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que fuere designado.

Sangolquí, a los 21 días del mes de marzo de 2025



PHD. ZAMBRANO VIZUETE OSCAR MARCELO

DOCENTE-TUTOR

C.I.

DEDICATORIA

A mis padres, por creer en mí incluso en mis momentos más difíciles, por nunca dudar de lo que soy capaz y por su apoyo incondicional, aun cuando eso significó sacrificios para ustedes. Gracias por ser mi ejemplo y mi mayor motivación.

A mi esposa y a mis hijos, por su paciencia, por entender las horas que tuve que dedicar a este camino y por estar siempre ahí, animándome a seguir. Este logro también es de ustedes.

Fernando Gavilanes.

DEDICATORIA

Con todo mi cariño y gratitud, dedico este trabajo a mi hermano Fernando Amagua, quien con su generosidad y confianza en mí hizo posible que este sueño se convirtiera en realidad. Su apoyo incondicional, esfuerzo y sacrificio han sido una inspiración para mí, y por ello, siempre le estaré profundamente agradecido.

A mi familia, por ser mi refugio, mi fortaleza y mi mayor inspiración. Por cada palabra de aliento, por cada sacrificio silencioso y por acompañarme en cada paso de este camino.

A mis profesores, por compartir su conocimiento y enseñarme a superar los retos con determinación. A mis compañeros, por las experiencias y aprendizajes compartidos que hicieron de este viaje una etapa inolvidable.

Este logro es un reflejo de amor, esfuerzo y del apoyo inquebrantable de quienes han creído en mí.

Fausto Dario Amagua Tituaña.

AGRADECIMIENTO

Quiero agradecer, antes que nada, a mis padres, por su apoyo incondicional y por nunca dudar de mí, incluso en los momentos más difíciles. Su esfuerzo y sacrificio han sido clave para que hoy pueda alcanzar esta meta.

A mi esposa y a mis hijos, por su paciencia y comprensión. Sé que muchas veces tuve que restar tiempo en familia para dedicarlo a este camino, y, aun así, siempre me brindaron su apoyo y motivación.

A mis docentes, mis compañeros y al personal del ISTER, por compartir sus conocimientos, por su orientación y por cada enseñanza que me ayudó a crecer tanto académica como personalmente. Su dedicación ha sido fundamental en este proceso.

A todos los que de alguna manera me apoyaron en este camino, ¡gracias de corazón!

Fernando Gavilanes.

AGRADECIMIENTO

Antes que nada, agradezco a Dios por darme la fortaleza, la perseverancia y la oportunidad de llegar hasta aquí.

A mi hermano Fernando Amagua, con todo mi corazón, por ser mi mayor apoyo en este camino. Su esfuerzo, generosidad y confianza en mí han sido el pilar que me permitió alcanzar esta meta.

A mi familia, por su amor incondicional, por su paciencia en los momentos difíciles y por ser mi refugio en cada desafío. A mis padres, por enseñarme el valor del esfuerzo y la dedicación.

A mis docentes del Instituto Superior Tecnológico Rumiñahui, por compartir su conocimiento y guiarme en este proceso de formación.

A mis amigos y compañeros, por su compañía, por cada risa compartida en medio del esfuerzo y por demostrarme que el aprendizaje se construye en equipo.

Finalmente, a todas las personas que, de una u otra manera, contribuyeron a la realización de este proyecto, mi más sincero agradecimiento.

Fausto Dario Amagua Tituaña

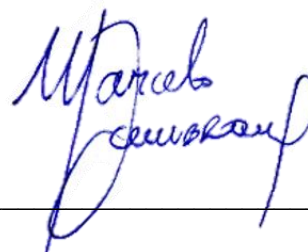
RESUMEN EJECUTIVO

La investigación tuvo como objetivo desarrollar una plataforma web para la gestión y monitoreo del consumo eléctrico en hogares urbanos. Para ello, se implementó un sistema distribuido de recolección de datos en tiempo real, integrando sensores capaces de medir voltaje, corriente, potencia y energía consumida. Estos dispositivos transmiten la información a través de un ESP32, permitiendo su visualización local en una pantalla y su monitoreo remoto mediante Blynk. Además, se desarrollaron servicios web alojados en un Servidor Privado Virtual (VPS) en Google Cloud, donde la transmisión de datos se gestiona mediante un Broker MQTT y un servidor Node-RED. Los datos recopilados se almacenan en una base de datos MySQL y se visualizan en Grafana, lo que permite a los usuarios supervisar su consumo eléctrico y estimar costos según los kWh utilizados.

Para validar la precisión del sistema, se aplicó una metodología descriptiva y exploratoria basada en observación, encuestas y pruebas experimentales. Los resultados demuestran que la plataforma optimiza la gestión energética y tiene potencial escalable. Sin embargo, enfrenta desafíos como el costo de implementación y la necesidad de educar a los usuarios en la interpretación de datos. En conclusión, esta solución innovadora no solo mejora la eficiencia energética de cada hogar, sino que también puede integrarse en iniciativas de ciudades inteligentes, promoviendo el uso responsable de la electricidad y fomentando la sostenibilidad urbana.

PALABRAS CLAVE

Variables eléctricas, plataforma web, IoT, MQTT, Grafana.



PHD. ZAMBRANO VIZUETE OSCAR
MARCELO

TUTOR DEL TRABAJO DE TITULACIÓN

ABSTRACT

The research aimed to develop a web platform for the management and monitoring of electricity consumption in urban homes. For this purpose, a distributed real-time data collection system was implemented, integrating sensors capable of measuring voltage, current, power and energy consumption. These devices transmit the information through an ESP32, allowing its local visualization on a screen and its remote monitoring through Blynk. In addition, web services were developed and hosted in a Virtual Private Server (VPS) in Google Cloud, where data transmission is managed through an MQTT Broker and a Node-RED server. The collected data is stored in a MySQL database and visualized in Grafana, allowing users to monitor their electricity consumption and estimate costs according to kWh used.

To validate the accuracy of the system, a descriptive and exploratory methodology based on observation, surveys and experimental tests was applied. The results show that the platform optimizes energy management and has scalable potential. However, it faces challenges such as the cost of implementation and the need to educate users in data interpretation. In conclusion, this innovative solution not only improves the energy efficiency of each household, but can also be integrated into smart city initiatives, promoting the responsible use of electricity and fostering urban sustainability.

KEYWORDS

Electrical variables, web platform, IoT, MQTT, Grafana

ÍNDICE GENERAL

TODOS LOS DERECHOS RESERVADOS.....	iii
APROBACIÓN DEL TUTOR/A	x
DEDICATORIA	xi
DEDICATORIA	xi
AGRADECIMIENTO.....	xii
AGRADECIMIENTO.....	xiii
RESUMEN EJECUTIVO	xiv
PALABRAS CLAVE.....	xiv
ABSTRACT	xv
KEYWORDS.....	xv
1. INTRODUCCIÓN.....	1
1. CAPÍTULO I.....	3
1.1. Planteamiento del problema.....	3
1.2. Preguntas de directrices	4
1.3. Objetivo General	4
1.4. Objetivos Especifico	4
1.5. Justificación.....	5
1.6. Alcance	6
2. CAPÍTULO II	7
2.1. Fundamento teórico relacionado al trabajo de integración curricular	7
2.1.1. Energía eléctrica	7
2.1.2. Voltaje	7
2.1.3. Corriente	8
2.1.4. Potencia Eléctrica.....	8
2.1.5. Cálculo de consumo eléctrico.....	8
2.2. Internet de las cosas IoT	9
2.2.1. Arquitectura IoT	9
2.2.2. Aplicaciones y Servicios de IoT	10
2.2.3. Plataformas IOT.....	11
2.3. Microcontroladores.....	16
2.3.1. Microcontrolador ESP 32.....	17
2.4. Componentes de medición y visualización	18
2.4.1. Modulo sensor Pzem-004T.....	18
2.4.2. Pantalla OLED.....	19

2.5.	Página web de IoT	20
2.5.1.	Frontend	21
2.5.1.1.	Dashboards de Visualización.....	22
A.	React	22
C.	Grafana.....	23
2.5.2.	Backend.....	24
2.5.2.1.	Servicios Backend para IoT.....	24
3.	CAPÍTULO III.....	29
3.1.	El método científico.....	29
3.1.1.	Métodos teóricos de investigación.....	29
3.1.1.1.	Fase de Diseño	29
3.1.1.2.	Fase de Implementación.....	30
3.1.1.3.	Fase de Evaluación.....	30
3.1.2.	Técnicas e Instrumentos de recolección de datos	32
4.	CAPÍTULO IV.....	33
4.1.	Diseño de la propuesta	33
4.1.1.	Estudio de viabilidad para el desarrollo de un sistema de monitoreo energético. 33	
4.1.2.	Diseño del esquema para el prototipo de monitoreo.....	36
4.1.3.	Diseño conceptual de la plataforma web.....	38
4.1.4.	Diseño del circuito electrónico y PCB.....	40
4.1.5.	Diseño 3D de la carcasa protectora del prototipo.	43
4.2.	Implementación	49
4.2.1.	Etapa1: Impresión 3D de la carcasa protectora del prototipo.....	49
4.2.2.	Etapa2: Fabricación de la PCB y montaje de los elementos.	50
4.2.3.	Etapa3: Desarrollo del código para ESP32.	54
4.2.4.	Etapa4: Desarrollo e implementación de Plataforma web IoT.....	65
4.2.5.	Etapa5: Integración del Prototipo de medición con la Pagina Web	112
4.3.	Pruebas.....	118
4.3.1.	Etapa1: Verificación de acceso correcto a la plataforma.	118
4.3.2.	Etapa2: Inducción sobre el manejo de la plataforma	122
4.3.3.	Etapa3: Retroalimentación por parte de los usuarios (entorno de la plataforma satisfacción con la información obtenida).	127
4.3.4.	Etapa4: Evaluación de Datos visualizados en los dashboards de la Plataforma Web IoT. 130	
5.	CAPITULO V	142

6.	CONCLUSIONES Y RECOMENDACIONES.....	142
5.1	Conclusiones	142
5.2	Recomendaciones	143
7.	Referencias.....	144
8.	ANEXOS	146

Índice de Tablas

Tabla 1 Datos técnicos sensor Pzem-004T.....	19
Tabla 2 Datos técnicos pantalla oled 1.3 pulgadas.	20
Tabla 3 Técnicas e Instrumentos de recolección de datos.....	32

Índice de Figuras

Figura 1 Internet de las Cosas.....	9
Figura 2 Capas de la arquitectura IoT.	10
Figura 3 Blynk Console y Apps.....	12
Figura 4 IoT Dashboards Kaa.....	13
Figura 5 Dashboards Mainflux.	14
Figura 6 Data Visualización.	15
Figura 7 Monitoreo de energía inteligente.	16
Figura 8 Microcontroladores.	17
Figura 9 Distribución de pines ESP 32 WROOM.	18
Figura 10 Sensor de PZEM-004T V3.	18
Figura 11 Pantalla OLED I2C IIC de 1.3 pulgadas.	20
Figura 12 Desarrollo de plataformas Internet Of Things.	21
Figura 13 Frontend vs. Backend.	22
Figura 14 Analítica y monitorización con Grafana.....	23
Figura 15 Servidores en la Nube.....	25
Figura 16 Eclipse Mosquitto MQTT Broker y Cliente.	26
Figura 17 Bases de datos.	28
Figura 18 Pregunta # 1	33
Figura 19 Pregunta # 2	34
Figura 20 Pregunta # 3	34
Figura 21 Pregunta # 4	35
Figura 22 Pregunta # 5	35
Figura 23 Diseño de conexión de componentes.	36
Figura 24 Diseño conceptual de la plataforma Web.	37
Figura 25 Caratula de presentación.....	38
Figura 26 Ventana de ingreso a la plataforma web.....	39
Figura 27 Ventana visualización de los datos de medición.....	39
Figura 28 Selección de librerías de componentes.....	40
Figura 29 Diseño del esquemático.....	41
Figura 30 Pistas de conexión.....	42
Figura 31 Ruteo de PCB terminada.	42
Figura 32 Vista 3D.....	43
Figura 33 Plano de Caja Software Fusion 360.	44
Figura 34 Extrusión de la base.....	45

Figura 35 Bocetos Superficiales Puerto AC y Tipo C.	45
Figura 36 Sujetadores de Fijación.....	46
Figura 37 Corte de Sujetadores de fijación.	47
Figura 38 Vaciado de Tapa.....	48
Figura 39 Diseño Terminado.....	48
Figura 40 Impresión de Caja 3D en material ABS.	49
Figura 41 Remoción y Limpieza de Soportes.....	50
Figura 42 Fabricación de PCB.....	51
Figura 43 Limpieza de PCB.	52
Figura 44 Elaboración de Cable UART.	52
Figura 45 Conexiones AC-DC de alimentación.	53
Figura 46 Incorporación de componentes electrónicos y carcasa protectora.	54
Figura 47 Diagrama de Flujo código ESP32 a Blynk.	55
Figura 48 Configuración de Blynk y Librerías Arduino IDE.....	56
Figura 49 Configuración de Constantes Oled, Sensor y Wifi.	57
Figura 50 Lectura de datos y Envío de Información.....	58
Figura 51 Diagrama de Flujo ESP32 y Pagina IoT implementada.	60
Figura 52 Librerías Arduino IDE.....	61
Figura 53 Configuración de Cliente MQTT.....	62
Figura 54 Configuración Inicial de Void Setup.....	63
Figura 55 Configuración Inicial de Void Loop.	64
Figura 56 Configuración de Conexión WiFi y Comunicación MQTT.....	65
Figura 57 Creación de Cuenta IoT Blynk.....	66
Figura 58 Configuración de Pines Virtuales.....	67
Figura 59 Web Dashboard configurada	67
Figura 60 Inicio de configuración GOOGLE CLOUD.....	68
Figura 61 Creación de nuevo proyecto.	68
Figura 62 Definir valor para pagar mensualmente.	69
Figura 63 Crear un VPS.	69
Figura 64 Habilitar la API.	69
Figura 65 Ingresar a Marketplace.	70
Figura 66 Elección sistema operativo.	70
Figura 67 Elegir CPU.....	71
Figura 68 Elegir Disco.	71
Figura 69 Firewall.....	71

Figura 70 Configurar IP estática.	72
Figura 71 Ubicar IP externa.....	72
Figura 72 Dar nombre.	73
Figura 73 Generación de Keys en PUTTY.....	74
Figura 74 Copiar key.....	74
Figura 75 Agregar nueva clave en el servidor.	75
Figura 76 Iniciar sesión en MobaXterm.....	75
Figura 77 Instalar Ubuntu y Apache.	76
Figura 78 Confirmar la correcta instalación.	76
Figura 79 Crear cuenta en Domain.	77
Figura 80 Elegir el nombre del dominio.	78
Figura 81 Apuntar el DNS a la IP externa.....	78
Figura 82 Crear Zona DNS.....	79
Figura 83 Vincular VPS con el dominio.	79
Figura 84 Colocar nombres de los dominios del VPS en Domain.	80
Figura 85 Verificar Dominio.	80
Figura 86 Crear política de Firewall.	81
Figura 87 Crear Nueva Regla Firewall.	81
Figura 88 Ingresar al VPS.	82
Figura 89 Preparación del Entorno.	83
Figura 90 Instalación Certbot y Generación del Certificado.....	84
Figura 91 Configuración de MySql.	85
Figura 92 Configuración de seguridad en MySQL.	86
Figura 93 Creación de Usuarios en MySQL.	86
Figura 94 Aplicación HeidiSql e ingreso a MariaDB mediante Mobaxterm.	87
Figura 95 Configuración de sqlheidy para conexión con IP externa.	88
Figura 96 Configuraciones.	88
Figura 97 Configuración de HeidiSQL e ingreso a interfaz.	89
Figura 98 Instalación y configuración de EMQX.....	90
Figura 99 Reinicio de VPS, EMQX.....	91
Figura 100 Configuración de Nodename.	92
Figura 101 Creación de base de Datos MqttIoT_bd.	93
Figura 102 Creación de Tabla mqtt_user	93
Figura 103 Creación de Tabla mqtt_acl y ejecución de comandos.....	94
Figura 104 Adición de Nuevo Usuario Web_iot_cliente.	95

Figura 105 Configuración de comunicación para EMQX con MySQL.....	96
Figura 106 Certificados en EMQX.	97
Figura 107 Verificación de Conexión Segura.	98
Figura 108 Certificación en EMQX y WSS.	98
Figura 109 Verificación de conexión segura WSS.	99
Figura 110 Instalación Node.js y GIT.....	100
Figura 111 Instalación de Node-red.....	100
Figura 112 Iniciar Node-red.	101
Figura 113 Ingreso a Node-red.	101
Figura 114 Creación usuario y contraseña.	102
Figura 115 Acceso a menú para agregar nodos.	102
Figura 116 Instalar nodos.	103
Figura 117 Ingreso al archivo de configuración.	104
Figura 118 Certificado https.	104
Figura 119 Ingreso seguro Node-red.....	104
Figura 120 Agregar archivo de instalación a MobaXterm.	105
Figura 121 Instalación de Grafana.	105
Figura 122 Acceso Grafana por primera vez.....	106
Figura 123 Modificación archivo grafana.ini.	106
Figura 124 Mover archivos desde apache a Grafana.	107
Figura 125 Ingreso al archivo grafan.ini.	107
Figura 126 Habilitar protocolo https.	108
Figura 127 Establecer rutas para los archivos.	108
Figura 128 Reinicio Grafana.	108
Figura 129 Verificar acceso seguro.	109
Figura 130 Código ejemplo para Arduino IDE – ESP32-EMQX.	110
Figura 131 Conexión de ESP 32, Topic, Visualización de datos.	111
Figura 132 Interacción entre MQTTX y Arduino.....	112
Figura 133 Configuración de código Prototipos IoT.	113
Figura 134 Prototipos Conectados.	113
Figura 135 MQTT Explorer.	114
Figura 136 Tablas creadas en Base de Datos.....	115
Figura 137 Node Red, Nodos de Casa 1 y Casa 2.	116
Figura 138 Conexión de Base de Datos con Grafana.	117
Figura 139 Creación de Dashboards.	117

Figura 140 Creación de Dashboards y Usuarios.....	118
Figura 141 Medición de Variables eléctricas.	119
Figura 142 Instalación de Prototipo IoT y colocación de pinza amperimétrica.	119
Figura 143 Funcionamiento del Prototipo IoT.	120
Figura 144 Mediciones en Oled de 1.3"	120
Figura 145 Monitoreo en Grafana.....	121
Figura 146 Ingreso a panel principal de Casa número 2.	122
Figura 147 Inducción al manejo de la página Web IoT.	123
Figura 148 Ingreso de credenciales por parte del usuario.	124
Figura 149 Visualización de Panel principal por parte de usuario	124
Figura 150 Interpretación de parámetros y graficas de W y kWh.	125
Figura 151 Descripción al Usuario.	126
Figura 152 Pregunta # 1	127
Figura 153 Pregunta #2	128
Figura 154 Pregunta #3	128
Figura 155 Pregunta #4	129
Figura 156 Pregunta # 5	129
Figura 157 Panel Principal Casa 1.	130
Figura 158 Grafica de Potencia casa número 1.	131
Figura 159 Grafica de Consumo Eléctrico casa número 1.	132
Figura 160 Panel Principal Casa 2.	133
Figura 161 Grafica de Potencia casa número 2.	134
Figura 162 Grafica de Consumo Eléctrico casa número 2.	135
Figura 163 Panel Principal Casa 3.	136
Figura 164 Grafica de Potencia casa número 3.	137
Figura 165 Grafica de Consumo Eléctrico casa número 3.	138
Figura 166 Panel Principal Casa 4.	139
Figura 167 Grafica de Potencia casa número 4.	140
Figura 168 Grafica de Consumo Eléctrico casa número 4.	141

Lista de anexos

Anexo 1. Código Arduino IDE, ESP32, Sensor Pzem-004T, MQTT	146
Anexo 2. Datasheet modulo sensor Pzem-004T	151
Anexo 3. Datasheet pantalla oled i2c iic de 1.3 pulgadas	152
Anexo 4. Datasheet Esp32.....	153
Anexo 5. Árbol de problemas.....	154
Anexo 6. Árbol de Objetivos.....	155
Anexo 7. Cronograma de Actividades	156
Anexo 8. Presupuesto	157

INTRODUCCIÓN

El consumo eficiente de energía eléctrica se ha convertido en una prioridad global debido al incremento de la demanda y a la necesidad de reducir el impacto ambiental. En este contexto, el uso de tecnologías innovadoras para el monitoreo y gestión del consumo energético representa una solución clave para optimizar el uso de los recursos disponibles. La digitalización y la automatización han permitido la creación de herramientas que facilitan el acceso a información detallada sobre el consumo eléctrico, permitiendo a los usuarios tomar decisiones informadas para mejorar su eficiencia energética.

En el ámbito residencial, la falta de sistemas accesibles para la supervisión del consumo eléctrico impide que los hogares puedan identificar hábitos de uso ineficiente, lo que se traduce en costos elevados y un impacto ambiental innecesario. Actualmente, la mayoría de los consumidores solo tienen acceso a facturas mensuales sin un desglose detallado de su consumo, lo que limita su capacidad de ajustar su comportamiento energético de manera efectiva. Ante esta problemática, la implementación de soluciones tecnológicas basadas en el Internet de las Cosas (IoT) ha demostrado ser una alternativa viable para proporcionar datos en tiempo real y mejorar la eficiencia en el uso de la electricidad.

Este proyecto tiene como objetivo desarrollar una plataforma web para la gestión de datos de consumo eléctrico en hogares urbanos, utilizando sensores IoT y microcontroladores. La plataforma permitirá la recopilación, procesamiento y visualización de información relevante sobre el consumo energético, brindando a los usuarios herramientas para optimizar el uso de la electricidad y reducir costos. A través de una interfaz intuitiva, los consumidores podrán analizar sus patrones de consumo, identificar posibles anomalías y tomar decisiones fundamentadas para mejorar la eficiencia en el hogar.

La iniciativa se enmarca en la línea de investigación de Tecnología e Industria del Instituto Tecnológico Rumiñahui, alineándose con las tendencias actuales en Industria 4.0 y ciudades inteligentes. Su implementación no solo busca beneficiar a las familias al proporcionarles un mejor control sobre su consumo energético, sino que también fortalece el perfil del instituto como un centro de innovación tecnológica. Adicionalmente, el desarrollo de este sistema fomenta el aprendizaje práctico de los estudiantes en áreas como programación, electrónica e integración de tecnologías IoT, promoviendo la adquisición de competencias clave para el mercado laboral.

La plataforma propuesta está diseñada para ser adaptable y escalable, lo que permitirá su implementación en diversos entornos y su futura expansión a mayor escala. Además, su desarrollo empleará tecnologías de código abierto y componentes accesibles, asegurando una solución económicamente viable. Al integrar medición en tiempo real, análisis de datos y visualización interactiva, este proyecto contribuirá a la adopción de prácticas de consumo energético más eficientes, beneficiando tanto a los usuarios como al medio ambiente.

En conclusión, el desarrollo de una plataforma web para la gestión de datos de consumo eléctrico representa una alternativa innovadora y efectiva para mejorar la eficiencia energética en los hogares urbanos. La combinación de tecnologías IoT y análisis de datos permitirá a los usuarios tomar decisiones informadas, optimizar su consumo y reducir costos, al mismo tiempo que se fomenta el uso responsable de la electricidad. Con esta iniciativa, se busca generar un impacto positivo en la comunidad, promoviendo el desarrollo sostenible y la aplicación de soluciones tecnológicas en la vida cotidiana.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema

El crecimiento acelerado de las ciudades y el aumento de la demanda de energía eléctrica han convertido en una necesidad urgente la gestión eficiente de los recursos energéticos. A nivel global, la optimización del consumo eléctrico es considerada esencial para el desarrollo económico sostenible y la mitigación del impacto ambiental, dado que el sector residencial contribuye significativamente a las emisiones de gases de efecto invernadero. En este contexto, es indispensable que los usuarios de energía eléctrica en zonas urbanas cuenten con acceso continuo y preciso a la información sobre su consumo eléctrico, lo que les permitirá implementar estrategias de ahorro que reduzcan tanto los costos como la huella ambiental.

En Ecuador, los habitantes urbanos carecen de herramientas accesibles para monitorear y gestionar su consumo eléctrico en tiempo real. Esta limitación dificulta la identificación de patrones de uso y la implementación de ajustes que optimicen el gasto energético según las necesidades diarias. La falta de control y visibilidad sobre el consumo genera un uso ineficiente de la energía, resultando en costos elevados, ya que los usuarios solo disponen de facturas mensuales sin un desglose detallado ni acceso a datos específicos. Sin sistemas adecuados de monitoreo, la precisión de las facturas es incierta, y los usuarios no pueden identificar qué dispositivos o comportamientos incrementan su consumo, lo que afecta tanto la economía familiar como los objetivos de sostenibilidad del país.

Diversos estudios demuestran que los sistemas de monitoreo en tiempo real, mediante tecnologías IoT, pueden reducir significativamente el consumo energético al proporcionar información precisa y oportuna sobre el uso de electricidad. No obstante, en Ecuador, la implementación de estas tecnologías en los hogares urbanos sigue siendo limitada, lo que representa un obstáculo para una gestión energética eficaz. La falta de herramientas asequibles que proporcionen datos detallados e inmediatos impide a los usuarios mejorar la eficiencia energética y ajustar su consumo.

Este proyecto propone desarrollar una plataforma web que permita el monitoreo en tiempo real del consumo eléctrico en hogares urbanos mediante el uso de sensores IoT. La plataforma estará diseñada para recopilar y almacenar datos precisos sobre el consumo energético los

cuales estarán disponibles a través de una interfaz web. Con esta herramienta, los usuarios podrán analizar patrones de consumo, identificar anomalías o costos elevados y realizar ajustes que resulten en ahorro económico. Además, al promover un uso más responsable de la energía, el proyecto contribuirá a reducir las emisiones contaminantes, alineándose con los objetivos de sostenibilidad energética.

1.2. Preguntas de directrices

¿Cómo la integración de tecnologías IoT en hogares urbanos permite optimizar la medición, transmisión y visualización del consumo eléctrico en tiempo real?

¿Qué impacto tiene el acceso en tiempo real a los datos de consumo eléctrico en la toma de decisiones de los usuarios y en sus hábitos energéticos?

¿Cuáles son las principales barreras y oportunidades para la adopción de sistemas de monitoreo energético basados en IoT en Ecuador?

1.3. Objetivo General

Desarrollar una plataforma web que permita la recolección y gestión de datos de consumo eléctrico en cuatro hogares de zonas urbanas mediante el uso de sensores, microcontroladores y tecnologías de comunicación, brindando a los usuarios una herramienta para visualizar y analizar su consumo energético de manera continua, compararlo con la facturación oficial y fomentar un uso eficiente y controlado de los recursos.

1.4. Objetivos Especifico

OBJ 1: Analizar los fundamentos teóricos sobre el consumo eléctrico doméstico y la tecnología IoT aplicada al monitoreo de energía.

OBJ 2: Definir los requisitos técnicos necesarios para implementar la plataforma web, considerando los sensores, tecnologías de comunicación y el procesamiento de datos.

OBJ 3: Desarrollar la plataforma web que incorpore los datos recopilados, estructurando una interfaz que permita visualizar el consumo eléctrico y realizar comparaciones periódicas con las facturas de servicio.

1.5. Justificación

En la actualidad, las tecnologías de Internet de las Cosas (IoT) están revolucionando diversas industrias, incluida la gestión energética. IoT permite la interconexión de dispositivos inteligentes, facilitando la recopilación y el análisis de datos en tiempo real. En el ámbito energético, los medidores inteligentes y sistemas de monitoreo IoT ofrecen la posibilidad de optimizar el uso de la energía al proporcionar información precisa sobre los hábitos de consumo. Sin embargo, la falta de un sistema integrado para monitorear el consumo energético en los hogares urbanos limita la comprensión de estos hábitos y dificulta la validación de los costos reflejados en las facturas, lo que genera incertidumbre. Esta falta de control impide a los usuarios identificar patrones de consumo ineficiente, como el uso prolongado de aparatos de alto consumo o picos de demanda en horarios específicos, lo que conlleva un gasto innecesario y una carga económica adicional.

El consumo energético ineficiente también ejerce una presión considerable sobre la infraestructura eléctrica. Al operar cerca de su capacidad máxima, la red eléctrica sufre un mayor desgaste en sus componentes de transmisión y distribución, como transformadores y líneas de suministro. Este incremento en la demanda no solo eleva los costos de mantenimiento y la necesidad de expansión de la red, sino que también incrementa el riesgo de interrupciones en el servicio. A su vez, la alta demanda energética agrava los problemas ambientales, ya que se requieren más recursos y energía para satisfacer las necesidades de los usuarios, lo que contribuye al deterioro de la sostenibilidad del sistema energético.

La implementación de una plataforma web que permita la recopilación, análisis y visualización de datos energéticos en tiempo real ofrece una solución innovadora a esta problemática. Esta herramienta proporcionará a los usuarios una interfaz accesible y fácil de usar para monitorear su consumo, identificar patrones y tomar decisiones informadas para optimizar el uso de la energía. Además, permitirá ajustar las facturaciones según el consumo real, garantizando así una facturación más precisa y justa. A largo plazo, se promoverá un uso más eficiente de la energía, reduciendo costos para los usuarios y aliviando la presión sobre la red eléctrica. Con ello, no solo se optimizará el consumo en los hogares, sino que también se contribuirá a la sostenibilidad del sistema energético y al bienestar general de la comunidad.

1.6. Alcance

El proyecto abarcará la implementación de un prototipo funcional para monitorear y gestionar el consumo eléctrico en tiempo real en cuatro hogares urbanos, utilizando tecnología IoT. El sistema empleará el sensor PZEM-004T, diseñado para medir parámetros eléctricos en corriente alterna (AC). Los datos recopilados incluirán voltaje, corriente, potencia, energía, factor de potencia y frecuencia, los cuales serán transmitidos por un microcontrolador ESP32. La visualización se realizará localmente a través de una pantalla OLED de 1.3", mientras que el monitoreo remoto se llevará a cabo mediante la plataforma Blynk. En esta fase, se validará la precisión de los datos obtenidos bajo condiciones típicas de uso.

En una segunda fase, se desarrollará una plataforma web interactiva, que incluirá la implementación de un Servidor Privado Virtual (VPS) en Google Cloud, donde se alojarán los servicios necesarios para su funcionamiento. El flujo de información comenzará con los sensores PZEM-004T que capturarán los datos de consumo y los enviarán al microcontrolador ESP32 y este a su vez a un Broker MQTT, encargado de distribuir la información a los suscriptores. Luego, un servidor web con Node-RED procesará los datos y los almacenará en una base de datos MySQL, la misma que se gestionará con HeidiSQL para su posterior análisis. Finalmente, los usuarios podrán visualizar en Grafana los valores de voltaje, corriente, potencia, consumo energético y mediante un cálculo obtener un valor en dólares aproximado en base a los (kWh) consumidos, lo que permitirá comparar los resultados con las facturas eléctricas reales y generar estadísticas detalladas.

Aunque inicialmente el sistema se implementará en cuatro hogares, su diseño modular permitirá la integración de más viviendas en el futuro. Este enfoque no solo validará la funcionalidad del sistema en un entorno real, sino que también fomentará el uso de tecnologías IoT para optimizar el consumo eléctrico en zonas urbanas, contribuyendo a una mayor eficiencia energética.

CAPÍTULO II

MARCO DE REFERENCIAS

2.1. Fundamento teórico relacionado al trabajo de integración curricular

2.1.1. Energía eléctrica

El uso de la energía eléctrica se ha expandido globalmente, mejorando la calidad de vida al facilitar tareas domésticas e industriales, también es fundamental en los medios de comunicación, permitiendo el funcionamiento de la radio, la televisión y el internet, impulsando así el desarrollo tecnológico y el confort en la sociedad. Las empresas distribuidoras garantizan un suministro eléctrico confiable y continuo hasta el medidor del consumidor. Además, operan bajo normativas que establecen estándares mínimos de calidad, así como procedimientos técnicos para la medición y evaluación del servicio. (Lozano Guamán, 2020).

2.1.2. Voltaje

El voltaje, también conocido como tensión eléctrica, es la diferencia de potencial eléctrico entre dos puntos de un circuito. Se trata de la fuerza que impulsa el movimiento de los electrones a través de un conductor, generando corriente eléctrica. En términos convencionales, la corriente se considera en dirección opuesta al flujo de electrones. Esta magnitud se representa con la letra "V" y se mide en voltios (V). Para su medición, se emplea un instrumento denominado voltímetro. La generación de voltaje en un circuito es posible gracias a una fuente de energía eléctrica, denominada fuerza electromotriz (f.e.m), cuya unidad también es el voltio. Esta fuerza es proporcionada por dispositivos como pilas, baterías y alternadores, los cuales inducen el movimiento de electrones dentro de un sistema cerrado, generando así la tensión eléctrica necesaria para el funcionamiento de diversos circuitos.

En un circuito, la tensión eléctrica se encarga de hacer que los electrones se desplacen ordenadamente a través de las líneas conductoras, permitiendo la aparición de una corriente eléctrica y garantizando el correcto funcionamiento de los dispositivos conectados. (Ternium, 2020).

2.1.3. Corriente

La corriente eléctrica es el flujo ordenado de electrones que atraviesan un conductor en un circuito cerrado. Su magnitud es proporcional a la diferencia de potencial eléctrico entre los extremos del conductor y se representa con la letra "I". Su unidad de medida es el amperio (A), que se define como el paso de una carga eléctrica de un coulomb (C) por segundo a través de un punto fijo del circuito. Cuando se aplica un voltaje a un material conductor, los electrones libres en el mismo se desplazan de un átomo a otro, formando un flujo continuo de carga eléctrica. La dirección de la corriente depende de la polaridad de la fuente de voltaje, estableciendo un flujo de electrones desde el polo negativo al positivo en corriente continua (DC) o alternando su dirección periódicamente en corriente alterna (AC). (Ternium, 2020).

Un amperio equivale al flujo de aproximadamente 6.28×10^{18} electrones por segundo a través de un conductor. La intensidad de la corriente depende de la resistencia del material y de la tensión aplicada, según la ley de Ohm, que establece que $I = V/R$, donde "I" es la corriente en amperios, "V" es el voltaje en voltios y "R" es la resistencia en ohmios (Ω). (Ternium, 2020).

2.1.4. Potencia Eléctrica

La potencia se refiere a la velocidad con la que se convierte una forma de energía en otra dentro de un período de tiempo específico. En el caso particular de la potencia eléctrica, esta representa la cantidad de energía eléctrica que un dispositivo es capaz de consumir o producir durante un intervalo de tiempo definido. (Hernández Acevedo, 2016).

$$Potencia (W) = \frac{Energia (J)}{Tiempo (s)}$$

2.1.5. Cálculo de consumo eléctrico

En el Sistema Internacional de Unidades (SI), la energía se mide en julios [J], la potencia en vatios [W] y el tiempo en segundos [s]. Esto implica que, al conocer la potencia de un dispositivo en vatios y el tiempo de funcionamiento en segundos, se puede calcular la cantidad de energía eléctrica transformada en otras formas, como luz, calor o movimiento. Debido a que el julio es una unidad relativamente pequeña, las compañías eléctricas prefieren medir la energía suministrada a los consumidores en kilovatios-hora (kWh). Un kilovatio-hora

real, como modificar la temperatura del aire acondicionado. (Rivera Berrío & Lopera Sánchez, 2024).

B. Capa de Red: Se encarga de transformar los datos recolectados por los sensores en formato digital para su procesamiento e intercambio. Utiliza conexiones a través de redes WAN inalámbricas (como Wi-Fi o redes móviles) o cableadas para transmitir la información. (Rivera Berrío & Lopera Sánchez, 2024).

C. Capa de Aplicación: Proporciona servicios de IoT orientados a diversos entornos, como hogares, ciudades o fábricas inteligentes. (Rivera Berrío & Lopera Sánchez, 2024).

Figura 2

Capas de la arquitectura IoT.



Nota: Tomado de ISTECS (05 mayo 2022)_Internet of things, IoT - Istec Digital.

2.2.2. Aplicaciones y Servicios de IoT

- **Seguridad y hogares inteligentes:** Mejora la eficiencia energética y la seguridad mediante sensores, alarmas y dispositivos inteligentes para el control remoto de electrodomésticos. Incluye termostatos, detectores de humo y cerraduras inteligentes, así como sistemas de salud y educación en casa. (IEEE, 2015).
- **Ciudades inteligentes y transporte:** Facilita la gestión eficiente del tráfico, el transporte público, el estacionamiento y la energía. Utiliza sensores para monitorear la

contaminación, administrar recursos como agua y basura, y coordinar emergencias. También promueve la participación ciudadana y la gobernanza digital. (IEEE, 2015).

- **Educación:** Conecta aulas virtuales y presenciales, mejora el acceso a bibliotecas digitales y fomenta el aprendizaje continuo y colaborativo. Ofrece herramientas para la gestión de asistencia y la educación personalizada. (IEEE, 2015).
- **Electrónica de consumo:** Incluye dispositivos como teléfonos, televisores y electrodomésticos inteligentes, así como gadgets personales como relojes, gafas y sensores para mascotas, optimizando la experiencia del usuario y la automatización. (IEEE, 2015).
- **Salud:** Permite el monitoreo remoto de pacientes, seguimiento de tratamientos y mejora de la calidad de vida con dispositivos como pulseras conectadas, etiquetas inteligentes para medicamentos y biochips para diagnósticos avanzados. (IEEE, 2015).
- **Automoción:** Integra sensores y sistemas inteligentes en vehículos para mejorar la seguridad, el diagnóstico en tiempo real y la conducción autónoma. Incluye control de tráfico y localización GPS. (IEEE, 2015).
- **Agricultura y medio ambiente:** Ayuda a monitorear el clima y la contaminación mediante sensores, además de optimizar la gestión de recursos agrícolas y residuos con etiquetas RFID y cálculos de nutrición. (IEEE, 2015).
- **Energía:** Ofrece medición y análisis inteligente del consumo energético, pronostica tendencias y fomenta la eficiencia mediante redes de sensores y tecnologías de reciclaje de energía. (IEEE, 2015).
- **Conectividad inteligente:** Proporciona servicios de comunicación en tiempo real, análisis de datos, computación en la nube, realidad aumentada y virtual, y seguridad en redes. (IEEE, 2015).
- **Fabricación:** Implementa sensores avanzados y aprendizaje automático para optimizar procesos, prevenir problemas y mejorar la logística y el control de calidad en la producción. (IEEE, 2015).
- **Compras:** Optimiza el comercio minorista mediante RFID, control de inventarios, trazabilidad de productos y mejora de la calidad y seguridad alimentaria. (IEEE, 2015).

2.2.3. Plataformas IOT

Existe variedad de plataformas que ofrecen distintos servicios y dependiendo del enfoque del proyecto es importante diferenciar entre plataformas genéricas y productos más específicos,

muchas de estas ofrecen versiones gratuitas con sus limitaciones y también versiones de paga para implementaciones más grandes. (Cantos, 2021).

2.3.3.1 Blynk

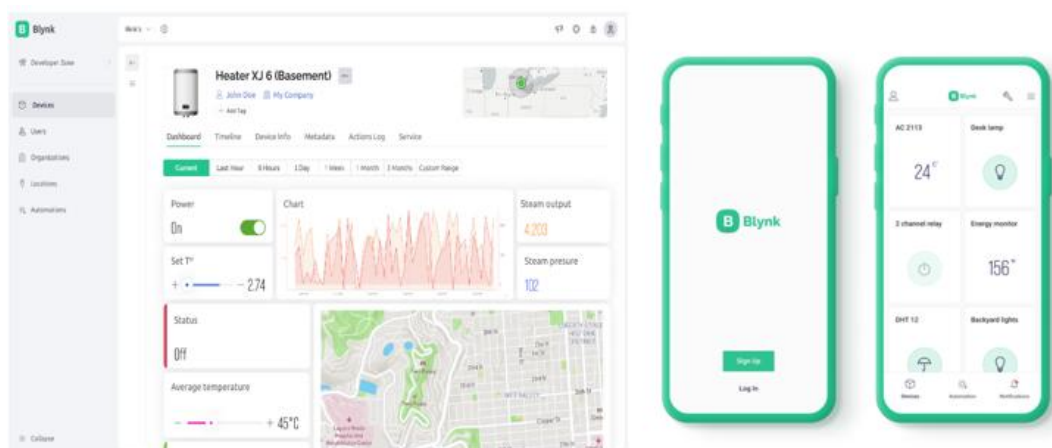
Plataforma IoT de bajo código diseñada para ayudar a empresas y desarrolladores a crear aplicaciones web y móviles completas para proyectos de hardware conectado. Con una interfaz intuitiva basada en el método de arrastrar y soltar, permite a los usuarios diseñar tableros de control en tiempo real, implementar widgets interactivos y recopilar datos. Es compatible con diversos componentes de hardware, como microcontroladores, ordenadores de placa única y dispositivos como Raspberry Pi. Además, Blynk incluye una biblioteca de API y wrappers que facilitan la conexión de proyectos con hardware de terceros y servicios en la nube. (Rosencrance, 2024).

Beneficios de plan gratuito:

- 2 dispositivos y 5 usuarios.
- 3 plantillas de dispositivos.
- Widgets gratuitos.
- Flujos de datos por plantilla.
- 1 semana de almacenamiento de datos históricos.

Figura 3

Blynk Console y Apps.



Nota: Interfaz Web y compatibilidad con IOS y Android para monitoreo remoto. Tomado de Blynk Documentation (02/12/2024) <https://docs.blynk.io/en>.

2.3.3.2 Kaa

Es una herramienta de código abierto que permite crear ecosistemas de dispositivos conectados. Esta solución destaca entre las opciones de tecnología en la nube por su diseño basado en micro servicios, compuesto por módulos independientes que pueden ensamblarse según las necesidades de cada proyecto. Gracias a su flexibilidad, los desarrolladores pueden diseñar implementaciones a medida para distintos sectores y aplicaciones específicas. También soporta una amplia gama de protocolos y formatos, lo que asegura su compatibilidad con una diversidad de componentes, como sensores avanzados, sistemas industriales y dispositivos móviles. (Rosencrance, 2024).

Beneficios plan gratuito:

- Hasta 5 dispositivos.
- 14 días de prueba gratuita.
- Acceso multiusuario.
- Canal de soporte comunitario en Gitter.
- Todas las funciones de Kaa Enterprise.

Figura 4

IoT Dashboards Kaa.



Nota: Tomado de KAA (02 diciembre de 2024) KAA IoT Cloud. Kaa IoT Platform.
<https://www.kaaiot.com/iot-dashboards>.

2.3.3.3 Mainflux

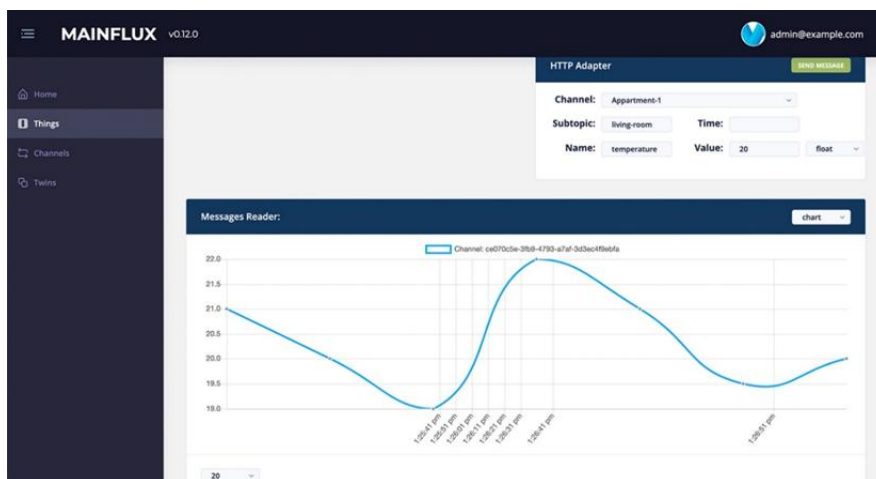
Ofrece una solución IoT de código abierto, sin restricciones de uso, desarrollada en Go y distribuida bajo licencia Apache 2.0. Esto permite que las compañías la incorporen en sus productos comerciales sin costos adicionales. Además, sirve como middleware para construir sistemas IoT avanzados. La arquitectura combina microservicios en contenedores Docker, gestionados mediante Kubernetes, y está diseñada para ser escalable, eficiente y confiable. Esta herramienta soporta múltiples protocolos, como MQTT, HTTP, CoAP y WebSocket. También dispone de SDKs, APIs integradas y gateways, lo que facilita a las organizaciones implementar proyectos en áreas como automatización del hogar, ciudades inteligentes y procesos industriales.(Rosencrance, 2024).

Beneficios plan gartuito:

- Alojamiento gratuito en cualquier plataforma compatible que funcione con Linux o Docker.
- Soporte de la comunidad.

Figura 5

Dashboards Mainflux.



Nota: Tomado de MAINFLUX LABS (02 diciembre 2024) Remote Monitoring and Management of IoT Devices & Gateways. <https://mainflux.com/iot-device-management.html>.

2.3.3.4 OpenRemote

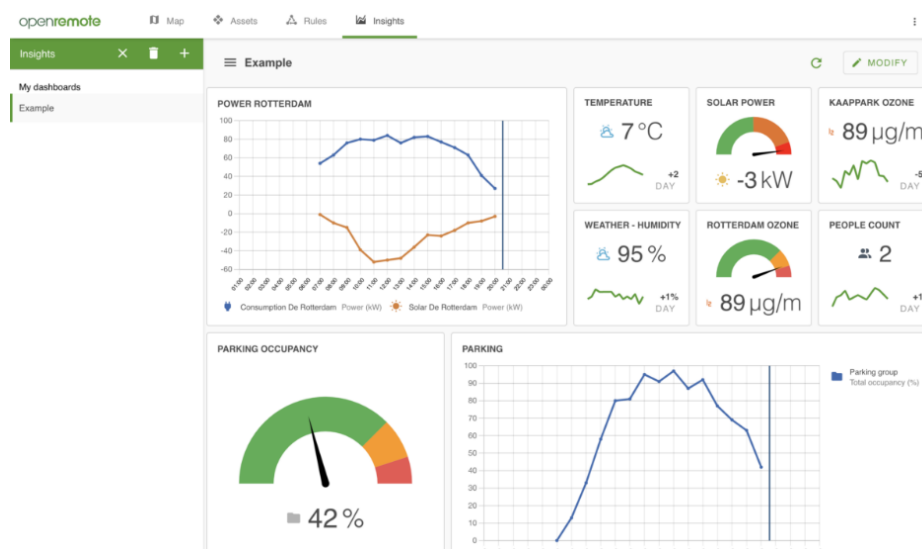
Ofrece un software de libre acceso que facilita la conexión y supervisión de dispositivos y sensores inteligentes, la recopilación y evaluación de información, así como la implementación de soluciones automatizadas. La herramienta incluye una interfaz intuitiva, capacidades para mostrar datos en tiempo real, análisis avanzados y la posibilidad de integrarse con otros sistemas tecnológicos. Además, es adaptable y puede ampliarse para satisfacer necesidades en diversos entornos, como hogares y urbes modernas, agricultura de precisión, administración de equipos conectados y optimización del uso energético.(Rosencrance, 2024).

Beneficios plan gratuito:

- Una plataforma IoT de código abierto 100% gratuita con capacidades ilimitadas.
- Una comunidad activa de usuarios y colaboradores, sin embargo, ofrece un servicio de asistencia técnica de pago.

Figura 6

Data Visualización.



Nota: Tomado de OpenRemote (02 diciembre 2024) Open source enterprise IoT platform | OpenRemote.

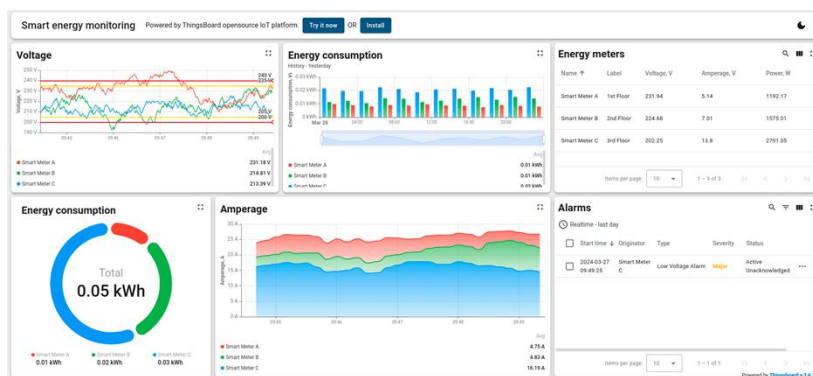
2.3.3.5 ThingsBoard

Es una herramienta de software libre que permite a las empresas gestionar equipos conectados y supervisar datos generados por dispositivos IoT. Facilita la conexión de sistemas, el almacenamiento de registros y el análisis de información procedente de sensores, gateways

y controladores. Esta plataforma ofrece un entorno sencillo para administrar equipos, establecer notificaciones, definir reglas, diseñar paneles y explorar métricas al instante. ThingsBoard se adapta a diferentes escenarios, integrando opciones avanzadas de seguridad y personalización. Está diseñada para sectores como agricultura, logística, urbanización y procesos industriales.(Rosencrance, 2024).

Figura 7

Monitoreo de energía inteligente.



Nota: Variedad de dashboards para monitoreo energetico. Tomado de ThingsBoard (02 diciembre 2024) IoT Energy Management & Monitoring. <https://thingsboard.io/smart-energy/>.

2.3. Microcontroladores

Un microcontrolador es un circuito integrado que integra en su interior una unidad central de procesamiento (CPU), módulos de memoria (RAM y ROM), puertos de entrada y salida, y diversos periféricos. Todos estos componentes están conectados internamente, conformando lo que se denomina una microcomputadora. En esencia, un microcontrolador puede considerarse una microcomputadora completa encapsulada dentro de un único circuito integrado.(Electrónica Estudio, 2019).

Figura 8

Microcontroladores.



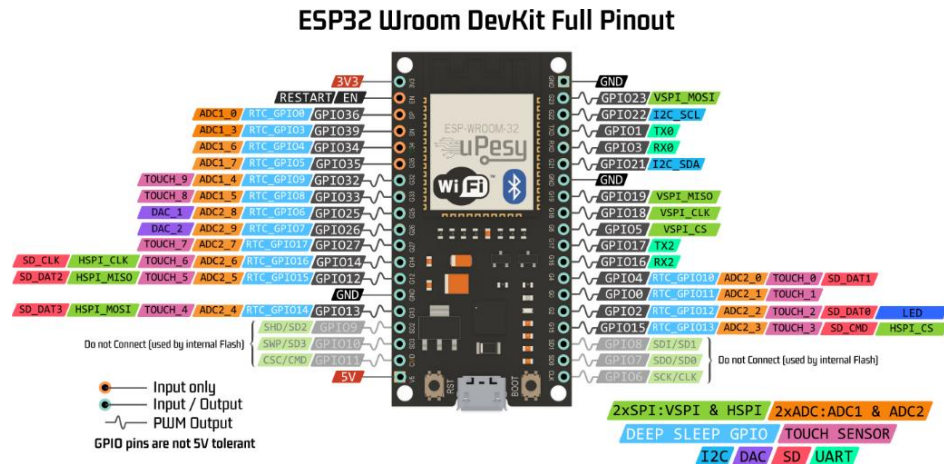
Nota: Tipos generales de microcontroladores tomado de Ariath Technology (02 diciembre de 2024) Análisis integral de microcontroladores. <https://acortar.link/oKYkFd>.

2.3.1. Microcontrolador ESP 32

El ESP32, desarrollado por el fabricante chino Espressif Systems, es un SoC (System on a Chip) diseñado como una versión mejorada del ESP8266. Este microcontrolador incorpora conectividad WiFi y Bluetooth, lo que lo hace ideal para proyectos de IoT debido a su versatilidad y tamaño compacto, facilitando su integración en circuitos impresos. Equipa un procesador Xtensa LX6 de dos núcleos, que permite la ejecución simultánea de dos procesos, y dispone de un modo de bajo consumo energético (deep sleep), haciéndolo eficiente en términos de energía y coste. Además, incluye tecnologías avanzadas como amplificadores de potencia y receptores de bajo ruido que optimizan sus comunicaciones inalámbricas(Ikiss, 2020).

Figura 9

Distribución de pines ESP 32 WROOM.



Nota: Tomado de uPesy (05 diciembre de 2024) ESP32 Pinout guide. <https://acortar.link/JeOCZC>.

2.4. Componentes de medición y visualización

2.4.1. Modulo sensor Pzem-004T

El PZEM-004T es un módulo sensor diseñado para la medición de parámetros eléctricos como voltaje, corriente, potencia y energía en sistemas de corriente alterna. Incorpora un sensor de voltaje y un transformador de corriente (CT) para una lectura precisa de los valores eléctricos. Este dispositivo se comunica mediante UART, facilitando su integración con microcontroladores como el ESP32 o Arduino. Está diseñado para uso en interiores y debe operar dentro de los límites de potencia especificados para garantizar un funcionamiento seguro y confiable.(Harahap et al., 2020).

Figura 10

Sensor de PZEM-004T V3.



Nota: Tomado de InnovatorsGuru (13 febrero 2024) PZEM-004T V3 Module | Arduino & NodeMCU Code, Circuit, Pinout And Library. <https://innovatorsguru.com/pzem-004t-v3/>.

Tabla 1*Datos técnicos sensor Pzem-004T.*

Símbolo	Parámetro	Clasificación
V_{IN}	Voltaje de entrada	80 – 280 V AC
I_{PN}	Corriente nominal	0 - 100 A
P_M	Potencia medida	0 – 9999kW
V_{CC}	Voltaje de alimentación	5 V DC
C_{omm}	Comunicación	UART (TTL Serial)
F_{unc}	Funciones de medición	V, A, W, kWh, Fp, Hz
M_{em}	Almacenamiento de datos	Si guarda energía tras apagarse
$P_{resicion}$	Margen de error	$\pm 0.5\%$ y 1%

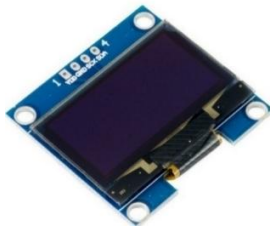
Nota: Esta tabla muestra las características técnicas del sensor Pzem-004T de acuerdo con el Datasheet. Tomado de Datasheet-User-Manual.pdf at main · vortigont/pzem-edl.

2.4.2. Pantalla OLED

La pantalla OLED SSH1106 de 1.3 pulgadas es un display gráfico de 128x64 píxeles, que ofrece alto contraste y nitidez gracias a la tecnología OLED (Organic Light-Emitting Diode), la cual elimina la necesidad de retroiluminación al emitir luz propia, reduciendo así el consumo energético. Utiliza la interfaz I2C (Inter-Integrated Circuit) para la comunicación con microcontroladores mediante dos líneas la SCL (Serial Clock Line) para la sincronización y la SDA (Serial Data Line) para la transmisión de datos. Su controlador SSH1106 optimiza la gestión de los píxeles, lo que la hace ideal para aplicaciones en dispositivos portátiles, instrumentación y sistemas embebidos.

Figura 11

Pantalla OLED I2C IIC de 1.3 pulgadas.



Nota: Tomado de Naylamp Mechatronics - Perú. (13 febrero 2025) Display OLED 1.3" I2C 128*64 SH1106. <https://acortar.link/1OjVKH>.

Tabla 2

Datos técnicos pantalla oled 1.3 pulgadas.

Parámetro	Clasificación
Resolución	128 × 64 px
Comunicación	I2C SAD-SCL
Voltaje de operación	3.3 – 5 V DC
Librería compatible	Adafruit_SH1106, U8g2
Cantidad de texto	8 líneas x 21 caracteres

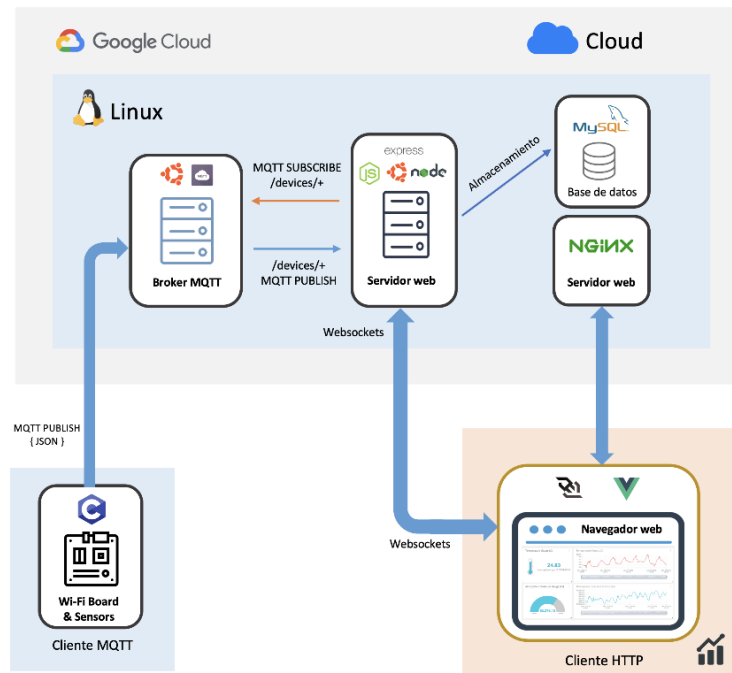
Nota: Esta tabla muestra las características de la pantalla oled de acuerdo con el Datasheet. Tomado de Pololu (2023). <https://www.pololu.com/file/0J1813/SH1106.pdf>.

2.5. Página web de IoT

Es una solución de software diseñada para integrar hardware, puntos de acceso y redes de datos, con la capacidad de recopilar, procesar, almacenar y visualizar en tiempo real los datos provenientes de sensores. Además, este tipo de sistemas debe ser capaz de garantizar la transmisión eficiente de la información mediante protocolos de comunicación adecuados, permitiendo la interacción entre dispositivos y usuarios a través de plataformas intuitivas. Esta funcionalidad es fundamental para aplicaciones en entornos IoT, donde el monitoreo y análisis de datos en tiempo real optimizan la toma de decisiones y el rendimiento del sistema.(Digital Academy, 2024).

Figura 12

Desarrollo de plataformas Internet Of Things.



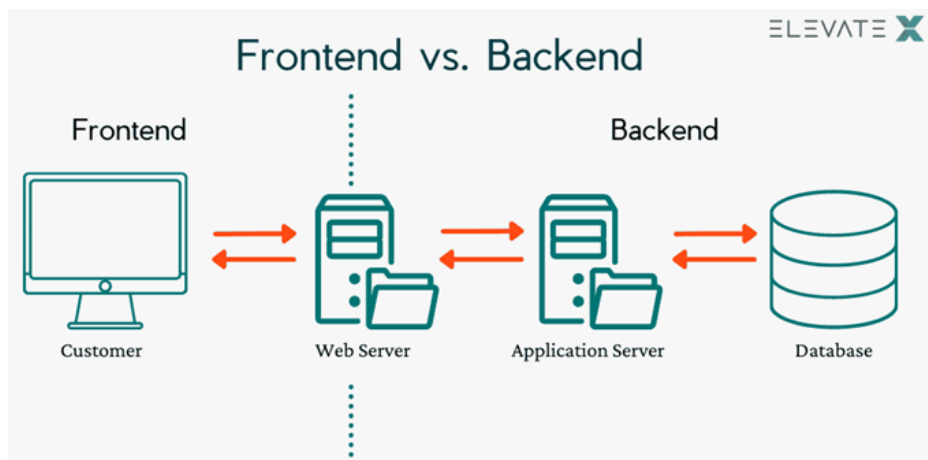
Nota: Arquitectura IoT. Tomado de *Dignal | Microcontroladores, electrónica, software*. (10 enero 2025). <https://dignal.com/>.

2.5.1. Frontend

El Frontend es la parte visible de una aplicación con la que los usuarios interactúan, también conocida como el lado del cliente. Representa todo lo que se muestra en la pantalla cuando un usuario accede a un sitio web o aplicación. Está compuesto por diversos elementos que permiten una navegación intuitiva y agradable. Entre estos destacan los tipos de letra, los colores, los efectos de interacción con el ratón o teclado, así como el diseño responsivo, que asegura la adaptación a diferentes tamaños de pantalla. Además, incluye efectos visuales, desplazamientos y otros detalles interactivos que, en conjunto, conforman la experiencia del usuario (UX), facilitando la navegación y la comprensión de la aplicación o página web. (Stefaniak, 2019).

Figura 13

Frontend vs. Backend.



Nota: Tomado de ElevateX. (10 de enero 2025). Frontend vs. Backend vs. Full-Stack – The Difference Explained. ElevateX. <https://acortar.link/84gMnW>.

2.5.1.1. Dashboards de Visualización

A. React

React es una herramienta diseñada para simplificar la creación de interfaces de usuario interactivas. Permite desarrollar vistas específicas para cada estado de una aplicación, gestionando de manera eficiente la actualización y renderización de los componentes necesarios cuando ocurren cambios en los datos. Aunque la lógica de los componentes se escribe directamente en JavaScript en lugar de utilizar plantillas, React facilita el flujo de datos a lo largo de la aplicación y permite mantener el estado de la misma fuera del DOM, optimizando su funcionalidad.(React, 2021).

B. Bootstrap

Bootstrap es una herramienta poderosa que facilita la creación de sitios web adaptativos o responsivos, capaces de ajustarse a diversos dispositivos y tamaños de pantalla.(Ovando Ortega, 2019).

Entre sus ventajas destacan:

- **Organización visual:** permite construir sitios web con un diseño limpio y estructurado.
- **Sistema de grillas:** utiliza un sistema basado en columnas que simplifica la maquetación del contenido. (Ovando Ortega, 2019).
- **Integración con JavaScript:** se combina de manera eficiente con diversas librerías para mejorar la funcionalidad. (Ovando Ortega, 2019).

- **Amplia comunidad:** cuenta con un respaldo sólido de una comunidad activa y extensa.(Ovando Ortega, 2019).
- **Compatibilidad con preprocesadores CSS:** admite el uso de herramientas como Sass y Less, optimizando la experiencia de desarrollo y personalización. (Ovando Ortega, 2019).

C. Grafana

Es una plataforma de código abierto flexible y poderosa que ha transformado cómo los desarrolladores visualizan y comprenden los datos. Al integrar diversas fuentes de información en un solo lugar, Grafana facilita la colaboración entre equipos, permitiendo personalizar la experiencia mediante complementos que se adaptan a necesidades específicas.(Khanh, 2021).

Entre las principales características de Grafana se encuentra la visualización de datos, que ofrece gráficos rápidos y altamente personalizables para mostrar métricas y registros en paneles dinámicos que pueden reutilizarse y ajustarse según los requisitos. Además, permite explorar métricas y registros a través de consultas ad-hoc y proporciona la capacidad de agregar anotaciones y combinar múltiples fuentes de datos en un único gráfico, lo que permite un análisis más profundo. Grafana también permite configurar alertas y recibir notificaciones en tiempo real, mejorando la capacidad de respuesta ante eventos importantes.(Khanh, 2021).

Figura 14

Analítica y monitorización con Grafana.



Nota: Analítica y monitorización con Grafana. Tomado de González, O. (15 de enero 2025). <https://2y2.co/GHvmze>.

2.5.2. Backend

El Backend es la parte del software responsable de gestionar el acceso a los datos y no es visible ni accesible para el usuario final. Su principal función es asegurar la funcionalidad, seguridad y la optimización de recursos en aplicaciones o sitios web. Esta capa contiene la lógica interna necesaria para que el sistema funcione correctamente, gestionando tareas como la conexión a bases de datos y el almacenamiento de información en servidores.(Pérez Ibarra et al., 2021).

El Backend facilita procesos esenciales como la ejecución de operaciones lógicas, la conectividad con bases de datos, la gestión eficiente de archivos y la optimización del rendimiento del sistema. Además, se utilizan herramientas y librerías especializadas para funciones específicas, como la compresión y pre visualización de imágenes, así como la mejora de la eficiencia general del sistema. (Pérez Ibarra et al., 2021).

2.5.2.1.Servicios Backend para IoT

A. Servidores en la Nube

Las soluciones de computación en la nube han transformado la manera en que empresas y desarrolladores implementan y administran sus aplicaciones. Entre los principales proveedores en este ámbito se encuentran Amazon Web Services (AWS) y Google Cloud Platform (GCP), los cuales brindan herramientas escalables y eficientes para el almacenamiento, procesamiento y gestión de datos.(Gupta et al., 2021).

AWS, una de las plataformas en la nube más consolidadas desde su lanzamiento en 2006, ofrece una amplia variedad de servicios que incluyen bases de datos, almacenamiento, análisis de datos, redes, computación móvil, Internet de las cosas (IoT) y soluciones empresariales. Actualmente, cuenta con 63 zonas de disponibilidad a nivel global, lo que garantiza alta disponibilidad y escalabilidad para sus clientes. Por su parte, Google Cloud Platform (GCP), introducida en 2011, proporciona servicios en la nube enfocados en almacenamiento, big data, bases de datos, inteligencia artificial (IA), seguridad y gestión de redes. Su infraestructura abarca 21 zonas de disponibilidad en todo el mundo y destaca por su integración con herramientas avanzadas de análisis de datos y machine learning. (Gupta et al., 2021).

Figura 15

Servidores en la Nube.



Nota: Comparativa entre AWS vs. Azure vs. Google. Tomado de Peñaloza, J. (17 de enero 2025). <https://2y2.co/XdfHMz>.

B. Protocolo MQTT

El protocolo MQTT (Message Queuing Telemetry Transport), desarrollado por IBM, está diseñado para la comunicación entre máquinas (M2M). Su principal ventaja es el bajo consumo de ancho de banda, lo que lo hace ideal para dispositivos con recursos limitados. Utiliza una arquitectura en estrella, donde un bróker central gestiona la comunicación a través de un esquema de publicación/suscripción.(Guasch Llobera & Calleja Collado, 2019).

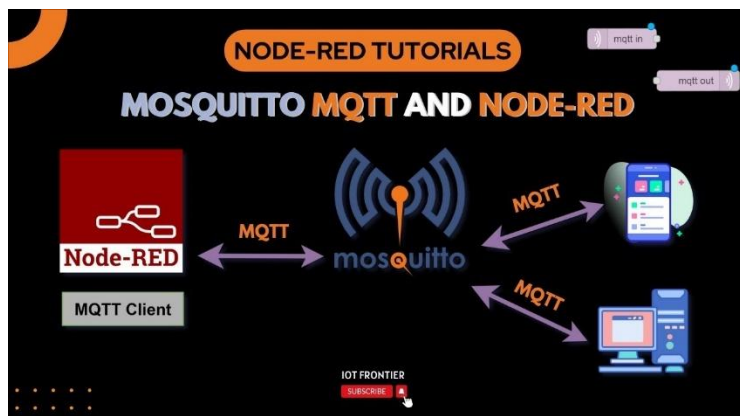
Dentro de la arquitectura de MQTT, un elemento clave es el topic, que estructura la comunicación. Para que un emisor y un receptor puedan intercambiar mensajes, ambos deben estar suscritos a un mismo topic, asegurando una comunicación eficiente y organizada.(Guasch Llobera & Calleja Collado, 2019).

C. Eclipse Mosquitto

Eclipse Mosquitto es un bróker de mensajería de código abierto desarrollado por Roger Light, que implementa el protocolo MQTT para comunicación asíncrona. Su principal ventaja es su ligereza, ya que puede ejecutarse tanto en dispositivos con recursos limitados como en servidores de alto rendimiento. Su consumo de recursos es mínimo, requiriendo solo 3 MB de memoria por cada 1000 clientes conectados, lo que lo convierte en una opción altamente eficiente. Mosquitto forma parte de la Fundación Eclipse y se basa en una librería en C, permitiendo la implementación de clientes MQTT. Además, es compatible con múltiples sistemas operativos, incluyendo Windows, CentOS y Ubuntu. (Eclipse Mosquitto, 2020).

Figura 16

Eclipse Mosquitto MQTT Broker y Cliente.



Nota: Recuperado de Automation Community. (18 de febrero 2022). How to set up Eclipse Mosquitto MQTT Broker and connect a client. Bosch Rexroth <https://lc.cx/iDenTs>.

D. Node RED

Node-RED es una plataforma de desarrollo de código abierto, basada en JavaScript y creada por IBM, que permite la integración de hardware IoT, API y servicios en la nube. Su enfoque visual facilita la conexión de nodos de manera intuitiva, eliminando la necesidad de escribir largas líneas de código. Esto permite a los usuarios añadir, eliminar y configurar nodos para gestionar flujos de datos en tiempo real, optimizando el envío, recepción y procesamiento de información. (Carrillo Baquerizo & León Arias, 2021).

Node-RED cuenta con tres componentes principales

- **Panel de Nodos:** Contiene los elementos funcionales para construir flujos.
- **Panel de Flujo:** Muestra la estructura y conexión entre nodos.
- **Panel de Depuración e Información:** Facilita la identificación y solución de errores en los flujos.

E. Base de datos

La información se considera un conjunto de datos que, al ser analizados, facilitan la toma de decisiones. Este conjunto se procesa y presenta de forma clara para permitir una mejor interpretación, ayudando a identificar tendencias o descubrir patrones. En este contexto, una base de datos se define como un sistema organizado que permite almacenar, gestionar y recuperar información, desempeñando un papel crucial en la estructura de los sistemas IoT. (Recalde & Zambrano, 2022).

- **Bases de datos NoSQL:** Estas son ideales para manejar grandes volúmenes de datos no estructurados que se generan en entornos IoT. Ejemplos destacados de este tipo incluyen MongoDB y Cassandra. (Recalde & Zambrano, 2022).
- **Bases de datos en tiempo real:** Diseñadas para procesar datos de forma inmediata, son especialmente útiles en sistemas IoT que generan información continuamente y requieren respuesta inmediata. Ejemplos comunes son Apache Kafka y Firebase Realtime Data base. (Recalde & Zambrano, 2022).
- **Bases de datos relacionales:** Reconocidas por su precisión, consistencia y capacidad para realizar consultas complejas, resultan adecuadas para sistemas IoT que trabajan con datos estructurados. Ejemplos habituales incluyen MySQL y PostgreSQL. (Recalde & Zambrano, 2022).

La selección del tipo de base de datos depende directamente de los requerimientos específicos del sistema IoT en cuestión. Cada tipo de base de datos ofrece ventajas únicas y posibles limitaciones, por lo que la elección debe alinearse con los objetivos del sistema en su conjunto.

▪ **MySQL**

MySQL es una de las bases de datos más utilizadas a nivel mundial debido a su velocidad, estabilidad y facilidad de configuración. Su arquitectura optimizada permite un alto rendimiento, gracias a la reutilización de código dentro del software. Se trata de un sistema de gestión de bases de datos relacional (RDBMS) que organiza la información en tablas interconectadas mediante relaciones, lo que facilita la combinación de datos y mejora la eficiencia en las consultas. MySQL fue desarrollado por la empresa MySQL AB, fundada por sus propios creadores.(Oracle, 2025).

Una de sus características más destacadas es la separación del motor de almacenamiento del resto de los componentes de su arquitectura, lo que permite el desarrollo de motores específicos según las necesidades de cada aplicación. Esto proporciona mayor flexibilidad y adaptabilidad para distintos entornos y volúmenes de datos. (Oracle, 2025).

Figura 17

Bases de datos.



Nota: Tomado de Desarrollo Web. (14 febrero 2025). Bases de datos.
<https://desarrolloweb.com/home/bases-de-datos>.

CAPÍTULO III

MARCO METODOLÓGICO

3.1. El método científico

3.1.1. Métodos teóricos de investigación

3.1.1.1.Fase de Diseño

Se llevó a cabo un estudio descriptivo para evaluar la viabilidad técnica, operativa y económica del sistema de monitoreo energético. A través del método de observación y la aplicación de encuestas, se recopilaban datos sobre las condiciones del entorno y las necesidades de los usuarios, lo que permitió identificar oportunidades y limitaciones en su implementación.

Posteriormente, mediante un enfoque exploratorio y utilizando el método hipotético-deductivo, se definió la disposición de los sensores, microcontroladores y módulos de comunicación. A través de listas de chequeo y registros en diarios de campo, se documentaron las pruebas realizadas para optimizar la captura y transmisión de datos, permitiendo ajustar el diseño del prototipo de acuerdo con los requerimientos establecidos.

El desarrollo conceptual de la plataforma web se abordó formulando hipótesis sobre la mejor manera de presentar la información obtenida por el sistema. Mediante entrevistas no estructuradas, se analizaron las necesidades de los usuarios para definir una interfaz intuitiva y eficiente.

De forma paralela, se diseñó el circuito electrónico y la PCB, asegurando la integración de los componentes y su correcto funcionamiento. El registro anecdótico permitió documentar cada etapa del proceso, lo que facilitó la identificación y corrección de posibles fallas antes de la fabricación final.

Finalmente, se elaboró el modelo 3D de la carcasa protectora, considerando criterios de resistencia estructural y compatibilidad con los elementos internos. A partir de observaciones registradas en diarios de campo, se realizaron ajustes en el diseño para garantizar su funcionalidad y eficiencia en la protección del sistema. El esquema gráfico de diseño del proyecto lo veremos en el siguiente capítulo.

3.1.1.2.Fase de Implementación

Para materializar el diseño propuesto, se imprimió en 3D la carcasa protectora del prototipo, evaluando su precisión dimensional y resistencia mecánica. A través de listas de chequeo, se verificó el cumplimiento de los estándares definidos, permitiendo realizar ajustes en caso de ser necesario.

La fabricación de la PCB y el montaje de los componentes electrónicos se llevaron a cabo mediante un proceso sistemático de medición, garantizando la correcta integración de los elementos. Se utilizaron listas de verificación y aparatos de medición para validar la continuidad de las conexiones y detectar posibles errores en la ensambladura.

En el ámbito del software, se desarrolló y optimizó el código para el microcontrolador ESP32, asegurando su eficiencia en la adquisición, procesamiento y transmisión de datos. Mediante registros anecdóticos y diarios de campo, se documentó cada iteración del proceso de programación, permitiendo la depuración y ajuste del código según los requerimientos del sistema.

Simultáneamente, se implementó la plataforma web IoT, estructurando su arquitectura y optimizando la visualización de datos en tiempo real. A través del método analítico y registros en diarios de campo, se llevaron a cabo pruebas funcionales que permitieron corregir errores y mejorar la experiencia del usuario.

Finalmente, se integró el prototipo con la plataforma web, verificando la estabilidad en la transmisión y visualización de la información. Mediante experimentación científica y registros anecdóticos, se documentaron ajustes en los protocolos de comunicación para garantizar la precisión del monitoreo energético.

3.1.1.3.Fase de Evaluación.

Se verificó el acceso y correcto funcionamiento de la plataforma web, asegurando la disponibilidad y estabilidad del sistema. Se utilizaron mediciones documentadas en diarios de campo para evaluar tiempos de carga, posibles errores en la interfaz y compatibilidad con distintos dispositivos.

Se llevó a cabo una inducción sobre el manejo de la plataforma, asegurando que los usuarios comprendieran su funcionamiento. Mediante pruebas de conocimiento y observaciones registradas en diarios de campo, se identificaron áreas de mejora en la interfaz y en la presentación de la información.

Además, se recopiló retroalimentación a través de encuestas, permitiendo evaluar la experiencia de los usuarios con el sistema de monitoreo. A partir de estos datos, se identificaron oportunidades de optimización en la usabilidad y efectividad del sistema.

Finalmente, se evaluaron los datos obtenidos a través de los Dashboards principales de cada prototipo IoT en la plataforma web. Se analizaron los patrones de consumo energético en tiempo real, permitiendo visualizar tendencias y variaciones en el uso de la energía. A partir de esta información, se validó la efectividad del sistema en la optimización del consumo eléctrico y su impacto en la gestión energética de los hogares.

3.1.2. Técnicas e Instrumentos de recolección de datos

Tabla

3

Técnicas e Instrumentos de recolección de datos.

FASE	ETAPA	TÉCNICA	INSTRUMENTO
Fase de Diseño	1. Estudio de viabilidad para el desarrollo del sistema de monitoreo energético	Observación	Lista de chequeos, Encuestas
	2. Diseño del esquema para el prototipo de monitoreo	Observación, Experimentación	Simulación.
	3. Diseño conceptual de la plataforma web	Encuestas, Entrevistas	Cuestionarios, Guías de entrevista
	4. Diseño del circuito electrónico y PCB	Observación, Experimentación	Registro de anotaciones, Pruebas electrónicas
	5. Diseño 3D de la carcasa protectora del prototipo	Observación, Experimentación	Diario de campo, Prototipado
Fase de Implementación	1. Impresión 3D de la carcasa protectora del prototipo	Observación, Experimentación	Lista de chequeo, Pruebas estructurales
	2. Fabricación de la PCB y montaje de elementos electrónicos	Observación, Experimentación	Registro de avances, Pruebas electrónicas
	3. Desarrollo del código para ESP32	Observación, Evaluación	Pruebas de código, Verificación
	4. Desarrollo e implementación de la plataforma web IoT	Observación, Evaluación funcional	Registro de mejoras, Diario de campo
	5. Integración del prototipo con la plataforma web	Observación, Pruebas funcionales	Diario de campo, Evaluación
Fase de Evaluación	1. Verificación de acceso correcto a la plataforma	Observación	Diario de campo, Registro de accesos
	2. Inducción sobre el manejo de la plataforma.	Observación	Diario de campo, Análisis cualitativo
	3. Retroalimentación por parte de los usuarios	Encuestas	Cuestionarios
	4. Evaluación de Datos Visualizados en los Dashboards de la Plataforma Web IoT	Observación	Diario de campo, Análisis cualitativo

Nota: En esta tabla se resumen fases y etapas, así como los métodos, así como la técnica y los instrumentos utilizados para el desarrollo de cada una. Fuente: Elaboración propia.

CAPÍTULO IV

DESARROLLO DEL PROYECTO TÉCNICO

4.1. Diseño de la propuesta

4.1.1. Estudio de viabilidad para el desarrollo de un sistema de monitoreo energético.

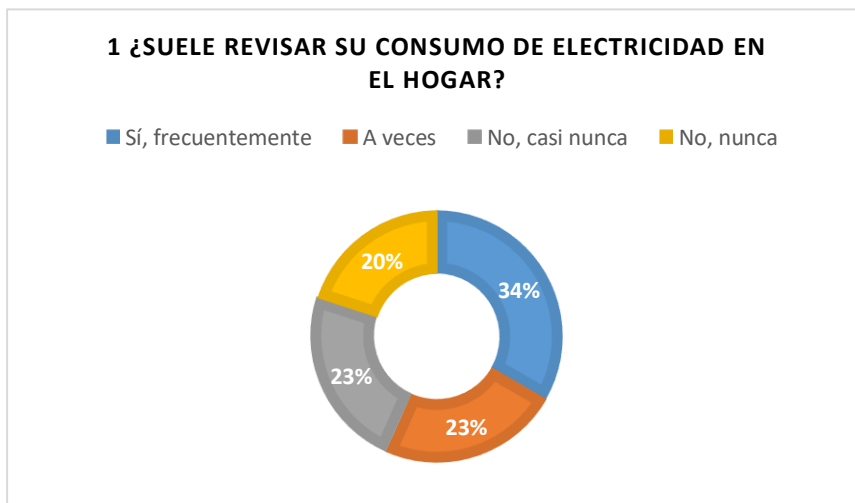
Para determinar la viabilidad del proyecto, se llevó a cabo una encuesta estructurada compuesta por cinco preguntas dirigidas a una muestra de 30 personas. El propósito de este proceso fue evaluar la percepción y aceptación de un sistema de monitoreo energético basado en IoT, identificando la necesidad real de su implementación.

A través del análisis de las respuestas obtenidas, se logró establecer el nivel de interés de los usuarios potenciales, así como sus preferencias en cuanto a la visualización y frecuencia de consulta de los datos de consumo eléctrico. Esto permitió definir con mayor precisión los requerimientos del sistema y validar su aplicabilidad en entornos domésticos y comerciales.

A continuación, se presentan los resultados obtenidos en cada pregunta, acompañados de sus respectivas representaciones gráficas que facilitan la interpretación de los datos recopilados.

Figura 18

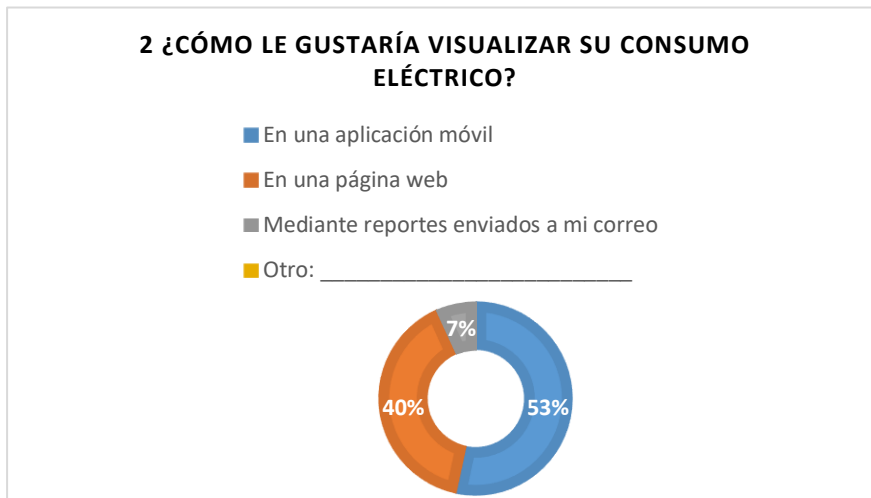
Pregunta # 1



Nota: En esta imagen se observa que el 34% de las personas encuestadas revisan el consumo de electricidad en su hogar. Tomado de autoría propia (22 febrero 2025).

Figura 19

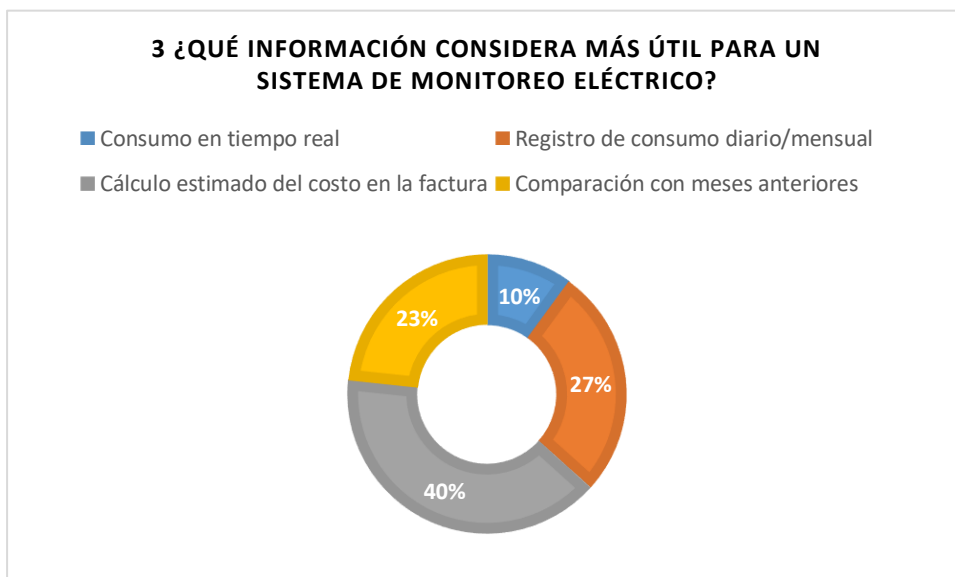
Pregunta # 2



Nota: En esta pregunta podemos definir las preferencias de la gente para poder observar información referente al consumo energético, las opciones preferidas son "Aplicación móvil" 53% y "Una página web" 40%. Tomado de Autoria propia (22 febrero 2025).

Figura 20

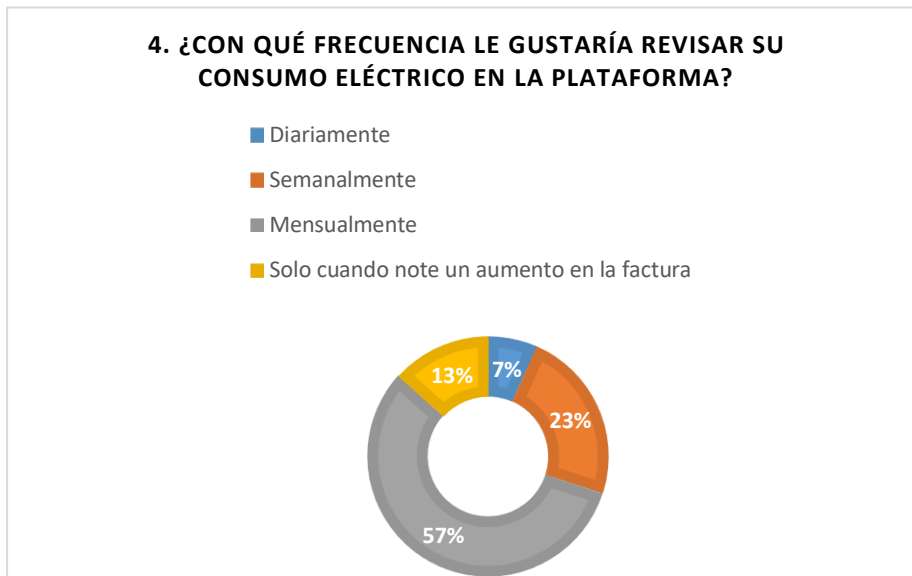
Pregunta # 3



Nota: En esta figura se puede confirmar que la gente tiene gran interés en confirmar el valor a pagar si coincide con lo facturado por la empresa con el 40% y otra opción de interés es comparar esas mediciones con los meses anteriores 23%. Tomado de Autoria propia (22 febrero 2025).

Figura 21

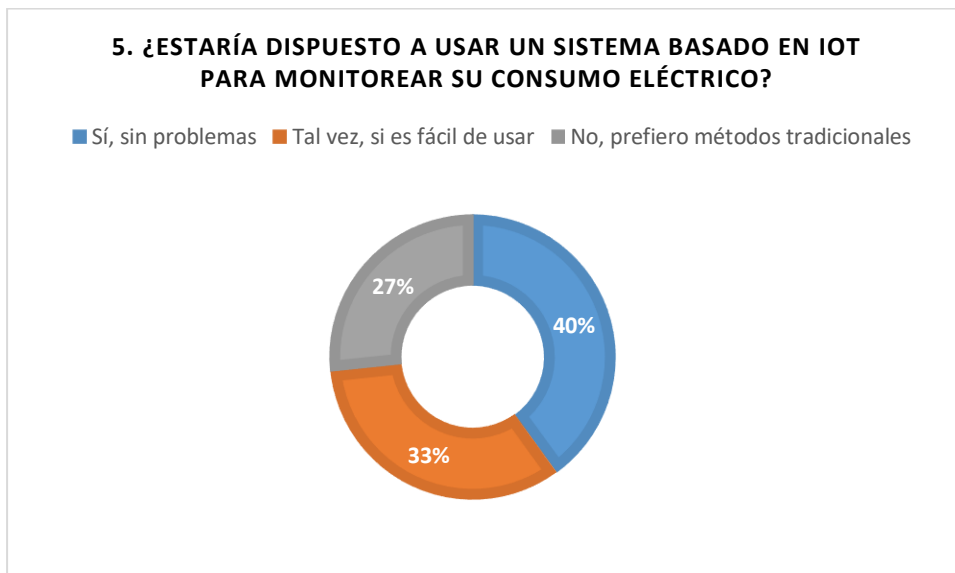
Pregunta # 4



Nota: Se observa que hay un alto porcentaje de personas que les gustaría revisar su consumo de manera semanal con el 23%, pero en su mayoría prefiere hacerlo de manera mensual con el 50%. Tomado de Autoría propia (22 febrero 2025).

Figura 22

Pregunta # 5



Nota: En esta figura se puede observar que la idea de usar un sistema IoT tiene buena aceptación, 40% un sí y el 33% está dispuesto si es sencillo de usar. Tomado de Autoría propia (22 febrero 2025).

4.1.2. Diseño del esquema para el prototipo de monitoreo.

Para esta etapa, se establecieron las conexiones eléctricas y de comunicación entre los componentes principales del medidor de consumo energético, utilizando diagramas y esquemas representativos que facilitaron su integración. Los elementos empleados en el diseño incluyen:

- Placa de desarrollo ESP32
- Pantalla OLED SSH1106 de 1.3 pulgadas
- Sensor PZEM-004T V3

En la Figura 23, se puede observar la conexión del sensor PZEM-004T V3, cuyo conector blanco ubicado en el lado derecho del módulo permite la comunicación serial mediante los pines TX y RX, los cuales fueron vinculados a los pines GPIO 16 y GPIO 17 del ESP32, respectivamente. Adicionalmente, desde este mismo puerto se suministró la alimentación al módulo, utilizando los pines VCC (5V) y GND provenientes del ESP32.

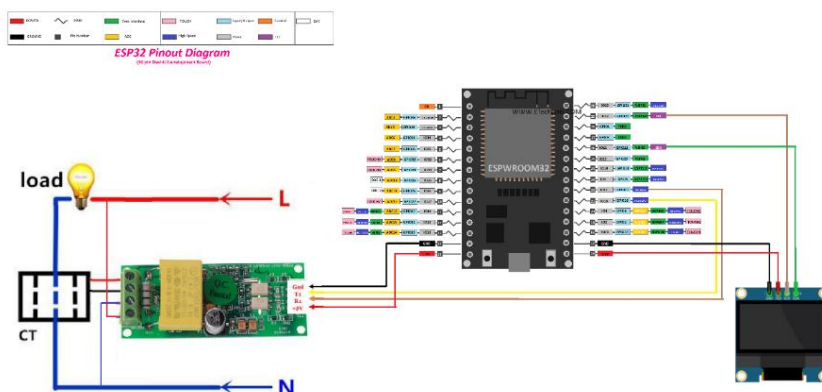
Por otro lado, en el extremo opuesto del sensor, la bornera de conexión cumple la función de entrada de alimentación AC (120V), donde se estableció la conexión con la red eléctrica, permitiendo el funcionamiento del sistema. Asimismo, en esta sección se incorporó la pinza amperimétrica, encargada de realizar la medición no invasiva de corriente.

En cuanto a la pantalla OLED SSH1106, esta fue alimentada desde los pines GND y VCC (3.3V) del ESP32 y configurada para recibir datos a través del protocolo I2C, estableciendo la comunicación con los pines GPIO 21 (SDA) y GPIO 22 (SCL). De esta manera, se logró la visualización en tiempo real de los valores obtenidos por el sistema de monitoreo energético.

• ELEBORACIOION DEL DISENO DEL EQUEMA IoT

Figura 23

Diseño de conexión de componentes.



Nota: Se puede ver los pines que se usa del ESP32 y como deben ir conectados los elementos de medición. Tomado de: Autoría propia (18 enero 2025).

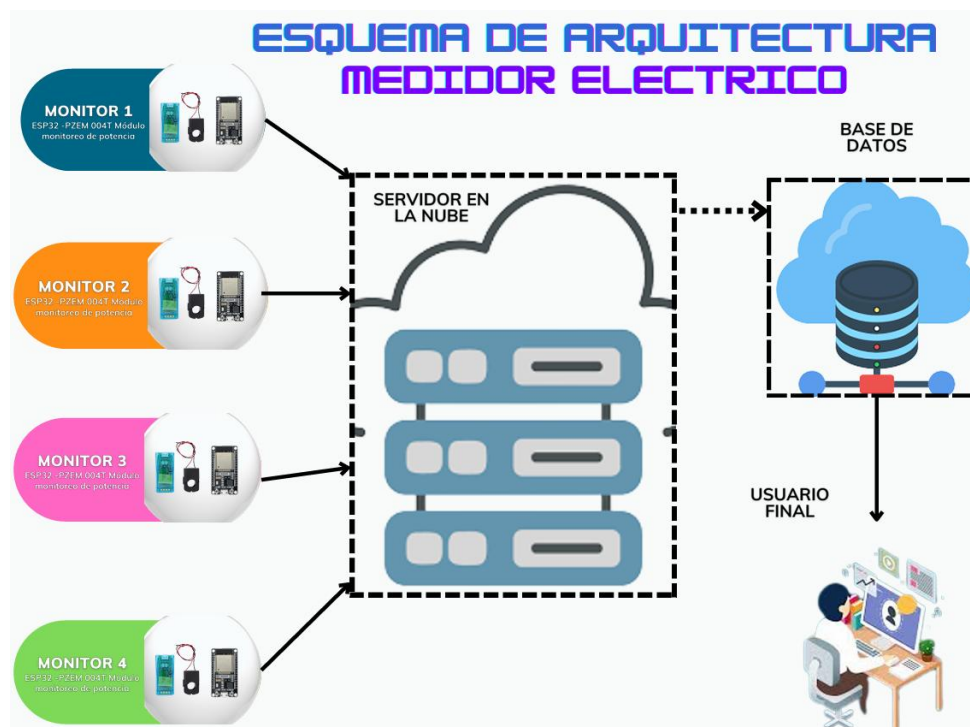
Adicionalmente, se definió la arquitectura general del proyecto, estableciendo la estructura de interconexión entre los distintos elementos. En la Figura 24, se representa el flujo de datos y la comunicación entre los componentes del sistema.

Los cuatro prototipos fueron configurados para transmitir las mediciones en tiempo real a través de una conexión a internet, enviando los datos a un servidor en la nube que alberga la base de datos central. Este repositorio de información permite el almacenamiento y gestión de los registros recopilados, asegurando su disponibilidad para su posterior análisis.

Finalmente, los usuarios pueden acceder a la información procesada mediante una página web, desde la cual es posible visualizar los valores obtenidos y realizar un monitoreo continuo del consumo energético. Esta arquitectura garantiza una comunicación eficiente entre los dispositivos, el servidor y la plataforma web, optimizando la operatividad del sistema.

Figura 24

Diseño conceptual de la plataforma Web.



Nota: Esta figura permite tener una idea visual más clara de cómo opera y se conecta medidor hasta llevar la información a la página web. Tomado de Autoría propia (19 enero 2025).

4.1.3. Diseño conceptual de la plataforma web.

Para esta etapa, se llevaron a cabo entrevistas no estructuradas con expertos en el área, con el propósito de definir de manera más precisa la visualización y funcionalidad de la página web. Como parte de este proceso, se desarrolló una presentación en PowerPoint, la cual permitió estructurar una propuesta inicial sobre la interfaz y disposición de los elementos dentro de la plataforma.

En la Figura 25, se presenta la portada del sistema, donde se incluyen los datos generales del proyecto. Posteriormente, en la Figura 26, se muestra la ventana de acceso, diseñada para la autenticación de los usuarios mediante la introducción de credenciales. Finalmente, en la Figura 27, se exhibe la interfaz principal, en la cual se despliega la información detallada sobre el consumo energético registrado por el sistema, permitiendo su monitoreo en tiempo real.

Figura 25

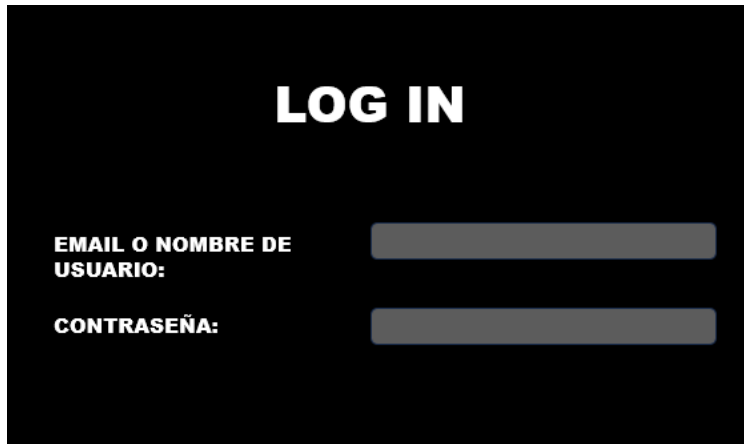
Caratula de presentación.



Nota: Presentación donde consta el nombre de la institución, el título del proyecto y los autores, Tomado de: Autoría propia (25 febrero 2025).

Figura 26

Ventana de ingreso a la plataforma web.



Nota: Se debe ingresar el usuario o email junto con la contraseña para poder acceder de manera correcta a la plataforma. Tomado de: Autoría propia (25 febrero 2025).

Figura 27

Ventana visualización de los datos de medición.



Nota: Se visualiza voltaje, amperaje, potencia y el valor estimado a pagar. Tomado de: Autoría propia (25 febrero 2025).

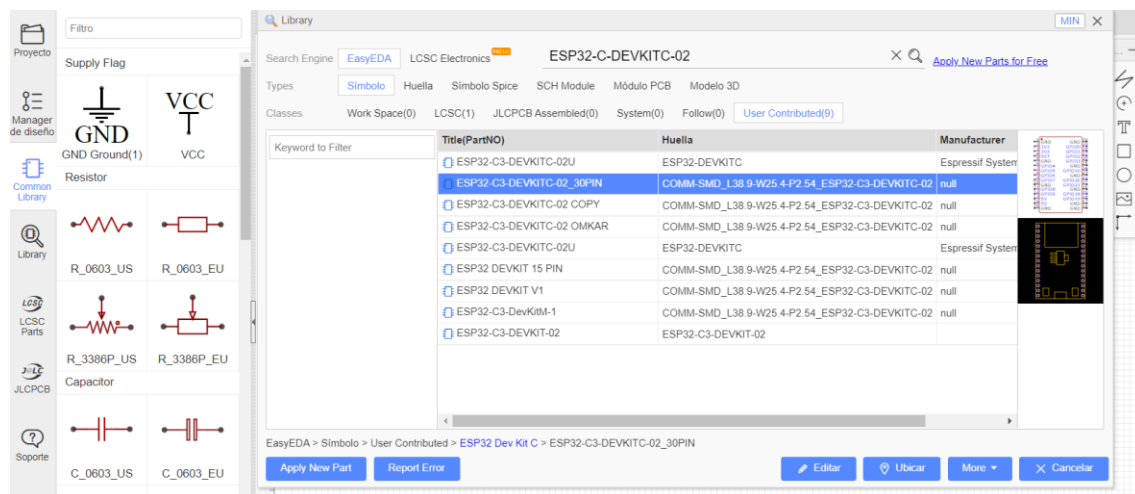
4.1.4. Diseño del circuito electrónico y PCB.

Para el desarrollo de esta etapa se usó el software EASYEDA que es una plataforma en línea para el diseño de circuitos electrónicos, que permite la creación de esquemas, simulaciones y diseño de PCB (placas de circuito impreso). Es utilizada por ingenieros, estudiantes y aficionados para desarrollar proyectos electrónicos de manera intuitiva y colaborativa. Su interfaz basada en la web facilita el acceso sin necesidad de instalación, permitiendo trabajar desde cualquier dispositivo con conexión a internet. EasyEDA ofrece una amplia biblioteca de componentes, integración con proveedores como LCSC y JLCPCB para fabricación, y herramientas de simulación para probar circuitos antes de fabricarlos. Para usarlo, solo se necesita crear una cuenta, diseñar el esquema, simular si es necesario, y luego generar el diseño PCB para su fabricación.

Una vez ingresamos y creamos un nuevo proyecto buscamos en la librería cada uno de los componentes necesarios para el diseño del circuito. (figura 28).

Figura 28

Selección de librerías de componentes.



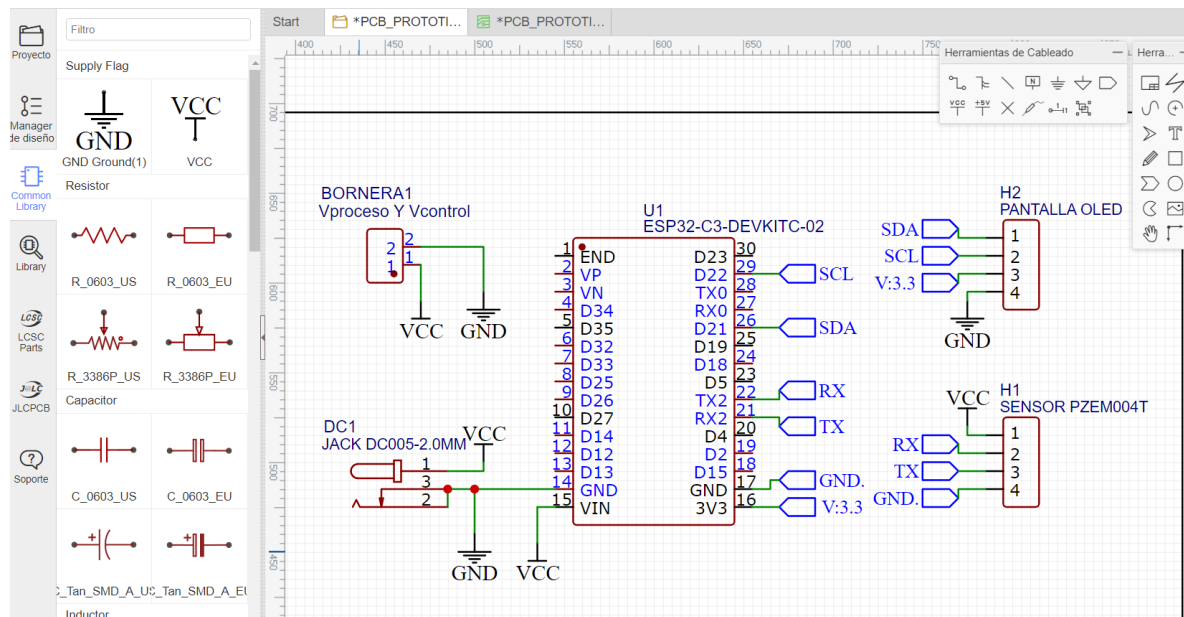
Nota: Los componentes a elegir pueden ser los disponibles en el software o los que usuarios crean y diseñan lo que facilita poder encontrar el elemento exacto. Tomado de: Autoría propia (23 febrero 2025).

A continuación, se realizó la selección y colocación de los componentes necesarios para el diseño del circuito, asegurando que cada conexión respetara la polaridad de las líneas de alimentación y que los datos fueran correctamente direccionados a los pines correspondientes. Para optimizar la organización y evitar una visualización confusa con múltiples líneas cruzadas,

se emplearon flags y etiquetas con nomenclaturas adecuadas, garantizando la correcta interconexión de los elementos. En la Figura 29, se puede observar la representación del esquema con estos ajustes.

Figura 29

Diseño del esquemático.

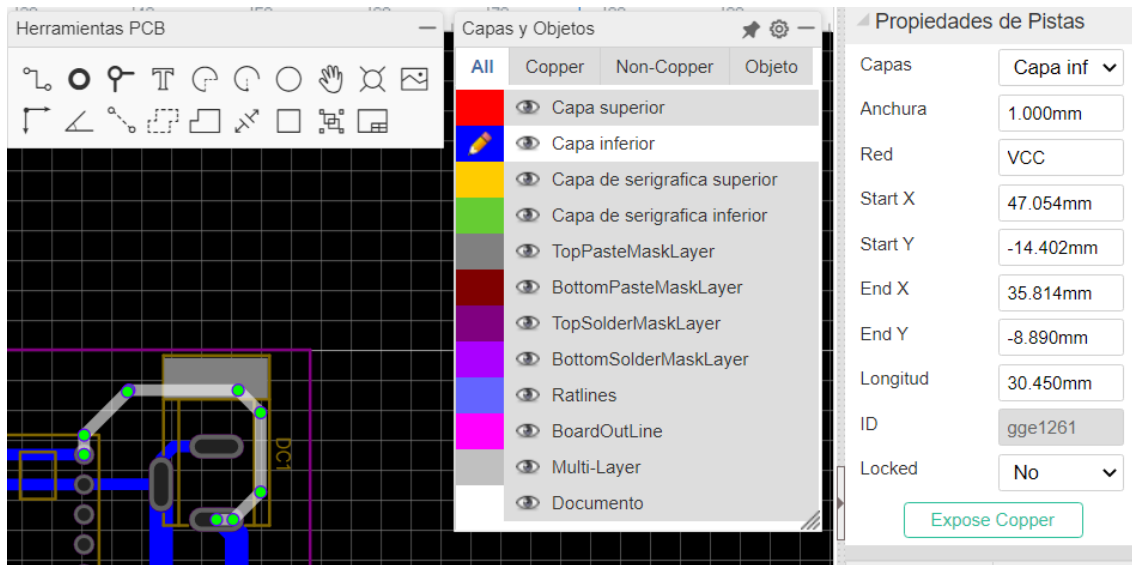


Nota: El circuito esta completo se ven las conexiones entre el ESP32, el módulo de medición, la pantalla oled (lado derecho) y las conexiones de alimentación DC (lado izquierdo). Tomado de: Autoría propia (12 febrero 2025).

Con el diseño esquemático finalizado, se procedió a la creación de la PCB, ubicando estratégicamente los componentes para simplificar el enrutamiento de pistas y minimizar errores. Se definió un ancho de 2 mm para las pistas de alimentación y 1 mm para las conexiones de señal, asegurando una distribución eficiente y sin interferencias, como se muestra en la Figura 30. Posteriormente, se generó el ruteado final de la PCB, permitiendo la correcta interconexión de los elementos, lo cual se aprecia en la Figura 31.

Figura 30

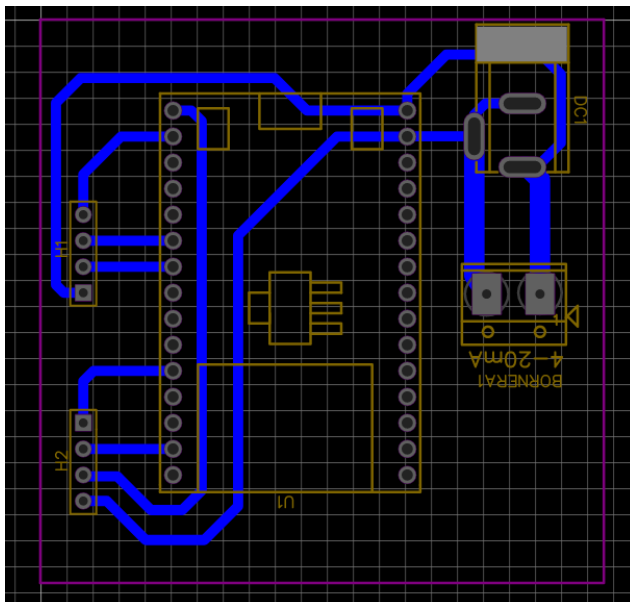
Pistas de conexión.



Nota: Se define el ancho de la pista para el tipo de valor que va a conducir, en este caso los datos que recibe del módulo de medición y los que transmite hacia la pantalla oled y para la alimentación de la ESP32. Tomado de: Autoría propia (14 febrero 2025).

Figura 31

Ruteo de PCB terminada.

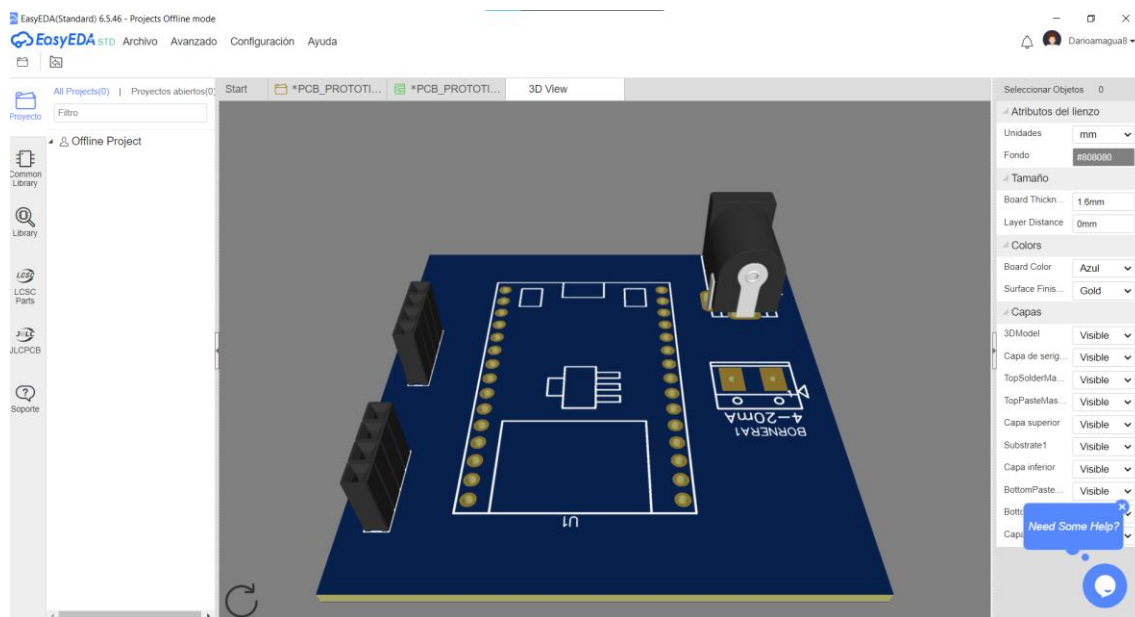


Nota: Se puede ver que el ruteo de las pistas se completó sin ningún error interconectando los componentes de manera adecuada. Tomado de: Autoría propia (14 febrero 2025).

Finalmente, el software permitió obtener una vista en 3D del diseño de la PCB, lo que facilitó la validación de la disposición de los componentes y la confirmación de que la estructura final cumplía con los requerimientos establecidos. En la Figura 32, se puede observar la visualización tridimensional del diseño, asegurando que la distribución y el tamaño de la placa fueran los adecuados para su fabricación.

Figura 32

Vista 3D.



Nota: Imagen 3D de la PCB terminada. Tomada de: Autoría propia (14 febrero 2025).

4.1.5. Diseño 3D de la carcasa protectora del prototipo.

Para esta etapa, procedemos al diseño 3D de la carcasa protectora del prototipo en el software FUSION 360, se toman en cuenta los siguientes pasos:

- Procedemos a la instalación del Software Fusion 360.
- Creamos una cuenta para albergar proyectos editables o de lectura.
- Creamos un nuevo diseño y seleccionamos un plano o cara a trabajar.
- Realizamos el boceto principal.

Boceto Principal

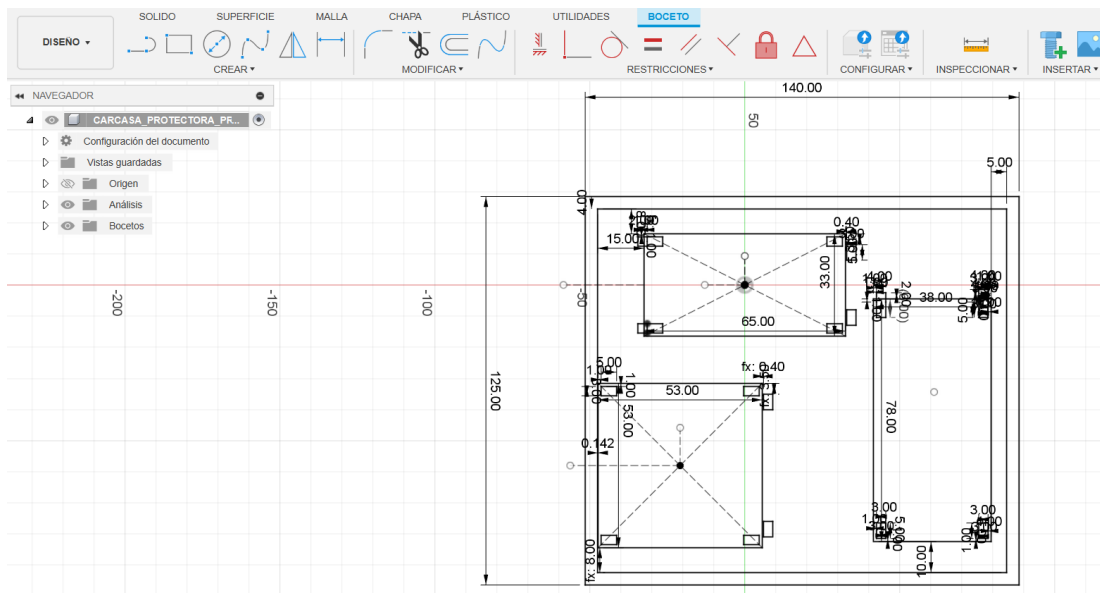
En el boceto principal, se definen las dimensiones de la carcasa protectora considerando el largo, ancho y altura adecuados para albergar los componentes electrónicos del prototipo. Se establecen las ubicaciones de cada componente, se determinan las distancias entre placas, la

disposición de los puertos de conexión y la integración de bases o soportes internos para fijar las placas electrónicas.

- Base estructural: Dimensiones de 140 mm x 125 mm con un espesor de 4 mm, proporcionando estabilidad.
- Componentes internos: Se han definido ubicaciones específicas para la fijación de placas electrónicas PCB de 53 x 53 mm, Fuente de 33x 65 mm, Sensor AC de 68 x 38 mm.
- Soportes de Fijación: Se incorporan 4 soportes en cada área de las tarjetas, estos varían de tamaño de 3 x 5 mm o 3 x 8 mm según el elemento.

Figura 33

Plano de Caja Software Fusion 360.



Nota: Boceto principal de la carcasa protectora del prototipo, mostrando la distribución de los componentes y soportes de fijación. Fuente: Elaboración propia.

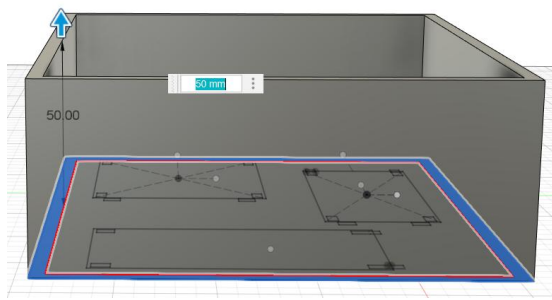
Modelado 3D de la carcasa protectora

Para el modelado 3D se utilizan diversas herramientas del software, como extrusión, corte, unión, empalme y chaflán. Además, se crean bocetos en distintas superficies para moldear los sujetadores, diseñar las ventilaciones, los puertos de alimentación AC- Tipo C, los orificios para cables y los soportes de fijación a la pared.

- Base estructural: Las dimensiones de ancho y largo de la caja se extruyeron a 50 mm desde la superficie inicial por otro lado, el área interior se extruyo 0.06 mm, finalmente las bases se extruyeron a 8 mm.

Figura 34

Extrusión de la base.

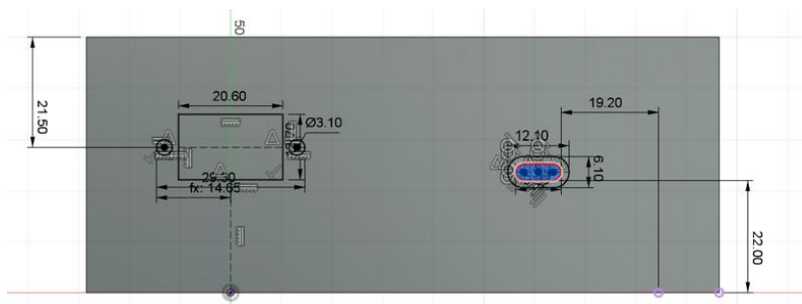


Nota: Extrusión del contorno inferior de la carcasa protectora a 50 mm de la base del boceto principal. Fuente: Elaboración propia.

- Bocetos superficiales: Para los orificios de puertos y conectores, se realizaron bocetos en cada superficie, determinando el tamaño de cada componente. Para el puerto AC se desde la superficie superior se tomó una distancia de 21.50 mm x 15.40 mm para realizar el primer círculo donde se ubica el tornillo, seguidamente con la herramienta de crear línea se realizó un rectángulo de 20.60mm x 12.70 mm. Luego, mediante la herramienta de corte, se generaron los espacios correspondientes con precisión.

Figura 35

Bocetos Superficiales Puerto AC y Tipo C.



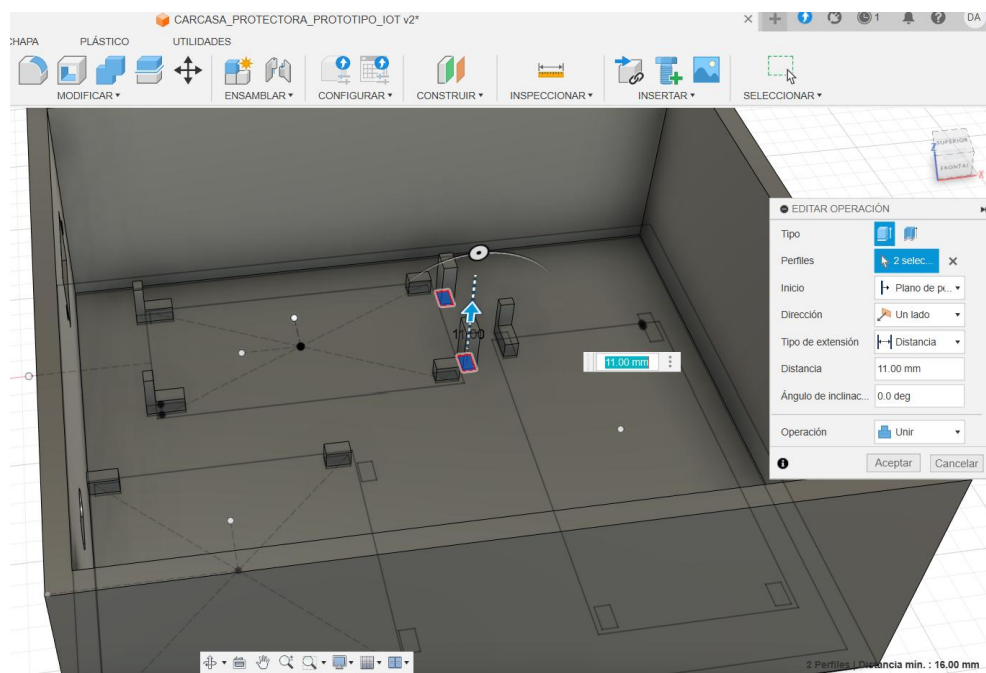
Nota: En la imagen se puede observar los bocetos realizados en la pared izquierda de la carcasa protectora se tomó medidas de cada componente para su creación. Elaboración propia.

- Sujetadores de fijación: Para la creación de los sujetadores se utilizaron bocetos basados en los pilares de soporte. Por ejemplo, en la fuente de alimentación, se extruyó 11 mm para la estructura base, luego se creó un rectángulo de 5 x 6 mm que se extruyó 2 mm.

Este procedimiento se repite para cada uno de los sujetadores que albergaran los componentes, módulos y circuitos electrónicos.

Figura 36

Sujetadores de Fijación.

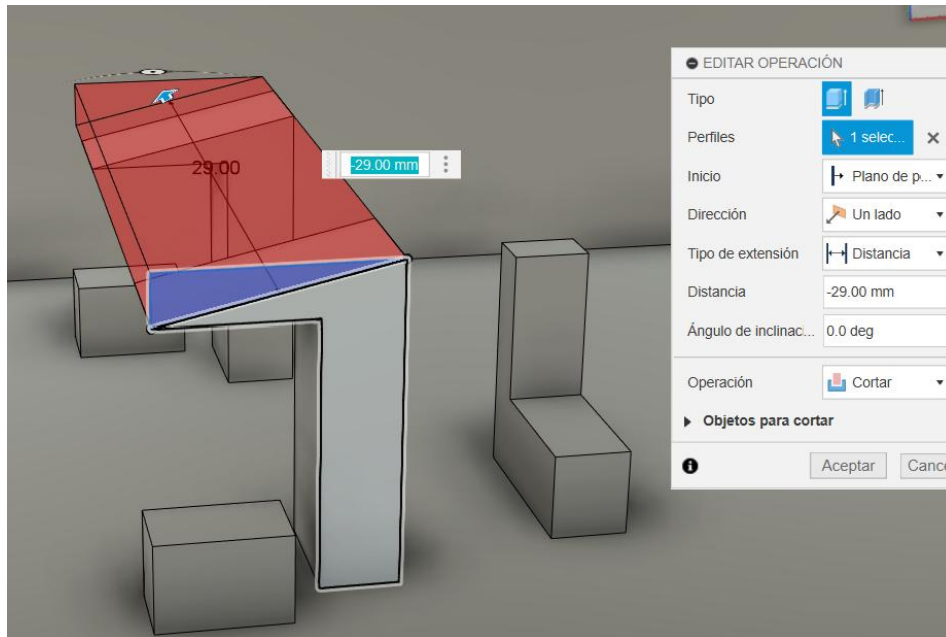


Nota: Se puede observar la extrusión de los pilares de la base de la fuente DC con la operación unir a una distancia de 11mm. Fuente Elaboración propia.

- Finalmente, se trazó una línea diagonal sobre uno de los rectángulos previamente creados al cual se le aplicó la operación de corte en una sola dirección a una distancia de -29 mm sin topar la pared de la carcasa protectora. Se cortan los dos sujetadores de la fuente ya que tienen las mismas dimensiones. Con esto se obtiene la forma de sujetador.

Figura 37

Corte de Sujetadores de fijación.



Nota: En esta imagen se puede observar el corte realizado en los 2 sujetadores de fijación de la fuente DC, se hace uso de la operación de corte a -29mm. Fuente: Elaboración propia.

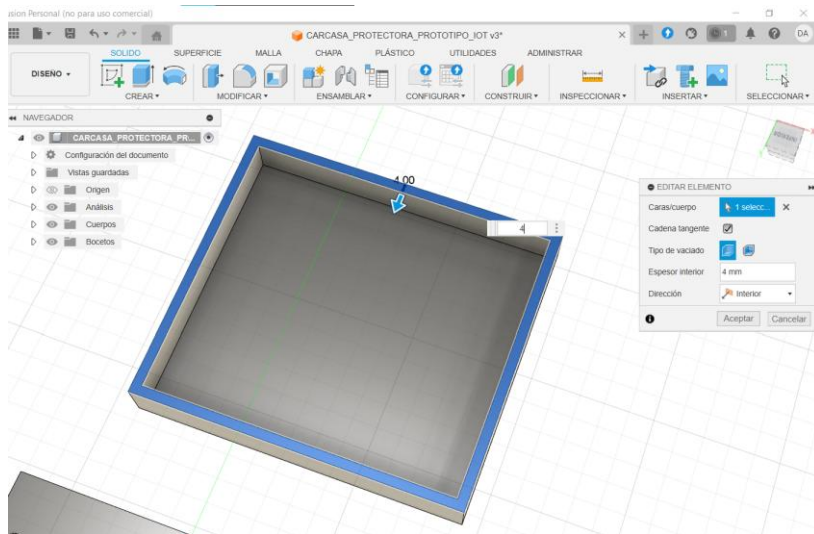
Diseño Finalizado

En la etapa final del diseño de la carcasa protectora se pulieron detalles con el uso de la herramienta de empalme la cual se aplicó al contorno de la base y 4 aristas a 1mm para suavizar la transición entre dos bordes. Con la herramienta de Chaflan, seleccionando la arista correspondiente se crearon los bordes a 2 mm para suavizar esquinas afiladas, mejorar la estética y funcionalidad de la pieza, o reducir esfuerzos mecánicos en ciertas áreas.

- Una vez terminada con la primera parte de la carcasa protectora se creó el boceto con la herramienta de rectángulo con las medidas de 125 x 140 mm, el cual se extruyó a 20 mm y se vació a 4 mm para obtener la forma de la tapa, de ahí se pulieron detalles como bordes, líneas de sujeción, el diseño del boceto de la pantalla oled.

Figura 38

Vaciado de Tapa.

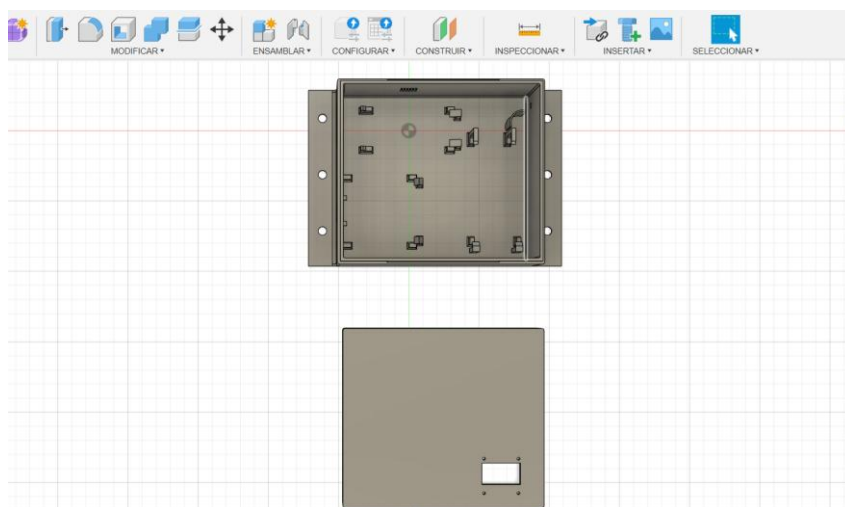


Nota: En esta imagen se puede observar la tapa de la carcasa protectora con el vaciado de 4 mm Fuente: Elaboración propia.

El diseño final de la carcasa protectora se ha modelado en Fusion 360, asegurando una estructura robusta y funcional para albergar los componentes electrónicos del prototipo, consideraron los siguientes aspectos clave:

Figura 39

Diseño Terminado.



Nota: En esta imagen podemos observar la vista superior de la carcasa protectora ya terminada, Fuente: Elaboración propia.

4.2. Implementación

4.2.1. Etapa1: Impresión 3D de la carcasa protectora del prototipo.

En esta etapa, se realiza la impresión 3D de la carcasa protectora utilizando material ABS, elegido por sus características de resistencia mecánica y térmica. Posteriormente, se lleva a cabo un proceso de postproducción para corregir imperfecciones.

Selección del material

Se optó por el ABS (Acrilonitrilo Butadieno Estireno) debido a su excelente resistencia mecánica y térmica, soportando temperaturas de hasta 100°C sin deformarse. Este material es ideal para aplicaciones electrónicas, ya que ofrece aislamiento eléctrico y resistencia a impactos, asegurando una mayor durabilidad del prototipo. A diferencia del PLA, el ABS es menos frágil y más adecuado para entornos donde la carcasa pueda estar expuesta a calor o esfuerzos mecánicos moderados.

Figura 40

Impresión de Caja 3D en material ABS.



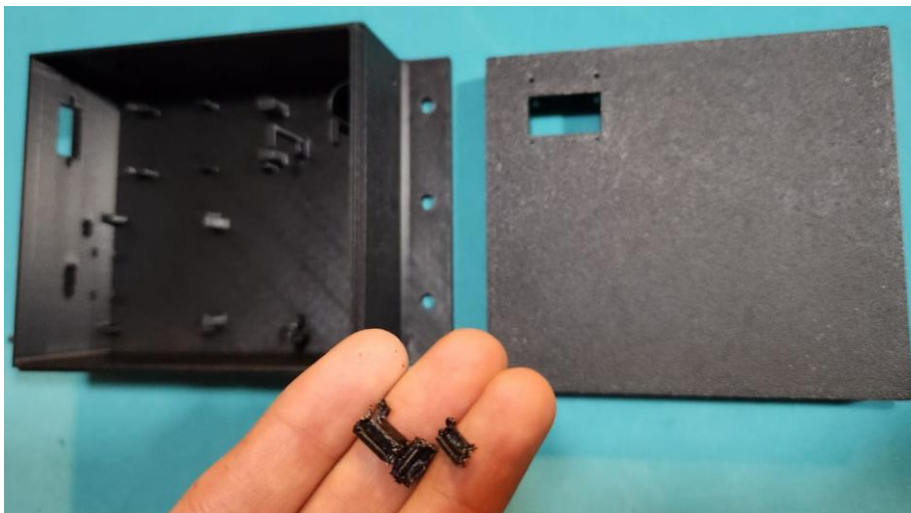
Nota: En la imagen se observa la carcasa protectora impresa en 3D con material ABS, se pueden distinguir los soportes internos diseñados para fijar los componentes electrónicos, así como los orificios de montaje y ventilación.

Impresión final

Una vez finalizada la impresión 3D de la carcasa, se procede a un proceso de postproducción para mejorar el acabado y garantizar el correcto ensamblaje de los componentes. Se realiza un lijado en las superficies que presenten imperfecciones o asperezas debido a la impresión en capas, se elimina las bases de soporte para los sujetadores, se revisa el estado de cada parte interna y se reconstruye en caso de ruptura o mala manipulación. Posteriormente, se montan los componentes electrónicos dentro de la carcasa para verificar su correcta fijación y alineación con los soportes internos.

Figura 41

Remoción y Limpieza de Soportes.



Nota: En la imagen se observa la carcasa protectora impresa en 3D con material ABS. Se pueden distinguir los elementos montados, como el PCB que alberga la ESP32, el módulo sensor PZEM-004T con su pinza amperimétrica y la tapa con la pantalla OLED instalada. Fuente: Elaboración propia.

4.2.2. Etapa2: Fabricación de la PCB y montaje de los elementos.

En esta etapa se detalla el proceso de fabricación y montaje de la PCB, desde la elaboración de la placa hasta su integración final con los componentes del prototipo, se abordan los procedimientos de soldadura, limpieza y verificación de las conexiones, garantizando la correcta funcionalidad del circuito.

Elaboración y ensamblaje de la PCB

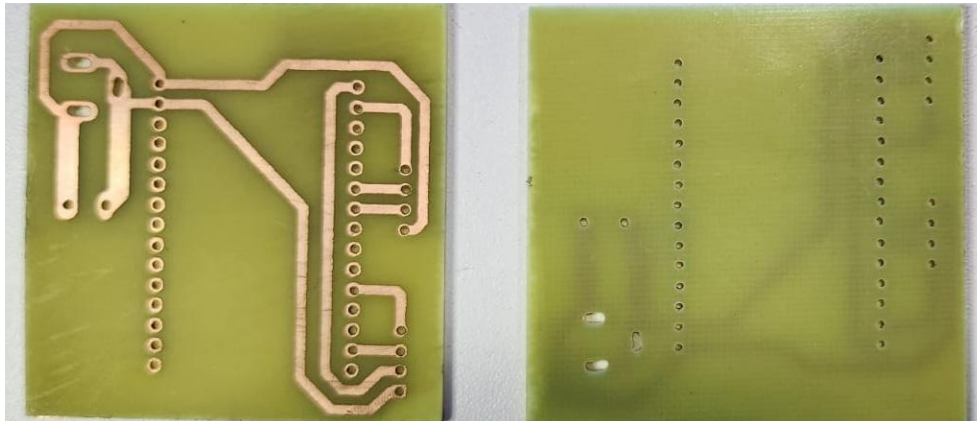
La PCB fue fabricada mediante el método de planchado por transferencia en una baquelita con recubrimiento de cobre de una sola cara. Una vez obtenida la baquelita después del proceso

químico y con sus pistas de cobre impresas en la baquelita, se realizaron las perforaciones necesarias para cada componente electrónico, asegurando una distribución adecuada.

Posteriormente, se llevó a cabo el proceso de soldadura con cautín y estaño para fijar cada componente como son las borneras, los jumpers hembra y macho, así como jack de alimentación DC y comunicación.

Figura 42

Fabricación de PCB.



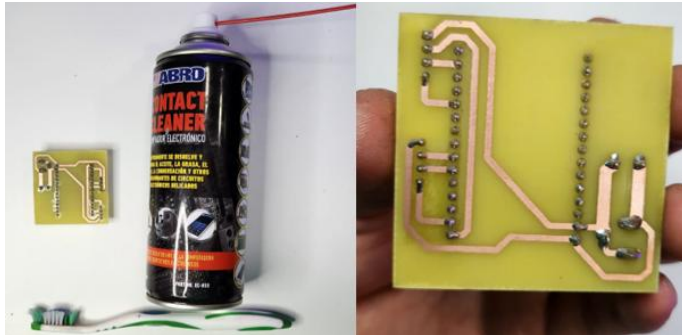
Nota: En la imagen se muestra la vista inferior de la PCB fabricada con las pistas de cobre y una vista superior con los orificios para los componentes. Fuente: Elaboración propia.

Limpieza y verificación de la PCB

En esta etapa se utilizó limpia contactos y un cepillo para eliminar los residuos de pasta de soldar y partículas de estaño presentes en la superficie de la PCB. Este proceso es fundamental para evitar cortocircuitos y garantizar una correcta conductividad en las conexiones. Finalmente, se llevó a cabo una verificación minuciosa de las soldaduras para descartar puentes de conexión no deseados o terminales flojos, asegurando así la funcionalidad y confiabilidad del circuito.

Figura 43

Limpieza de PCB.



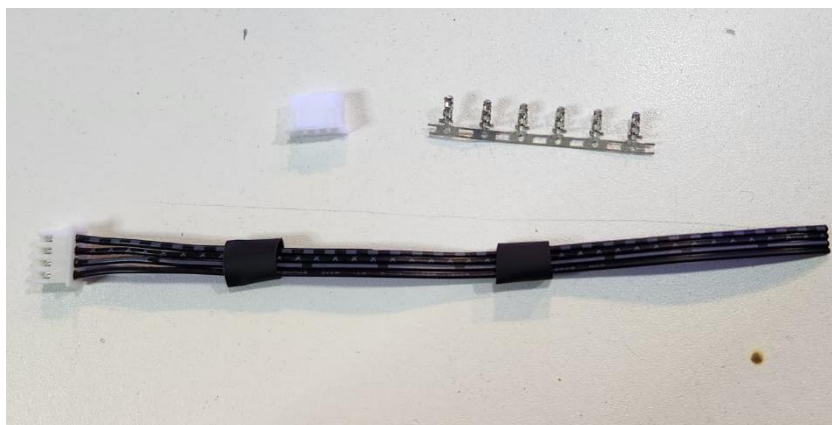
Nota: La imagen muestra los elementos utilizados en la limpieza de la PCB y como quedo la misma después del procedimiento. Fuente: Elaboración propia.

Cable de comunicación ESP32 y Sensor

Se fabricó el cable de comunicación UART para conectar el sensor Pzem-004T con la PCB, asegurando la correcta transmisión de datos. Dado que la comunicación UART requiere que los pines TX y RX estén invertidos entre dispositivos, se realizó la conexión adecuada utilizando conectores metálicos y una bornera de 4 pines. Cada terminal fue cuidadosamente soldado y ubicado según la posición correcta para garantizar el correcto funcionamiento del sistema.

Figura 44

Elaboración de Cable UART.



Nota: en la imagen se puede observar los componentes necesarios para fabricar el cable de comunicación UART entre el sensor y la placa con el ESP32. Fuente: Elaboración propia.

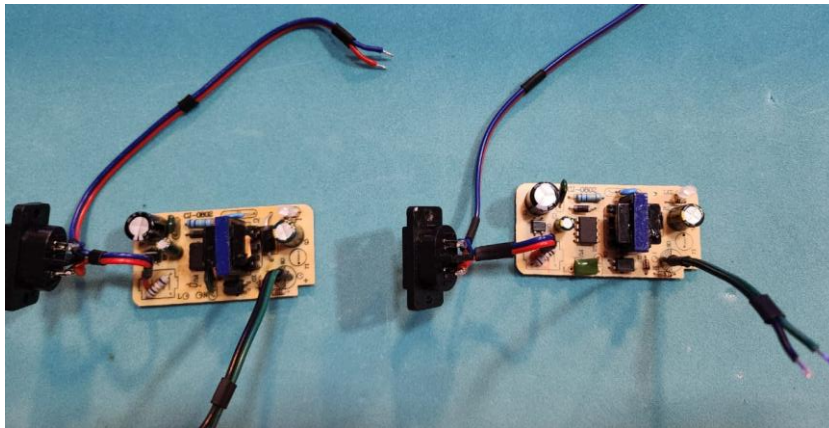
Alimentación de ESP32 y Sensor

La alimentación del sistema se realizó a través de un socket hembra para corriente alterna (AC) y una fuente de 5DC voltios. Se soldaron cables que permiten la conexión desde el conector AC de corriente alterna hacia la entrada de 120-240 AC de la fuente de 5 voltios, además de los cables de conexión de AC al sensor Pzem-004T.

Finalmente, para suministrar energía a la PCB, se tomaron los 5V de la fuente en paralelo con su capacitor de filtrado y en su fase se adiciono un diodo en serie asegurando una alimentación estable y adecuada para el correcto funcionamiento del circuito de la PCB y el microcontrolador.

Figura 45

Conexiones AC-DC de alimentación.



Nota: En la imagen se puede observar 2 de las 4 fuentes con sus respectivos cables de distribución de energía de AC y DC para los distintos componentes electrónicos. Fuente: Elaboración propia.

Integración de prototipo con Carcasa protectora

Una vez finalizados los cableados de los componentes del prototipo, se procede a la integración de todos ellos dentro de la carcasa protectora para ello se realizó lo siguiente:

- Fijación de la PCB, se fija la PCB en su base correspondiente, asegurándola con los sujetadores diseñados para mantenerla en su lugar de forma segura.
- Instalación de la fuente de alimentación, se instala la fuente de alimentación en su espacio designado dentro de la carcasa, utilizando los sujetadores incorporados. Se asegura el socket hembra para corriente alterna en su posición en la pared lateral

izquierda de la carcasa protectora. Luego, se realiza la conexión de los 5V DC de salida de la fuente de 5v hacia la bornera de la PCB.

- Posicionamiento del sensor Pzem-004T, el sensor Pzem-004T se coloca en su compartimento dentro de la carcasa, conectando su entrada AC proveniente del socket hembra de alimentación AC y posicionando la pinza amperimétrica en el lugar adecuado para la medición de corriente.
- Conexión de periféricos, se completan las conexiones de los periféricos, incluyendo el cable de comunicación UART entre la PCB y el sensor, así como el jumper hembra a hembra para la pantalla OLED.

Figura 46

Incorporación de componentes electrónicos y carcasa protectora.



Nota: En la imagen se puede observar el prototipo con todos sus periféricos integrados en la carcasa protectora. Fuente: Elaboración propia.

4.2.3. Etapa3: Desarrollo del código para ESP32.

Desarrollo de código para plataforma web IoT Blynk.

Como primer paso para desarrollar el código del sistema de monitoreo y gestión de consumo eléctrico en hogares urbanos, es fundamental diseñar un diagrama de flujo que represente las etapas del sistema de adquisición y transmisión de datos. Este diagrama servirá como una guía estructurada para la correcta implementación del código en el microcontrolador ESP32, asegurando que el proceso de lectura de sensores, procesamiento de datos y transmisión a la plataforma web o Blynk se realice de manera eficiente y organizada.

Desarrollo del diagrama de flujo del código del ESP32 y la plataforma BLYNK

Figura 47

Diagrama de Flujo código ESP32 a Blynk.



Nota: En este grafico podemos observar el esquema completo de programación para la validación del prototipo con la plataforma IoT Blynk, el cual se lo realizo en el software Lucidchart. Fuente Elaboración propia.

Instalación de Librerías y Configuración de Blynk

Para iniciar la programación del microcontrolador ESP32, se emplea el IDE de Arduino. Para ello, es fundamental instalar las librerías correspondientes tanto para el sensor Pzem-004T como para los periféricos integrados en el prototipo. Estas librerías facilitan la comunicación con los sensores, la pantalla OLED y la plataforma Blynk.

Al seleccionar la plataforma blynk se debe tener en cuenta los 3 parámetros necesarios para la comunicación como son

- Template ID: El cual es un identificador único asignado al proyecto dentro de Blynk, este define la estructura y los widgets que se usarán en la aplicación.
- Template Name: Es el nombre del proyecto o plantilla en la plataforma Blynk, utilizado para organizar y reconocer la configuración dentro del sistema.
- Auth Token: Es una clave de autenticación generada por Blynk, que permite la comunicación segura entre el microcontrolador y el servidor de Blynk. Este token debe ser copiado y utilizado en el código del microcontrolador para autorizar la conexión.

Figura 48

Configuración de Blynk y Librerías Arduino IDE.



```
1
2  /***** CONFIGURACIÓN DE BLYNK *****/
3  #define BLYNK_TEMPLATE_ID "TMPL276QjLQcR"
4  #define BLYNK_TEMPLATE_NAME "MONITORCASA1"
5  #define BLYNK_AUTH_TOKEN "2fZbLLhsSANKRB567x9uRzKmAxRxq7pA"
6
7  /***** INCLUSIÓN DE LIBRERÍAS *****/
8  #include <Wire.h>
9  #include <Adafruit_GFX.h>
10 #include <Adafruit_SSD1306.h>
11 #include <WiFi.h>
12 #include <BlynkSimpleEsp32.h>
13 #include <PZEM004Tv30.h>
14
```

Nota: En este grafico podemos observar las librerías utilizadas en Arduino además de los datos necesarios para la configuración de blynk. Fuente: Elaboración propia.

Definición de Pines y Configuración de Hardware

En esta fase, se definen las constantes esenciales para la configuración del sistema.

- Se establecen los parámetros de la pantalla OLED, especificando un tamaño de 128x64 píxeles y utilizando la librería Adafruit_SSD1306 para su control, con la conexión I2C gestionada a través del bus Wire y sin un pin de reinicio asignado (OLED_RESET = -1).
- Se configura el módulo PZEM004T v3.0 para la medición de parámetros eléctricos, asignando los pines GPIO16 (RX) y GPIO17 (TX) del ESP32 para la comunicación mediante Serial2. Además, se define la cantidad de módulos PZEM conectados al sistema, en este caso, uno solo (NUM_PZEMS = 1).
- Se establecen las credenciales de red WiFi, con un SSID y contraseña predefinidos, permitiendo la conexión del ESP32 para la transmisión y monitoreo de datos en la plataforma correspondiente.

Figura 49

Configuración de Constantes Oled, Sensor y Wifi.

```
15  /***** DEFINICIÓN DE CONSTANTES *****/
16  // Configuración de la pantalla OLED
17  #define SCREEN_WIDTH 128
18  #define SCREEN_HEIGHT 64
19  #define OLED_RESET -1
20  Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
21
22  /***** PZEM004T Configuración *****/
23  #define PZEM_RX_PIN 16
24  #define PZEM_TX_PIN 17
25  #define PZEM_SERIAL Serial2
26  #define NUM_PZEMS 1
27
28  PZEM004Tv30 pzem(PZEM_SERIAL, PZEM_RX_PIN, PZEM_TX_PIN);
29
30  // Datos de Wifi
31  const char* ssid = "dlink-E0EC";
32  const char* pass = "okxfq88706";
33
```

Nota: En este grafico podemos observar las líneas de código para la configuración del oled, el sensor Pzem-004T y las credenciales del Wifi. Fuente Elaboración propia.

Lectura de Sensores y Envío de Datos en el Loop

En esta fase, se ejecuta de forma continua la lectura de sensores y el envío de datos.

- Se mantiene activa la conexión con Blynk mediante la función Blynk.run(), asegurando la comunicación remota con la plataforma. Luego, se realiza la adquisición de datos del módulo PZEM004T, obteniendo los valores de voltaje (V),

corriente (A), potencia (W), energía acumulada (kWh), frecuencia (Hz) y factor de potencia (PF).

- Para garantizar la fiabilidad de los datos, se implementa una validación que detecta valores no numéricos (NaN). En caso de error en la lectura, se muestra un mensaje en el monitor serie y se omite el procesamiento de datos en ese ciclo.
- Los valores obtenidos se imprimen en el Monitor Serie, facilitando la supervisión en tiempo real. Los datos se organizan de manera estructurada con etiquetas claras para cada medición.
- Se actualiza la pantalla OLED, mostrando cada uno de los parámetros adquiridos con un tamaño de fuente estándar. Se establecen posiciones específicas para cada variable, asegurando una visualización ordenada y legible.
- Finalmente, los datos se envían a la plataforma Blynk mediante la escritura en pines virtuales (V1 - V6), permitiendo su monitoreo remoto. Se incorpora un retraso de 2 segundos antes de la siguiente lectura, optimizando el rendimiento del sistema y evitando actualizaciones innecesariamente rápidas.

Figura 50

Lectura de datos y Envío de Información.

```
102
103
104 /***** Mostrar datos de mediciones en OLED *****/
105 display.clearDisplay();
106 display.setTextSize(1); // Tamaño de fuente normal para las mediciones
107 display.setCursor(0, 0);
108 display.print("V: "); display.print(voltage); display.println("V");
109 display.setCursor(0, 10);
110 display.print("I: "); display.print(current); display.println("A");
111 display.setCursor(0, 20);
112 display.print("P: "); display.print(power); display.println("W");
113 display.setCursor(0, 30);
114 display.print("E: "); display.print(energy, 3); display.println("kWh");
115 display.setCursor(0, 40);
116 display.print("F: "); display.print(frequency, 1); display.println("Hz");
117 display.setCursor(0, 50);
118 display.print("PF: "); display.println(pf);
119
120 display.display();
121
122 /***** Enviar datos a Blynk *****/
123 Blynk.virtualWrite(V1, voltage);
124 Blynk.virtualWrite(V2, current);
125 Blynk.virtualWrite(V3, power);
126 Blynk.virtualWrite(V4, energy);
127 Blynk.virtualWrite(V5, frequency);
128 Blynk.virtualWrite(V6, pf);
129
130 delay(2000); // Espera entre lecturas
131
```

Nota: En este grafico podemos observar las líneas de código para visualizar los datos en el oled y su envío a plataforma Blynk. Fuente Elaboración propia.

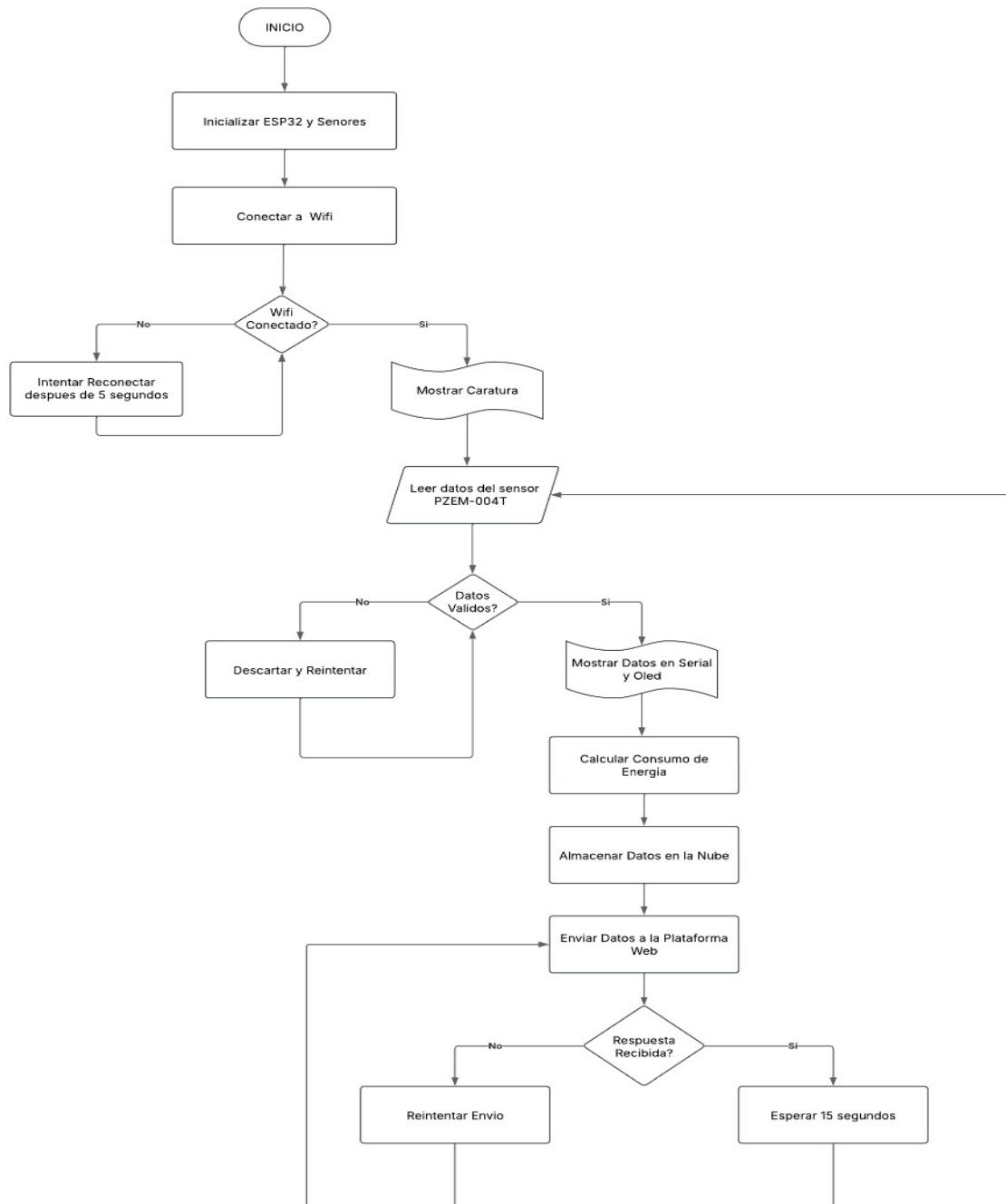
Desarrollo de código para plataforma web IoT implementada

Para desarrollar el código del sistema de monitoreo y gestión de consumo eléctrico en hogares urbanos, es fundamental diseñar un diagrama de flujo que represente las etapas del sistema de adquisición y transmisión de datos. Este diagrama servirá como una guía estructurada para la correcta implementación del código en el microcontrolador ESP32, asegurando que el proceso de lectura de sensores, procesamiento de datos y transmisión a la plataforma web IoT implementada se realice de manera eficiente y organizada.

Desarrollo del diagrama de flujo del código del ESP32 y la plataforma web IoT creada

Figura 51

Diagrama de Flujo ESP32 y Pagina IoT implementada.



Nota: Fuente Elaboración propia.

Instalación de Librerías y Configuración del Entorno

En esta sección, se lleva a cabo la instalación de las librerías y la configuración inicial del entorno para el funcionamiento del sistema. Se incluyen las librerías necesarias para la comunicación y control de los dispositivos conectados:

- **Arduino.h:** Proporciona las funciones esenciales para el desarrollo en plataformas basadas en Arduino.
- **WiFi.h:** Facilita la conexión del ESP32 a redes WiFi.
- **PubSubClient.h:** Permite la comunicación mediante el protocolo MQTT.
- **PZEM004Tv30.h:** Permite la interacción con el medidor de energía PZEM-004T.
- **Wire.h:** Habilita la comunicación I2C, esencial para dispositivos como la pantalla OLED.
- **Adafruit_GFX.h y Adafruit_SSD1306.h:** Facilitan el control gráfico del display OLED.

Figura 52

Librerías Arduino IDE.



```
Codigo_Esp32_MotPowMqtt_Oled.ino
1  | /*****
2  | //***  LIBRERÍAS  ***
3  | //****
4  | #include <Arduino.h>
5  | #include <WiFi.h>
6  | #include <PubSubClient.h>
7  | #include <PZEM004Tv30.h>
8  | #include <Wire.h>
9  | #include <Adafruit_GFX.h>
10 | #include <Adafruit_SSD1306.h>
11 |
```

Nota: En este gráfico podemos observar las librerías necesarias para cada función del prototipo IoT. Fuente Elaboración propia.

Configuración del Cliente MQTT

En esta sección, se define la configuración del protocolo MQTT para la transmisión y recepción de datos en la red. Se establecen los parámetros de conexión al servidor MQTT, como son los siguientes:

- mqtt_server: Dirección del servidor MQTT, este es el encargado de gestionar la comunicación entre los dispositivos.
- mqtt_port: Puerto de conexión al servidor, definido en 1883 el cual se utiliza para una comunicación estándar sin cifrado.
- mqtt_user y mqtt_pass: Credenciales de acceso al servidor para la autenticación del ESP32.
- Topic_Out y Topic_In: Se definen los tópicos de comunicación, donde Topic_Out enviará los datos de monitoreo y Topic_In recibirá comandos o configuraciones desde el servidor.

Para el correcto funcionamiento del sistema, se declaran variables destinadas a almacenar las mediciones obtenidas del sensor PZEM-004T, incluyendo voltaje (V), corriente (A), potencia (W), energía acumulada (kWh), frecuencia (Hz) y factor de potencia (pf). Además, se inicializa un cliente WiFi (espClient) y un objeto client basado en la biblioteca PubSubClient, el cual permitirá la comunicación con el broker MQTT para la transmisión y recepción de datos.

Asimismo, se definen funciones clave para la operatividad del sistema. La función setup_wifi gestiona la conexión del ESP32 a la red inalámbrica, asegurando la comunicación estable. La función callback(char* topic, byte* payload, unsigned int length) procesa los mensajes recibidos desde el servidor MQTT, permitiendo la respuesta a eventos remotos. Por otro lado, reconnect se encarga de restablecer la conexión con el broker en caso de desconexión.

Figura 53

Configuración de Cliente MQTT.

```

Codigo_Esp32_MotPowMqtt_Oled.ino
33  //*****
34  //***  CONFIGURACIÓN MQTT  ***
35  //*****
36  const char *mqtt_server = "monitor-casas.com";
37  const int mqtt_port = 1883;
38  const char *mqtt_user = "my_user_mqtt_casas";
39  const char *mqtt_pass = "rasp_casas_mqtt_2024";
40  const String Topic_Out = "Monitoreo/Casa_1/Out"; // Topic de salida de Casa Monitoreada
41  const String Topic_In = "Monitoreo/Casa_1/In"; // Topic de entrada de Casa Monitoreada

```

Nota: En este grafico podemos observar las líneas de código para la configuración de protocolo MQTT. Fuente Elaboración propia.

Configuración Inicial del Sistema del Setup y Bucle principal Loop

La configuración inicial del sistema se lleva a cabo en la función Setup, la cual se ejecuta una sola vez al iniciar el ESP32. En este proceso, se establece la comunicación serial a 115200 baudios para la depuración mediante el Monitor Serie, y se invoca la función `setup_wifi` para conectar el ESP32 a la red inalámbrica. Además, se configura el servidor MQTT con el puerto 1883 y se define la función callback para gestionar los mensajes recibidos. Paralelamente, se inicializa la pantalla OLED con la dirección I2C 0x3C, verificando su correcto funcionamiento. Como mensaje de inicio, se limpia la pantalla, se establece el tamaño del texto en 2 con color blanco y se muestra "MEDIDOR ELÉCTRICO" durante 500 ms, garantizando que el sistema esté correctamente configurado antes de entrar en operación.

Figura 54

Configuración Inicial de Void Setup.

```
61 void setup()
62 {
63     Serial.begin(115200);
64     setup_wifi();
65     client.setServer(mqtt_server, mqtt_port);
66     client.setCallback(callback);
67
68     if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
69         Serial.println(F("SSD1306 allocation failed"));
70         for(;;);
71     }
72     delay(500);
73     display.clearDisplay();
74     display.setTextSize(2);
75     display.setTextColor(WHITE);
76     display.setCursor(0, 16); display.println(" MEDIDOR ");
77     display.setCursor(0, 32); display.println(" ELECTRICO ");
78     display.display();
79     delay(500);
80 }
```

Nota: En este gráfico se presenta el código correspondiente a la configuración de comunicación WiFi, servidor MQTT y la inicialización de la pantalla oled. Fuente Elaboración propia.

El bucle principal loop se encarga de ejecutar continuamente el monitoreo del consumo eléctrico, actualizar la pantalla OLED y transmitir los datos mediante MQTT. Para ello, verifica y restablece la conexión MQTT en caso de desconexión, asegurando la comunicación estable con el servidor. La lectura de sensores se realiza cada 15 segundos, utilizando millis para determinar el tiempo transcurrido desde el inicio del ESP32. Se obtienen los datos del sensor

PZEM-004T, incluyendo voltaje, corriente, potencia, energía acumulada, frecuencia y factor de potencia, validando cualquier error en las mediciones. Los valores recopilados se imprimen en el Monitor Serie para depuración y se reflejan en la pantalla OLED tras limpiarla. Finalmente, los datos se concatenan en un solo mensaje y se envían al tópico MQTT Monitoreo/Casa_1/Out, permitiendo la supervisión remota del consumo eléctrico en tiempo real.

Figura 55

Configuración Inicial de Void Loop.

```
85 void loop() {
86     if (!client.connected()) {
87         reconnect();
88     }
89     client.loop();
90     long now = millis();
91     if (now - lastMsg > 15000){
92         lastMsg = now;
93         // Leer datos del sensor
94         voltage = sen_pzem.voltage();
95         current = sen_pzem.current();
96         power = sen_pzem.power();
97         energy = sen_pzem.energy();
98         frequency = sen_pzem.frequency();
99         pf = sen_pzem.pf();
100
101         // Validar datos
102         if(isnan(voltage)) { voltage = 0; }
103         if(isnan(current)) { current = 0; }
104         if(isnan(power)) { power = 0; }
105         if(isnan(energy)) { energy = 0; }
106         if(isnan(frequency)) { frequency = 0; }
107         if(isnan(pf)) { pf = 0; }
```

Nota: En este gráfico se observan las líneas de código encargadas del monitoreo de sensores, actualización de la pantalla y envío de datos a través de MQTT. Fuente: Elaboración propia

Módulos de Conexión WiFi y MQTT

En esta sección del código, se configura la conexión WiFi y la gestión de mensajes MQTT. La conexión a la red se establece mediante `WiFi.begin(ssid, password);`, manteniendo un bucle de espera hasta que el ESP32 obtiene una dirección IP (`WL_CONNECTED`). Una vez conectado, la IP asignada se muestra en el Monitor Serie para verificación. Seguidamente la función de callback de MQTT se ejecuta automáticamente cuando el dispositivo recibe un mensaje en un tópico suscrito. En esta función, el payload se convierte en un string (incoming) y se imprime en la consola junto con el tópico de origen.

Para manejar la reconexión automática al servidor MQTT, en el código se implementa un bucle (while (client.connected)) que verifica el estado de la conexión. Si el ESP32 pierde la conexión, se genera un ID aleatorio y se intenta reconectar al servidor. En caso de éxito, se suscribe nuevamente al tópico de entrada (Topic_In), mientras que, si falla, se imprime el código de error y se espera cinco segundos antes de reintentar.

Figura 56

Configuración de Conexión WiFi y Comunicación MQTT.

```
156 //*****  
157 /*** CALLBACK MQTT ***/  
158 //*****  
159 void callback(char* topic, byte* payload, unsigned int length)  
160 {  
161     String incoming = "";  
162     Serial.print("Mensaje recibido desde -> "); Serial.print(topic);  
163     Serial.println("");  
164     for (int i = 0; i < length; i++) {  
165         incoming += (char)payload[i];  
166     }  
167     incoming.trim();  
168     Serial.println("Mensaje -> " + incoming);  
169 }  
170  
171 //*****  
172 /*** RECONEXIÓN MQTT ***/  
173 //*****  
174 void reconnect()  
175 {  
176     while (!client.connected()) {  
177         Serial.print("Intentando conexión Mqtt...");  
178         String clientId = "esp32_";  
179         clientId += String(random(0xffff), HEX);  
180         if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {  
181             Serial.println("Conectado!");  
182             Serial.print("Nos suscribimos a: "); Serial.println(Topic_Out);  
183             client.subscribe(Topic_In.c_str());  
184         } else {  
185             Serial.print("falló :( con error -> ");  
186             Serial.print(client.state());  
187             Serial.println(" Intentamos de nuevo en 5 segundos");  
188             delay(5000);  
189         }  
190     }  
191 }
```

Nota: Se muestra las líneas de código del procedimiento de establecer la conexión WiFi, gestionar los mensajes MQTT y ejecutar la reconexión automática. Fuente: Elaboración propia.

4.2.4. Etapa4: Desarrollo e implementación de Plataforma web IoT

- **Plataforma IoT Blynk**

Para la validación del prototipo, se creó una cuenta en la plataforma IoT Blynk con el correo proyectoiot2025@gmail.com y la contraseña proyecto*****. A continuación, se generó un template o plantilla, un paso esencial para estructurar el dispositivo IoT y definir su funcionalidad dentro de la plataforma, se asignó un nombre y se selecciona en hardware que es la ESP32. Durante este proceso, se obtuvo el ID del template, que sirve como identificador

único, y se le asignó un nombre acorde al proyecto. Este template permitirá la interacción entre el hardware y la plataforma, facilitando la gestión de datos en tiempo real.

Figura 57

Creación de Cuenta IoT Blynk.

The image shows two side-by-side forms from the Blynk platform. On the left is the 'Sign Up' form, which includes a Blynk logo, a welcome message, an email input field (containing 'proyectoiot2025@gmail.com'), a checkbox for terms and conditions, and a 'Sign Up' button. On the right is the 'Create New Template' form, which includes fields for 'NAME' (containing 'MONITORCASA1'), 'HARDWARE' (a dropdown menu showing 'ESP32'), 'CONNECTION TYPE' (a dropdown menu showing 'WIFI'), and a 'DESCRIPTION' text area (containing 'MONITOREO CASA 1 SENSOR Y ESP32'). Both forms have 'Cancel' and 'Done' buttons at the bottom right.

```
#define BLYNK_TEMPLATE_ID
"TMPL276QjLQcR"
#define BLYNK_TEMPLATE_NAME
"MONITORCASA1"
```

Nota: La imagen muestra la creación de la página IoT blynk, el template y la obtención del template ID y template Name necesarios para el código. Fuente: Elaboración propia.

Posteriormente, se configuraron pines virtuales, que actúan como canales de comunicación entre el dispositivo físico y la plataforma Blynk. A cada pin se le asignó un nombre, un número específico, el tipo de dato a manejar (por ejemplo, flotante o entero) y un rango de valores según la medición esperada. Se crearon seis pines virtuales, desde V1 hasta V6, destinados a medir voltaje, corriente, potencia, consumo energético, frecuencia y factor de potencia, asegurando que cada parámetro relevante del sistema sea monitoreado correctamente. Luego, se diseñó el dashboard, donde cada widget visual se vinculó a un pin virtual, permitiendo representar los datos de forma clara y amigable para el usuario.

Figura 58

Configuración de Pines Virtuales.

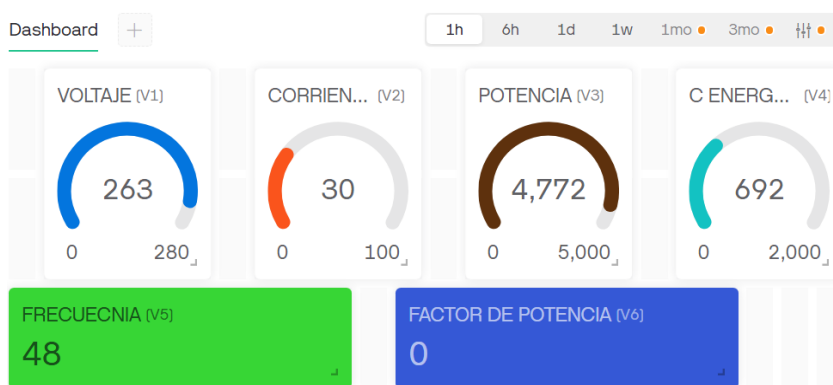
Id	Name	Pin	Color	Data Type	Units	Is Raw
2	CORRIENTE	V2	Blue	Double		false
3	POTENCIA	V3	Red	Double		false
4	C ENERGETICO	V4	Brown	Double		false
5	FRECUENCIA	V5	Cyan	Integer		false
6	FACTOR DE POTENCIA	V6	Green	Double		false

Nota: La imagen muestra la creación de los pines virtuales asignándoles un numero de pin, un rango de ser double y un título. Fuente: Elaboración propia.

Finalmente, en la web dashboard, se seleccionaron los widgets adecuados para cada tipo de medición, como gráficos de líneas, medidores de aguja o indicadores numéricos. Cada widget se configuró con su respectivo pin virtual y se establecieron parámetros adicionales como el rango de valores permitidos, el título y la unidad de medida (voltios, amperios, etc.). Esta configuración asegura que los datos sean presentados de manera organizada y comprensible. Con esto, se completó la configuración del dispositivo, obteniendo el auth token, un código único necesario para integrar el hardware con la plataforma y permitir el envío de datos en tiempo real desde el prototipo a la nube.

Figura 59

Web Dashboard configurada



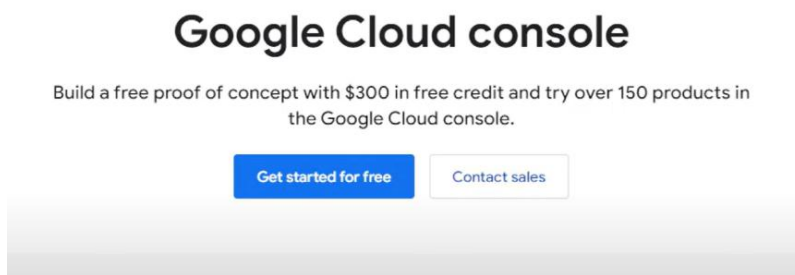
Nota: La imagen muestra la dashboard con los widgets de mediciones: Elaboración propia.

- **Google Cloud y VPS**

Para la implementación de la plataforma web IoT, se utilizó **Google Cloud** como proveedor de servicios en la nube, permitiendo la creación de una máquina virtual (**VPS**) para alojar la aplicación y gestionar las conexiones de los dispositivos.

Figura 60

Inicio de configuración GOOGLE CLOUD.



Nota: La plataforma permite hacer uso de \$300 de crédito para los proyectos de manera gratuita durante 3 meses. Tomado de: Autoría propia (20 febrero 2025).

- Primero, se configuró una cuenta en Google Cloud y se establecieron los parámetros de facturación (Figuras 58-59), lo que permitió acceder a los servicios de **Compute Engine**. Se creó una **instancia de máquina virtual (VM)** (Figuras 60-61) basada en **Ubuntu 20.04 LTS** (Figuras 62-63), seleccionando una configuración óptima de CPU y almacenamiento en disco **SSD persistente** (Figuras 64-65). Además, se activaron los permisos necesarios para tráfico **HTTP y HTTPS** en el firewall, garantizando el acceso a la plataforma.(Figura 66).

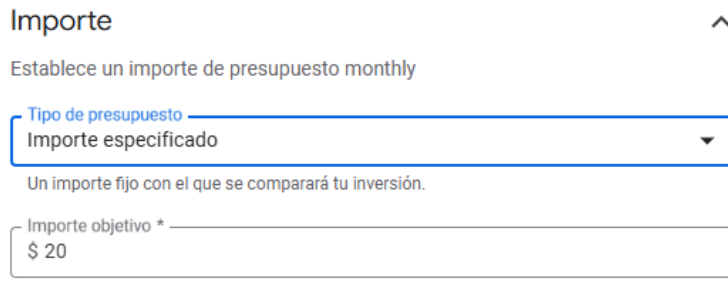
Figura 61

Creación de nuevo proyecto.

Nota: Se debe dar un nombre y presionar botón crear. Tomado de: Autoría propia (20 febrero 2025).

Figura 62

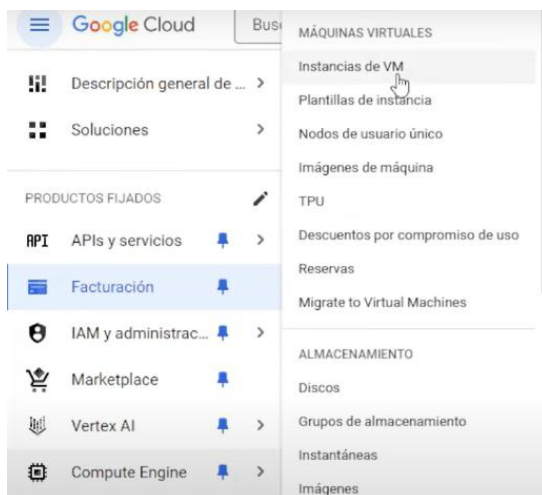
Definir valor para pagar mensualmente.



Nota: En esta sección se determina el valor estimado que podremos o queremos pagar mes a mes, en nuestro caso \$20. Tomado de: Autoría propia.(20 febrero 2025).

Figura 63

Crear un VPS.



Nota: Clic en las 3 barras de menú, seleccionamos “Compute Engine” y luego “Instancias VM”. Tomado de: Autoría propia (20 febrero 025).

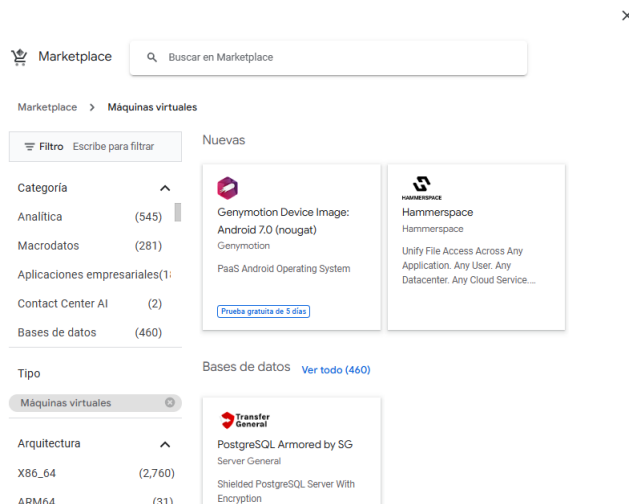
Figura 64

Habilitar la API.



Figura 65

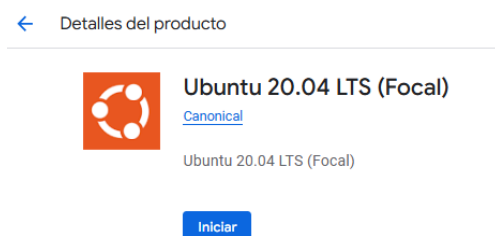
Ingresar a Marketplace.



Nota: Dentro de esta sección podemos elegir el SO necesario para nuestro VPS. Tomado de: Autoría propia.(20 febrero 2025)

Figura 66

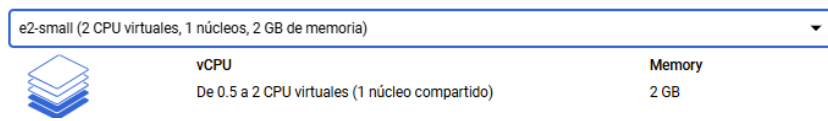
Elección sistema operativo.



Nota: En la ventana de búsqueda Escribimos Ubuntu y seleccionamos la versión que nos servirá para el proyecto. Tomado de: Autoría propia.(20 febrero 2025).

Figura 67

Elegir CPU



Nota: Seleccionamos esta opción de CPU, que será suficiente para el proyecto y permitirá hacer las instalaciones necesarias de manera rápida. Tomado de: Autoría propia.(20 febrero 2025).

Figura 68

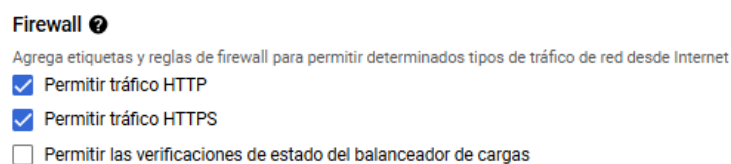
Elegir Disco.



Nota: Modificamos el disco de arranque, clic en cambiar y seleccionamos SSD persistente. Tomado de: Autoría propia(20 febrero 2025).

Figura 69

Firewall

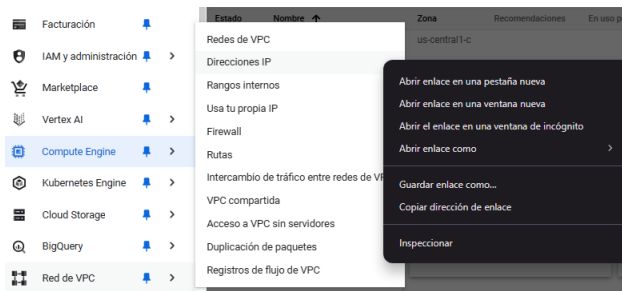


Nota: Activamos los permisos para tráfico HTTP y HTTPS y clic en crear. Tomado de: Autoría propia(20 febrero 2025).

- Para asegurar la conectividad estable con un dominio personalizado, se configuró una **dirección IP estática**, vinculándola a la instancia creada. Este procedimiento fue fundamental para la correcta integración del servidor con el sistema de nombres de dominio (DNS).

Figura 70

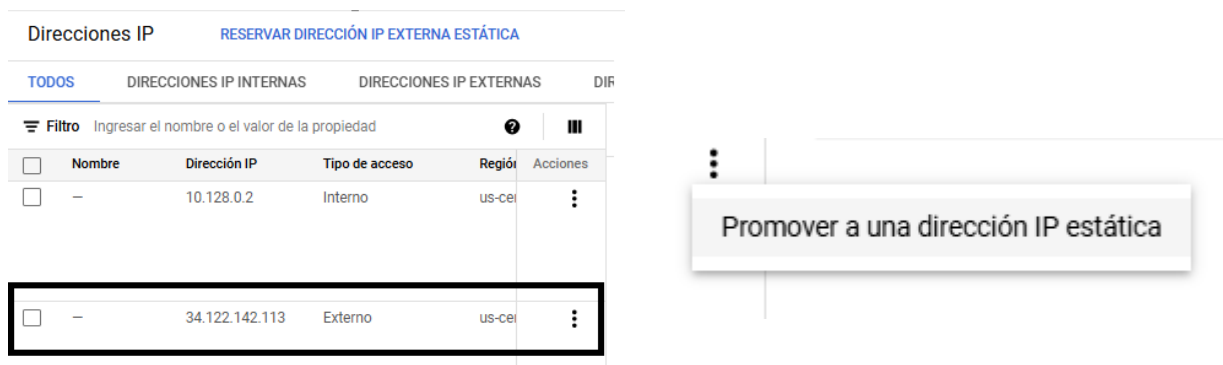
Configurar IP estática.



Nota: ponemos el cursor sobre RED DE VPC, clic derecho sobre DIRECCIONES IP y abrir en una nueva pestaña. Tomado de: Autoría propia.(20 febrero2025).

Figura 71

Ubicar IP externa



Nota: Damos clic en los 3 puntos y elegimos PROMOVER A IP ESTÁTICA. Tomado de: Autoría propia(20 febrero 2025).

Figura 72

Dar nombre.

Promover direcciones IP

Nombre *
vps-servicio-medidor

Se permiten letras minúsculas, números y guiones

Descripción
vps-servicio-medidor

CANCELAR RESERVAR

Filtro Ingresar el nombre o el valor de la propiedad

☐	Nombre	Dirección IP	Tipo de acceso	Región	Tipo ↓	Versión	En uso por
☐	vps-servicio-medidor	34.111.111.113	Externo	us-central1	Estática	IPv4	Instancia de VM vps-servicio-medidor (Zona us-central1-c)

Nota: En nombre y descripción colocamos los mismos datos del VPS, y clic en RESERVAR. Tomado de: Autoría propia. (20 febrero2025).

- **MobaXterm y Acceso al VPS**

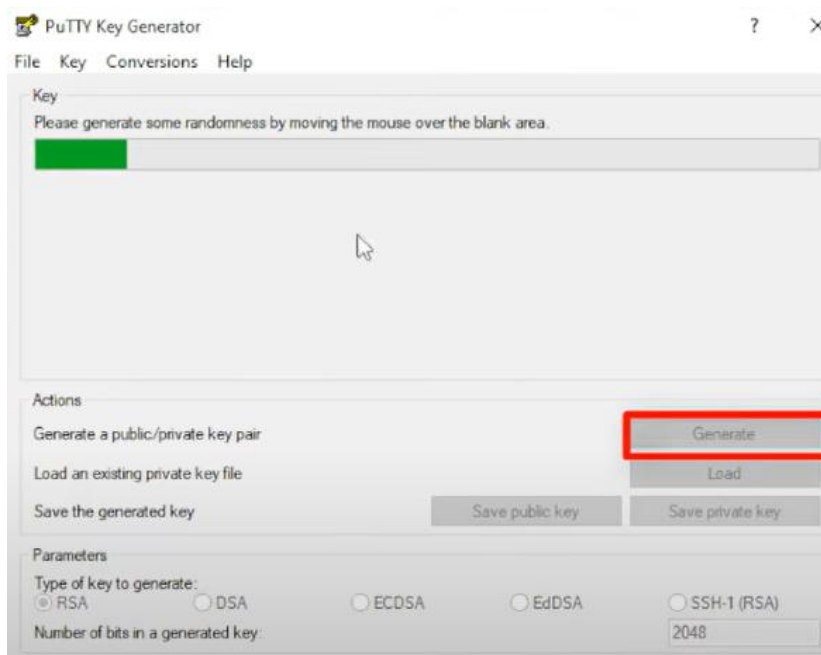
Para la administración del servidor, se optó por MobaXterm, una herramienta avanzada que facilita la gestión de sesiones SSH y la transferencia de archivos.

El proceso inició con la generación de claves SSH mediante PuTTY (Figura 70), creando un par de claves pública y privada que permitieron la autenticación segura entre la máquina local y el VPS. Posteriormente, la clave pública fue añadida a la configuración del servidor en Google Cloud (Figuras 71-72), permitiendo el acceso remoto mediante MobaXterm.(Figura 73).

Una vez establecida la conexión SSH con la IP externa del VPS, se procedió a la instalación de paquetes esenciales, incluyendo Apache como servidor web (Figura 74). Para verificar la correcta instalación, se accedió a la IP pública desde un navegador, confirmando que el servidor respondía correctamente a las solicitudes HTTP. (Figura 75).

Figura 73

Generación de Keys en PUTTY.



Nota: Clic en GENERAR y bajo la barra de progreso existe ese espacio en blanco hay que ir pasando el cursor del mouse para generar las KEYS. Tomado de: Autoría propia.(20 febrero 2025).

Figura 74

Copiar key.

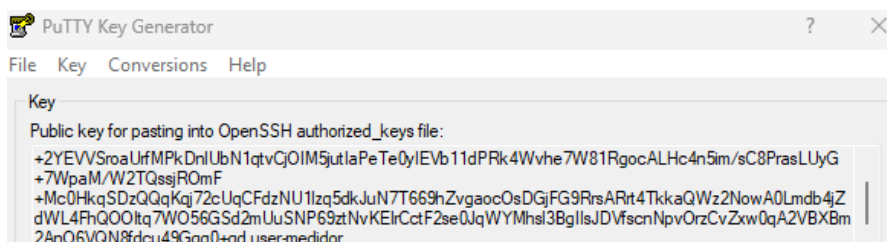
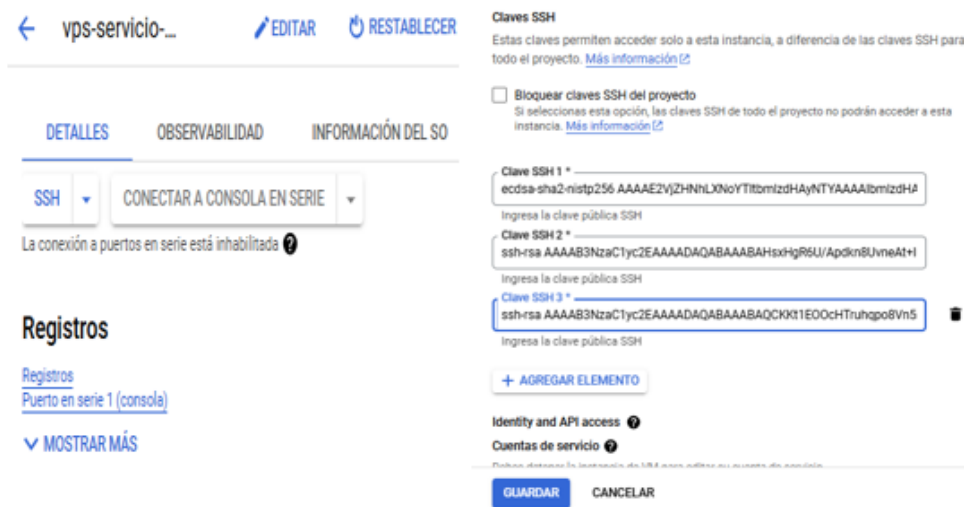


Figura 75

Agregar nueva clave en el servidor.



Nota: En el VPS, damos clic sobre el nombre, se abre y clic en EDITAR, aquí debemos buscar la sección CLAVES SSH, clic en AGREGAR ELEMENTO, pegamos la KEY que se generó en PUTTY y clic en GUARDAR.

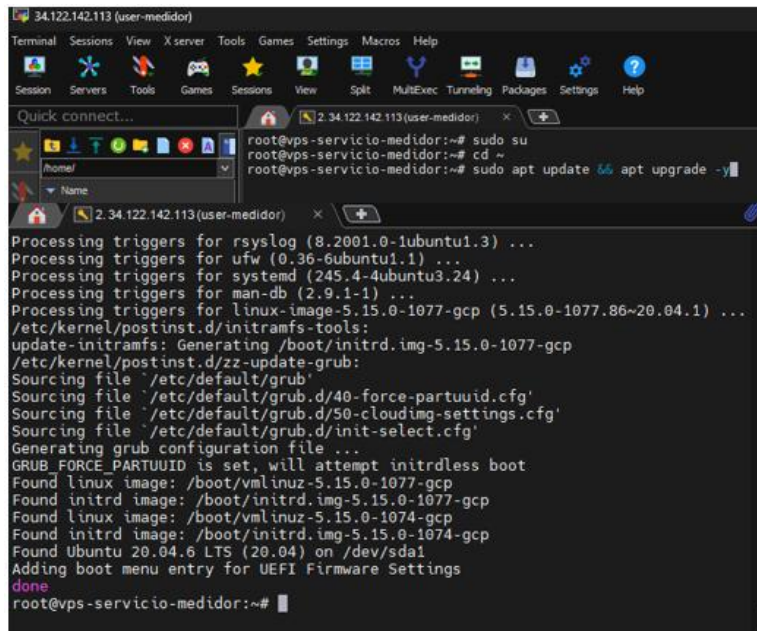
Figura 76

Iniciar sesión en MobaXterm.



Nota: Seleccionamos la opción SSH, se abre una ventana, donde debemos pegar la IP EXTERNA de nuestro VPS y el nombre de usuario que usamos en PUTTY. Clic en ADVANCED SSH habilitamos USE PRIVATE KEY y desde la carpeta de CREDENCIALES VPS cargamos la KEY PRIVADA la seleccionamos y clic en ABRIR, finalmente clic en OK. Tomado de: Autoría propia. (20 febrero 2025).

Figura 77
Instalar Ubuntu y Apache.

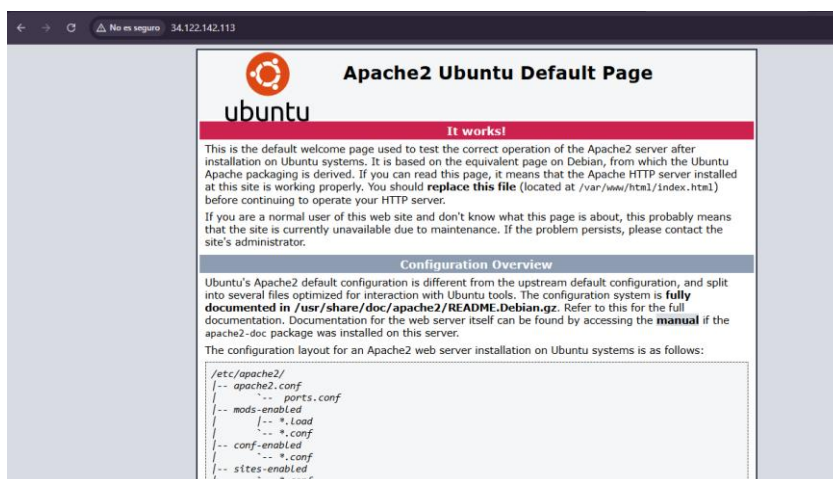


```
root@vps-servicio-medidor:~# sudo su
root@vps-servicio-medidor:~# cd ~
root@vps-servicio-medidor:~# sudo apt update && apt upgrade -y

Processing triggers for rsyslog (8.2001.0-1ubuntu1.3) ...
Processing triggers for ufw (0.36-6ubuntu1.1) ...
Processing triggers for systemd (245.4-4ubuntu3.24) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for linux-image-5.15.0-1077-gcp (5.15.0-1077.86~20.04.1) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.15.0-1077-gcp
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file '/etc/default/grub'
Sourcing file '/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file '/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-5.15.0-1077-gcp
Found initrd image: /boot/initrd.img-5.15.0-1077-gcp
Found linux image: /boot/vmlinuz-5.15.0-1074-gcp
Found initrd image: /boot/initrd.img-5.15.0-1074-gcp
Found Ubuntu 20.04.6 LTS (20.04) on /dev/sda1
Adding boot menu entry for UEFI Firmware Settings
done
root@vps-servicio-medidor:~#
```

Nota: Se ingresan los comandos necesarios para la instalación del sistema operativo y de Apache. Tomado de: Autoría propia.(20 febrero 2025).

Figura 78
Confirmar la correcta instalación.



Nota: En el navegador, copiamos nuevamente la IP desde el VPS y damos enter, podemos observar que Apache se instaló de manera correcta. Tomado de: Autoría propia.(20 febrero 2025).

- **Configuración del Dominio y DNS**

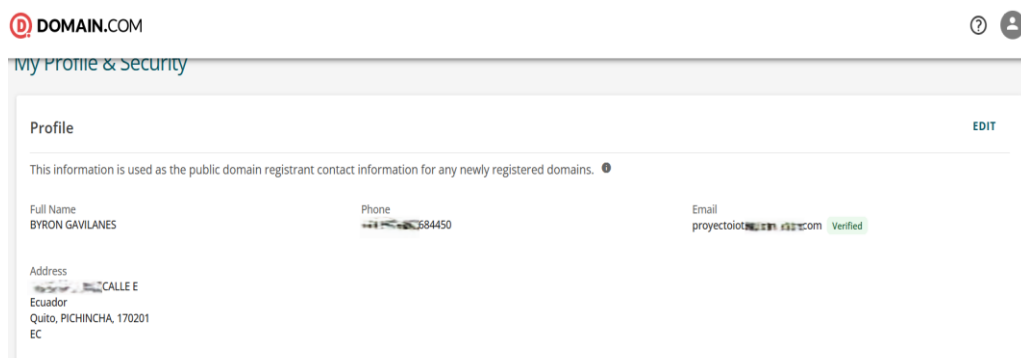
Para proporcionar un acceso más profesional y fácil de recordar, se adquirió un nombre de dominio a través de Domain.com. Se seleccionó un nombre adecuado y se completó el proceso de registro, asegurando la propiedad del dominio por el período determinado. (Figuras 76-77).

A continuación, en Google Cloud DNS, se creó una zona de DNS para gestionar el dominio, asociándolo con la dirección IP estática del VPS. Se configuraron los registros A y CNAME, estableciendo la relación entre el dominio y el servidor web.(Figuras 78-79-80-81).

Para validar la propagación del dominio, se utilizó la herramienta DNS Propagation Checker, permitiendo verificar la actualización de los registros a nivel global. Una vez propagado, se accedió al dominio desde un navegador, confirmando la correcta vinculación con el servidor. (Figura 82).

Figura 79

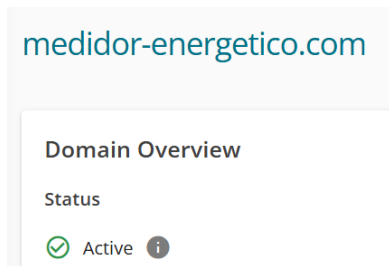
Crear cuenta en Domain.



Nota: Tomado de: Autoría propia.(20 febrero2025).

Figura 80

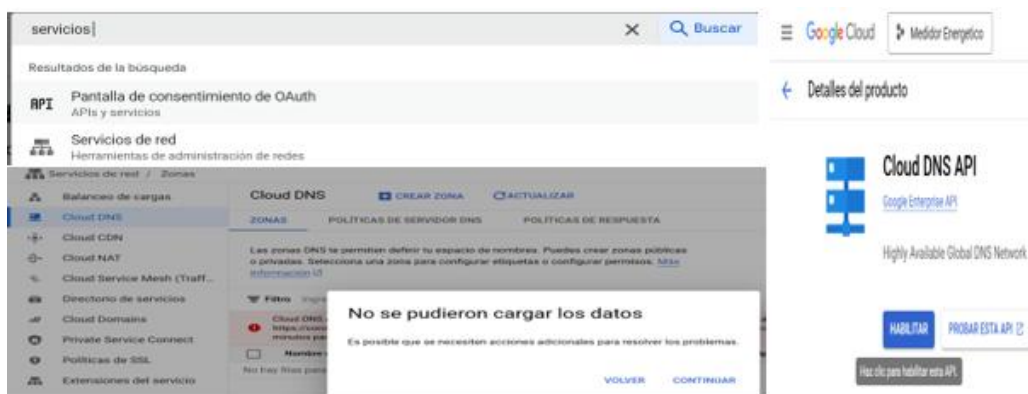
Elegir el nombre del dominio.



Nota: Buscamos un nombre acorde al proyecto y ver si está disponible. Posteriormente se realizará el pago correspondiente y habrá que esperar un momento para que se active. Tomado de: Autoría propia.(20 febrero 2025).

Figura 81

Apuntar el DNS a la IP externa.



Nota: Ingresar a servicios de red, seleccionar Cloud DNS y habilitar la API. Tomado de: Autoría propia(20 febrero 2025).

Figura 82

Crear Zona DNS.

The figure consists of three sequential screenshots from the Google Cloud DNS console. The first screenshot, titled 'Crea una zona del DNS', shows the 'Pública' (Public) zone type selected. The 'Nombre de la zona' (Zone name) is 'medidor-energetico', and the 'Nombre del DNS' (DNS name) is 'medidor-energetico.com'. The 'Descripción' (Description) is 'Adquirido en Domain'. The 'DNSSEC' status is 'Desactivado'. The second screenshot, titled 'Crea un conjunto de registros', shows the 'Tipo de registro de recursos' (Resource record type) set to 'A'. The 'Dirección IPv4' (IPv4 address) is '34.122.142.113'. The third screenshot, also titled 'Crea un conjunto de registros', shows the 'Tipo de registro de recursos' set to 'CNAME'. The 'Nombre canónico' (Canonical name) is 'www.medidor-energetico.com'.

Nota: Para configurar la zona DNS, crea una nueva zona con el nombre del dominio sin “.com” y su nombre DNS completo. Luego, agrega un registro A con la IP del VPS. Después, añade un registro CNAME, colocando “www” en el nombre DNS y el dominio con un punto al final en el nombre canónico. Tomado de: Autoría propia.(20 febrero 2025).

Figura 83

Vincular VPS con el dominio.

The screenshot shows the 'Manage Nameservers (DNS)' section of the Google Cloud DNS console. It displays the domain 'medidor-energetico.com.' with four nameservers: 'ns-cloud-d1.googlecloudnameservers.com.', 'ns-cloud-d2.googlecloudnameservers.com.', 'ns-cloud-d3.googlecloudnameservers.com.', and 'ns-cloud-d4.googlecloudnameservers.com.'. Below this, there is a section for 'Advanced Tools' with a 'MANAGE' button. The 'Nameservers (DNS)' section shows 'NS1.DOMAIN.COM (default)' and 'NS2.DOMAIN.COM (default)'. A 'CONTINUE' button is visible at the bottom right.

Nota: En el listado abrimos el NS y copiamos la primera línea, posterior vamos a DOMAIN y ADVANCED TOOLS click en MANAGE y continue. Tomado de: Autoría propia.(20 febrero2025).

Figura 84

Colocar nombres de los dominios del VPS en Domain.

Manage Nameservers (DNS)

[Revert to Defaults](#)

Nameserver 1 *	ns-cloud-d1.googledomains.com	×
Nameserver 2 *	ns-cloud-d2.googledomains.com	×
Nameserver 3 *	ns-cloud-d3.googledomains.com	×
Nameserver 4 *	ns-cloud-d4.googledomains.com	×

[+ NAMESERVER](#)

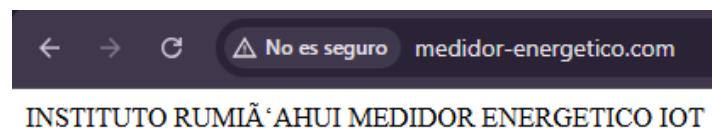
Note: You may assign custom nameservers but first you need to create them in the [Custom Nameserver](#) area.

[CANCEL](#) [SAVE](#)

Nota: Tomado de: Autoría propia.(20 febrero 2025).

Figura 85

Verificar Dominio.



Nota: Colocar el nombre del dominio en el navegador y dar ENTER. Se debe tener en cuenta que para que el dominio se propague hay que esperar un tiempo, que puede ser 1 hora o hasta 72 horas. Tomado de: Autoría propia.(20 febrero 2025).

- **Firewall**

Como medida de seguridad y para garantizar la correcta comunicación del sistema, se ajustaron las reglas del firewall en Google Cloud. Se eliminaron las reglas predefinidas(Figura 83) y se crearon nuevas políticas de acceso, habilitando puertos específicos según los servicios requeridos. (Figura 84).

Figura 86

Crear política de Firewall.

The screenshot shows the 'CREAR POLÍTICA DE FIREWALL' (Create Firewall Policy) interface. At the top, there is a button labeled 'CREAR POLÍTICA DE FIREWALL'. Below it, a table lists existing firewall rules:

	Nombre	Tipo	Destinos	Filtros	Protocolos/puertos	Acción	Prioridad
☐	default-allow-http	Entrada	http-server	Intervalos	tcp:80	Permitir	1000
☐	default-allow-https	Entrada	https-server	Intervalos	tcp:443	Permitir	1000

Below the table are buttons for 'ACTUALIZAR', 'CONFIGURAR REGISTROS', and 'BORRAR'. A filter input field is present with the placeholder 'Filtro Ingresar el nombre o el valor de la propiedad'. Below the filter, a table shows the same rules with checkboxes for selection:

	Nombre	Tipo	Destinos	Filtros	Protocolos/puertos	Acción	Prioridad
☒	default-allow-http	Entrada	http-server	Intervalos	tcp:80	Permitir	1000
☒	default-allow-https	Entrada	https-server	Intervalos	tcp:443	Permitir	1000

Below the table, a confirmation dialog asks: '¿Borrar 2 reglas de firewall?' (Delete 2 firewall rules?). It includes a sub-question '¿Seguro que quieres borrar 2 reglas de firewall?' and two buttons: 'CANCELAR' and 'BORRAR'.

Nota: Cuando se creó el VPS se crearon estas reglas para los puertos 80 y 443, se procede a eliminar los mismos para crear nuevas reglas. Tomado de: Autoría propia.(20 febrero 2025).

Figura 87

Crear Nueva Regla Firewall.

The screenshot shows the 'CREAR REGLA DE FIREWALL' (Create Firewall Rule) form. It includes a button labeled 'CREAR REGLA DE FIREWALL' and a link for 'Más información'. The form fields are:

- Nombre ***: A text input field containing 'puertos-habilitados'. A note below states: 'Se permiten letras minúsculas, números y guiones'.
- Destinos**: A dropdown menu showing 'Todas las instancias de la red'.
- Rangos de IPv4 de origen ***: A text input field containing '0.0.0.0/0'. A note below states: 'por ejemplo, 0.0.0.0/0, 192.168.2.0/24'.
- Protocolos**: A section with checkboxes for 'TCP', 'UDP', 'SCTP', and 'Otra'. The 'TCP' checkbox is checked. Below 'TCP' is a text input field for 'Puertos' containing '80,443,22,1883,1880,18083,3000,3306'. A note below states: 'P. ej., 20, 50-60'. Below 'Otra' is a text input field for 'Protocolos'.
- INHABILITAR REGLA**: A checkbox that is currently unchecked.
- Buttons**: 'CREAR' and 'CANCELAR' buttons at the bottom.

Nota: Una vez eliminados damos clic en CREAR NUEVA REGLA DE FIREWALL, Colocamos un nombre, En destinos colocamos “Todas las instancias de red”, Rangos de origen colocamos 0.0.0.0/0, Habilitar TCP para poder agregar los puertos.

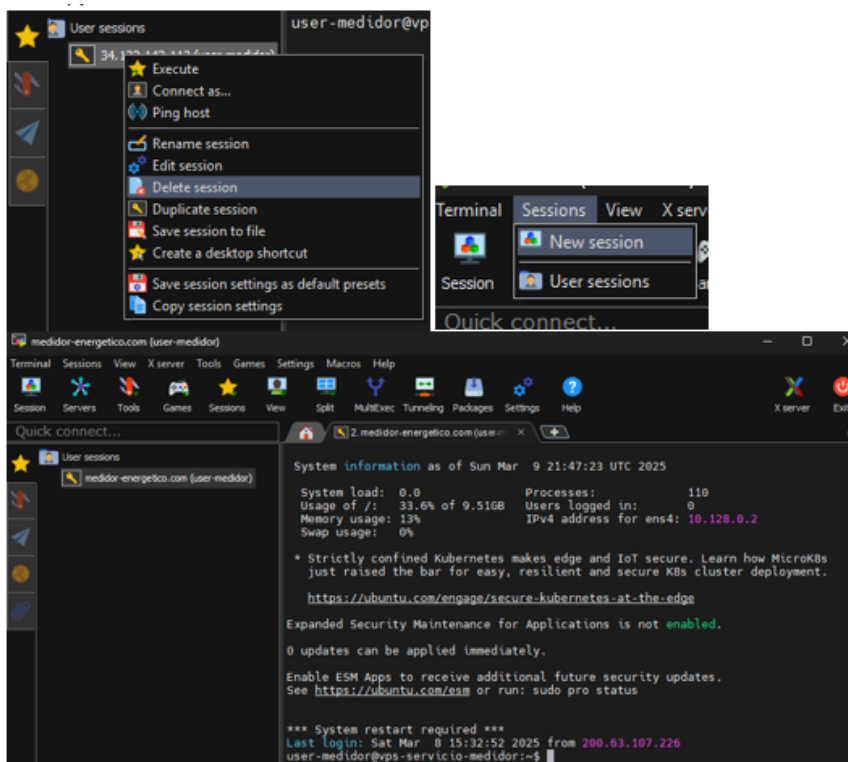
- **Certificados HTTPS y su Implementación en Apache**

Los certificados HTTPS son credenciales digitales que garantizan la seguridad de la comunicación entre un servidor web y sus usuarios, utilizando el protocolo TLS (Transport Layer Security). Su función principal es cifrar la información transmitida, protegiéndola contra interceptaciones o manipulaciones por terceros. Además, estos certificados validan la autenticidad del sitio web, evitando ataques de suplantación de identidad (phishing).

En la implementación de la plataforma web IoT, se configuró un certificado SSL en Apache, lo que permitió que el dominio operara de forma segura mediante HTTPS, asegurando la confidencialidad e integridad de los datos.

Figura 88

Ingresa al VPS.



Nota: Se ingresó a MobaXterm, eliminando la sesión con la IP del VPS y creando una nueva sesión utilizando el dominio sin HTTPS.

Figura 89

Preparación del Entorno.

```
root@vps-servicio-medidor:~# sudo apt-get install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.99.9.12).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@vps-servicio-medidor:~# sudo add-apt-repository universe
'universe' distribution component is already enabled for all sources.
root@vps-servicio-medidor:~# sudo apt install snapd
Reading package lists... Done
Building dependency tree
Reading state information... Done
snapd is already the newest version (2.66.1+20.04).
snapd set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@vps-servicio-medidor:~# snap --version
snap      2.67.1
snapd     2.67.1
series    16
ubuntu    20.04
kernel    5.15.0-1074-gcp
root@vps-servicio-medidor:~# sudo snap install core; sudo snap refresh core
core 16-2.61.4-20240607 from Canonical✓ installed
snap "core" has no updates available
```

Nota: Se ingresaron los comandos para elevar privilegios a superusuario (root), se instaló snapd, un gestor de paquetes necesario para la instalación de herramientas de seguridad, se verificó la versión instalada y se procedió a instalar Core, asegurando que todas las dependencias estuvieran actualizadas.

Figura 90

Instalación Certbot y Generación del Certificado.

```
root@vps-servicio-medidor:~# sudo snap install --classic certbot
certbot 3.2.0 from Certbot Project (certbot-eff) installed
root@vps-servicio-medidor:~# sudo ln -s /snap/bin/certbot /usr/bin/certbot
root@vps-servicio-medidor:~# clear
root@vps-servicio-medidor:~# sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): proyectoiot2025@gmail.com

-----
Please read the Terms of Service at:
https://letsencrypt.org/documents/LE-SA-v1.5-February-24-2025.pdf
You must agree in order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: Y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: N

Account registered.
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel): medidor-energetico.com
Requesting a certificate for medidor-energetico.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/medidor-energetico.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/medidor-energetico.com/privatekey.pem
This certificate expires on 2025-06-07.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for medidor-energetico.com to /etc/apache2/sites-available/000-default-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://medidor-energetico.com

-----
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
 * Donating to EFF: https://eff.org/donate-le
-----

root@vps-servicio-medidor:~# sudo certbot renew --dry-run
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Processing /etc/letsencrypt/renewal/medidor-energetico.com.conf
-----
Account registered.
Simulating renewal of an existing certificate for medidor-energetico.com

-----
Congratulations, all simulated renewals succeeded:
/etc/letsencrypt/live/medidor-energetico.com/fullchain.pem (success)
-----
```

Nota: Se instaló Certbot, una herramienta automatizada para la gestión de certificados SSL/TLS, y se ejecutó el comando para solicitar el certificado. Se ingresó el correo del proyecto, se aceptaron los términos de uso y se seleccionó la versión gratuita de Let's Encrypt. Finalmente, se introdujo el dominio del servidor, generando el certificado HTTPS.

Instalación de MySQL y creación de Usuarios en VPS

Para la instalación de MySQL en el VPS, se accedió al servidor mediante MobaXterm y, una vez cargado el sistema, se ingresó como superusuario utilizando los comandos `sudo su` y `cd ~` para moverse al directorio raíz. Posteriormente, se ejecutó la instalación de MySQL con `sudo apt -y install mysql-server` y se verificó su estado mediante `systemctl status mysql`, asegurando su correcta habilitación. Luego, se procedió a establecer una contraseña segura para el usuario `root` con `ALTER USER 'root'@'localhost' IDENTIFIED WITH`

mysql_native_password BY 'MySql2025';, recordando que cualquier espacio en blanco accidental en la clave podría generar errores de acceso.

Figura 91

Configuración de MySql.

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-03-10 01:48:53 UTC; 3min 45s ago
     Main PID: 130013 (mysqld)
    Status: "Server is operational"
      Tasks: 37 (limit: 2342)
     Memory: 363.5M
    CGroup: /system.slice/mysql.service
            └─130013 /usr/sbin/mysqld

Mar 10 01:48:52 vps-servicio-medidor systemd[1]: Starting MySQL Community Server...
Mar 10 01:48:53 vps-servicio-medidor systemd[1]: Started MySQL Community Server.
root@vps-servicio-medidor:~#

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by ' MySql2025';
Query OK, 0 rows affected (0.01 sec)

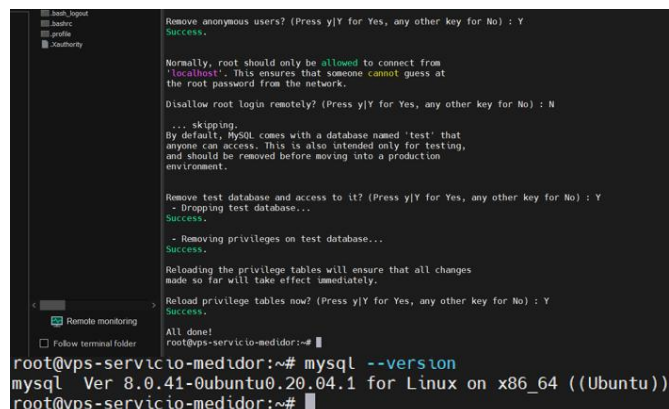
mysql> EXIT;
Bye
root@vps-servicio-medidor:~#
```

Nota: Se verificó el estado del servicio MySQL y se actualizó la contraseña del usuario root mediante mysql_native_password. Fuente: Elaboración Propia.

Después de configurar la contraseña, se ejecutó `sudo mysql_secure_installation` para fortalecer la seguridad del sistema, ingresando nuevamente la clave establecida y configurando el nivel de seguridad en 2. Durante este proceso, se habilitaron opciones como la eliminación de usuarios anónimos y la restricción de acceso remoto, asegurando que el servidor MySQL estuviera protegido. Para verificar la instalación, se ejecutó `mysql --version`, confirmando la versión instalada en este caso la MySQL 8.0.41-0ubuntu0.20.04.1, y se inició el servicio con `sudo service mysql start`.

Figura 92

Configuración de seguridad en MySQL.



```
root@vps-service-medidor:~# mysql --version
mysql Ver 8.0.41-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
root@vps-service-medidor:~#
```

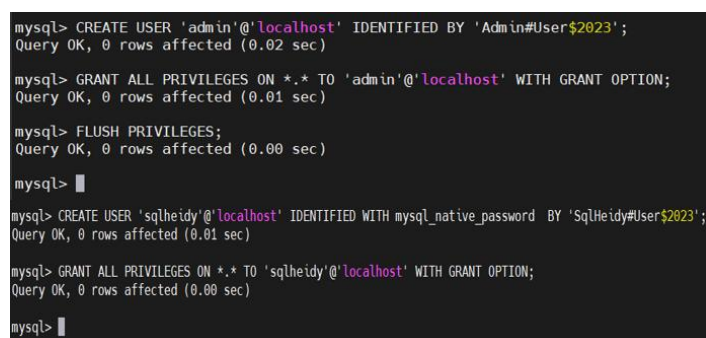
The screenshot shows the MySQL security configuration process. It starts with a prompt to remove anonymous users, which is confirmed with 'Y'. Then, it asks to disallow root login remotely, which is confirmed with 'N'. Next, it asks to remove the test database and access to it, which is confirmed with 'Y'. Finally, it asks to reload the privilege tables, which is confirmed with 'Y'. The process ends with the message 'All done!' and the prompt returns to the root user.

Nota: Se eliminaron usuarios anónimos, restringió el acceso de root y eliminó la base de datos de prueba. Luego, se verificó la versión de MySQL. Fuente: Elaboración Propia.

Finalmente, se procedió a la creación de usuarios con diferentes privilegios para la administración de la base de datos. Se accedió a MySQL con `sudo mysql -u root -p`, ingresando la contraseña establecida, y se crearon dos usuarios: "admin" con privilegios completos y "sqlheidy", quien también recibió permisos administrativos. Los comandos utilizados fueron `CREATE USER`, `GRANT ALL PRIVILEGES` y `FLUSH PRIVILEGES` para asegurar que las cuentas creadas tuvieran los permisos adecuados. Con esto, se finalizó el procedimiento, asegurando un entorno de base de datos correctamente instalado y configurado.

Figura 93

Creación de Usuarios en MySQL.



```
mysql> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'Admin#User$2023';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> CREATE USER 'sqlheidy'@'localhost' IDENTIFIED WITH mysql_native_password BY 'SqlHeidy#User$2023';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'sqlheidy'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

The screenshot shows the MySQL command-line interface. It displays the execution of several commands to create two users: 'admin' and 'sqlheidy'. Both users are created with full privileges and the ability to grant privileges. The commands are executed successfully, as indicated by the 'Query OK' messages.

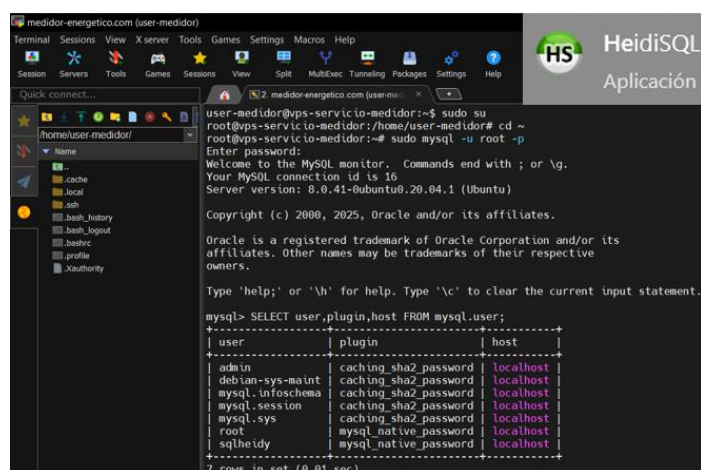
Nota: Se crearon los usuarios admin y sqlheidy en MySQL, asignándoles contraseñas y permisos administrativos. Se ejecutaron los comandos necesarios para su configuración y activación en el servidor. Fuente: Elaboración Propia.

Instalación de Heidi MySQL

Para facilitar la administración de la base de datos MySQL en el VPS, se instala HeidiSQL, un cliente que permite gestionar las bases de datos de manera visual desde un computador. Primero, se descarga e instala HeidiSQL desde su página oficial. Luego, se inicia sesión en el servidor VPS utilizando MobaXterm y se accede a MySQL como usuario root, con el comando `sudo mysql -u root -p`, ingresando la clave de acceso establecida previamente. Para verificar la configuración actual de los usuarios en MySQL, se ejecuta el comando `SELECT user, plugin, host FROM mysql.user;`, lo que permite observar el método de autenticación y el host asociado a cada usuario.

Figura 94

Aplicación HeidiSql e ingreso a MariaDB mediante MobaXterm.



The screenshot shows a MobaXterm terminal window with the HeidiSQL application running. The terminal output shows the user logging in as root and executing the command `SELECT user, plugin, host FROM mysql.user;`. The result is a table with 7 rows showing various MySQL users and their authentication plugins and hosts.

user	plugin	host
admin	caching_sha2_password	localhost
debian-sys-maint	caching_sha2_password	localhost
mysql.infoschema	caching_sha2_password	localhost
mysql.session	caching_sha2_password	localhost
mysql.sys	caching_sha2_password	localhost
root	mysql_native_password	localhost
sqlheidy	mysql_native_password	localhost

Nota: Se observa comandos ingresados y que sqlheidy tiene un (mysql_native_password) lo cual no nos permite la comunicación con una IP externa. Fuente: Elaboración Propia.

Para habilitar el acceso remoto, se cambia la configuración del usuario sqlheidy, permitiéndole conectarse desde cualquier IP. Esto se logra ejecutando los comandos `RENAME USER 'sqlheidy'@'localhost' TO 'sqlheidy'@'%';` y `ALTER USER 'sqlheidy'@'%'` IDENTIFIED WITH mysql_native_password BY 'SqlHeidy#User\$****';, seguidos de la asignación de privilegios globales con `GRANT ALL PRIVILEGES ON *.* TO 'sqlheidy'@'%';`. (Figura 95). Posteriormente, se reinicia el servicio MySQL con `sudo service mysql restart`, se habilita el puerto 3306 en el firewall (`sudo ufw allow 3306`), y se modifica la configuración del archivo `mysqld.cnf` para permitir conexiones externas, asegurándose de que MariaDB esté escuchando en todas las interfaces de red.

Figura 95

Configuración de sqlheidy para conexión con IP externa.

```
mysql> RENAME USER 'sqlheidy'@'localhost' TO 'sqlheidy'@'%';
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER USER 'sqlheidy'@'%' IDENTIFIED WITH mysql_native_password BY 'SqlHeidy#User$2';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'sqlheidy'@'%';
Query OK, 0 rows affected (0.00 sec)
```

Nota: Se muestra la configuración del usuario sqlheidy en MySQL para permitir conexiones remotas mediante la modificación del host, autenticación y asignación de privilegios. Fuente: Elaboración Propia.

Figura 96

Configuraciones.

```
mysql> SELECT user,plugin,host FROM mysql.user;
+-----+-----+-----+
| user          | plugin                | host          |
+-----+-----+-----+
| sqlheidy      | mysql_native_password | %             |
| admin         | caching_sha2_password | localhost     |
| debian-sys-maint | caching_sha2_password | localhost     |
| mysql.infoschema | caching_sha2_password | localhost     |
| mysql.session | caching_sha2_password | localhost     |
| mysql.sys     | caching_sha2_password | localhost     |
| root          | mysql_native_password | localhost     |
+-----+-----+-----+

bind-address            = 127.0.0.1
mysqlx-bind-address     = 127.0.0.1
#
bind-address            = 0.0.0.0
mysqlx-bind-address     = 0.0.0.0
#

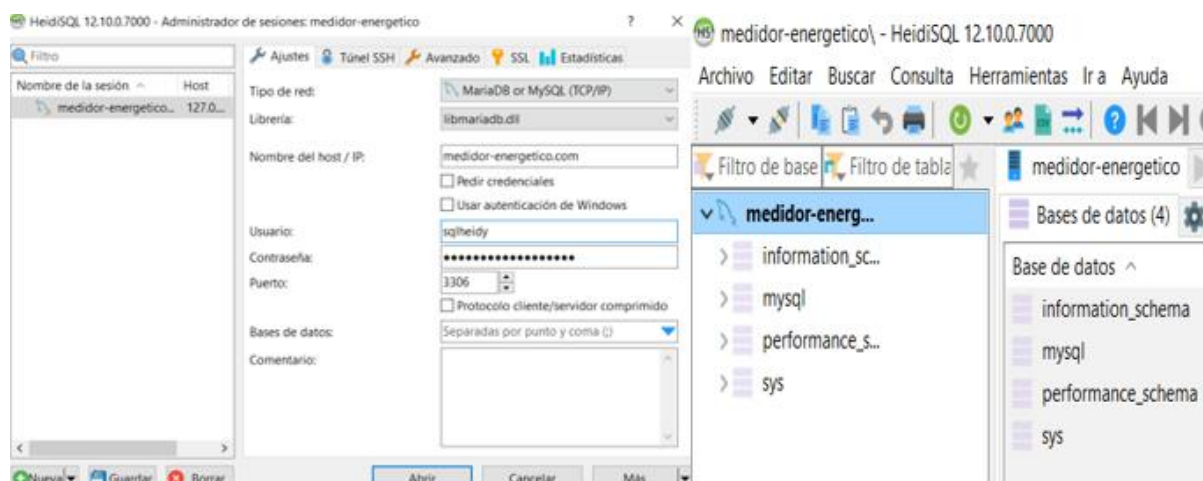
root@vps-servicio-medidor:~# netstat -ant | grep 3306
tcp        0      0 0.0.0.0:3306          0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:3306          0.0.0.0:*             LISTEN
root@vps-servicio-medidor:~#
```

Nota: Configuración y verificación de MySQL para permitir conexiones remotas, modificación del archivo de configuración para aceptar conexiones externas y la verificación del puerto 3306 en escucha. Fuente: Elaboración Propia.

Con la configuración lista, se abre HeidiSQL en el computador y se crea una nueva conexión. Se ingresa el dominio o la dirección IP del VPS en el campo host, el usuario sqlheidy y su contraseña correspondiente. Tras guardar los cambios y conectar, el sistema solicita una confirmación, y al aceptar, se establece correctamente la vinculación con la base de datos. Este procedimiento garantiza que se pueda administrar la base de datos desde un equipo externo sin necesidad de acceder manualmente al VPS, optimizando así el monitoreo y gestión de los datos de la plataforma IoT.

Figura 97

Configuración de HeidiSQL e ingreso a interfaz.



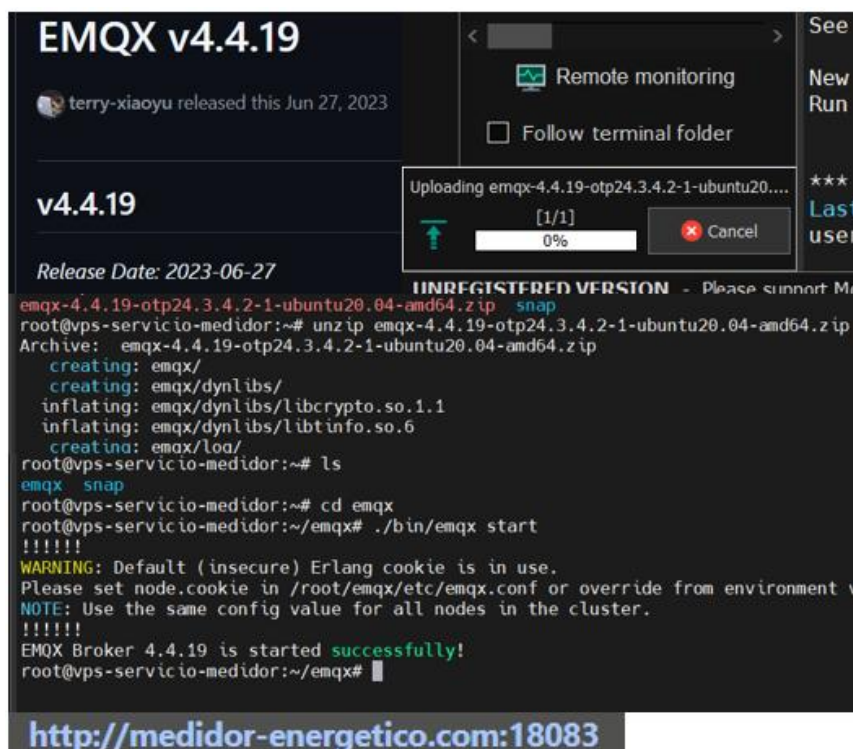
Nota: Ingreso a la base de datos del VPS mediante HeidiSQL. Fuente: Elaboración Propia.

Instalar EMQX, reinicio de EMQX y configuración de EMQX NodeName

Para instalar el Broker MQTT EMQX en el servidor, se comienza descargando la versión 4.4.19 en el sitio oficial en GitHub. Una vez descargado el archivo comprimido, se sube al servidor VPS utilizando MobaXterm. Para proceder con la instalación, se descomprime el archivo utilizando el comando `unzip` y se mueve a la carpeta raíz con el comando `mv emqx-4.4.19... /root`. Posteriormente, se accede al directorio de EMQX, se inicia el servicio con `./bin/emqx start` y se confirma su correcto inicio. Después de esto, se accede al entorno web de EMQX en el navegador, ingresando la IP del servidor seguida del puerto 18083, usando las credenciales predeterminadas de usuario `admin` y contraseña `public`. Es recomendable cambiar la contraseña predeterminada para mayor seguridad. Una vez dentro, se realiza el cambio de clave a una más segura y se guarda la configuración.

Figura 98

Instalación y configuración de EMQX.



```
EMQX v4.4.19
terry-xiaoyu released this Jun 27, 2023

v4.4.19
Release Date: 2023-06-27

root@vps-servicio-medidor:~# unzip emqx-4.4.19-otp24.3.4.2-1-ubuntu20.04-amd64.zip
Archive: emqx-4.4.19-otp24.3.4.2-1-ubuntu20.04-amd64.zip
  creating: emqx/
  creating: emqx/dynlibs/
  inflating: emqx/dynlibs/libcrypto.so.1.1
  inflating: emqx/dynlibs/libtinfo.so.6
  creating: emqx/priv/
root@vps-servicio-medidor:~# ls
emqx  snap
root@vps-servicio-medidor:~# cd emqx
root@vps-servicio-medidor:~/emqx# ./bin/emqx start
!!!!!!
WARNING: Default (insecure) Erlang cookie is in use.
Please set node.cookie in /root/emqx/etc/emqx.conf or override from environment
NOTE: Use the same config value for all nodes in the cluster.
!!!!!!
EMQX Broker 4.4.19 is started successfully!
root@vps-servicio-medidor:~/emqx#
```

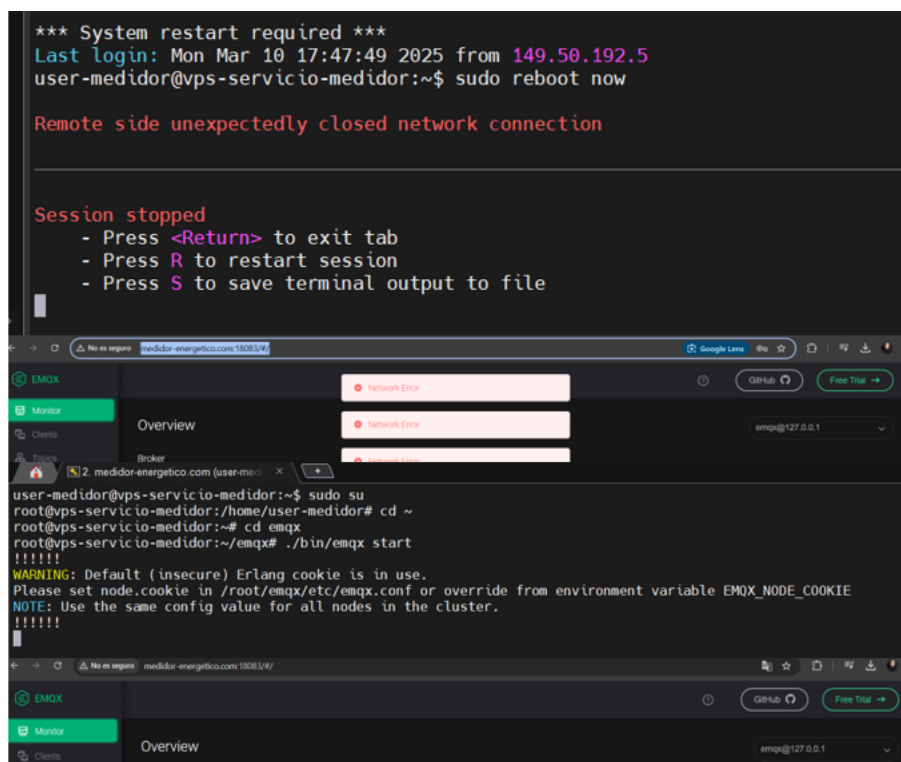
<http://medidor-energetico.com:18083>

Nota: Configuración e inicio del broker EMQX v4.4.19 en un servidor VPS mediante mobaxterm, incluyendo la extracción del archivo, ejecución del servicio y visualización de la URL de acceso. Fuente: Elaboración propia.

Para reiniciar el servicio EMQX primero se ejecuta el comando `sudo reboot now` para reiniciar el VPS. Al reiniciar el servidor, puede aparecer un error al intentar acceder al entorno web de EMQX debido a que el servidor está fuera de línea. Una vez transcurridos unos minutos, se reinicia el servicio de EMQX usando los comandos `sudo su`, `cd ~`, `cd emqx`, y `./bin/emqx start` para asegurar que el servicio esté funcionando correctamente. Si se requiere reiniciar solo el servicio de EMQX, se puede ejecutar `./bin/emqx restart`, y posteriormente, se accede a la interfaz web de EMQX para verificar que todo esté en funcionamiento. Esto garantiza que el sistema MQTT esté operativo y listo para la gestión de dispositivos IoT.

Figura 99

Reinicio de VPS, EMQX.

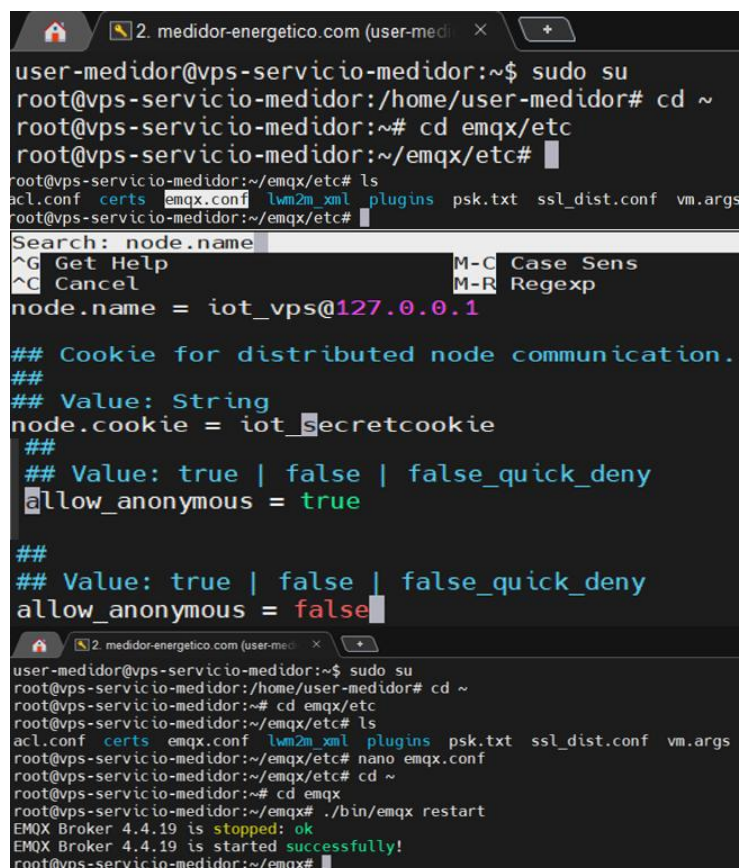


Nota: La imagen muestra el reinicio del VPS y EMQX, el panel de la interfaz web EMQX cuando se reinicia y se restablece funcionamiento Fuente: Elaboración propia.

Para cambiar el nombre del nodo y desactivar los usuarios anónimos en EMQX, se comienza cerrando sesión en la interfaz web de EMQX. Luego, se accede al servidor VPS mediante MobaXterm, se navega hasta la carpeta de configuración de EMQX (`cd emqx/etc`) y se edita el archivo `emqx.conf` utilizando un editor de texto como nano. Se busca y modifica el `node.name` y `node.cookie` los cuales fueron `iot_vps@127.0.0.1`, `iot_secretcookie` para personalizar el nombre del nodo y la clave secreta. Además, se busca y cambia la opción `allow_anonymous = true` por `allow_anonymous = false`, lo que asegura que solo usuarios autenticados puedan conectarse al servicio. Después de guardar los cambios con `Control + O`, `Enter` y salir con `Control + X`, se reinicia el servicio con el comando `./bin/emqx restart`. Esto desactiva los usuarios anónimos y asegura que solo usuarios con credenciales puedan acceder al servicio, mejorando la seguridad del sistema.

Figura 100

Configuración de Nodename.



```
user-medidor@vps-servicio-medidor:~$ sudo su
root@vps-servicio-medidor:/home/user-medidor# cd ~
root@vps-servicio-medidor:~# cd emqx/etc
root@vps-servicio-medidor:~/emqx/etc#
root@vps-servicio-medidor:~/emqx/etc# ls
acl.conf  certs  emqx.conf  lwm2m.xml  plugins  psk.txt  ssl_dist.conf  vm.args
root@vps-servicio-medidor:~/emqx/etc#
Search: node.name
^G Get Help          M-C Case Sens
^C Cancel            M-R Regexp
node.name = iot_vps@127.0.0.1

## Cookie for distributed node communication.
##
## Value: String
node.cookie = iot_secretcookie
##
## Value: true | false | false_quick_deny
allow_anonymous = true

##
## Value: true | false | false_quick_deny
allow_anonymous = false

user-medidor@vps-servicio-medidor:~$ sudo su
root@vps-servicio-medidor:/home/user-medidor# cd ~
root@vps-servicio-medidor:~# cd emqx/etc
root@vps-servicio-medidor:~/emqx/etc# ls
acl.conf  certs  emqx.conf  lwm2m.xml  plugins  psk.txt  ssl_dist.conf  vm.args
root@vps-servicio-medidor:~/emqx/etc# nano emqx.conf
root@vps-servicio-medidor:~/emqx/etc# cd ~
root@vps-servicio-medidor:~# cd emqx
root@vps-servicio-medidor:~/emqx# ./bin/emqx restart
EMQX Broker 4.4.19 is stopped: ok
EMQX Broker 4.4.19 is started successfully!
root@vps-servicio-medidor:~/emqx#
```

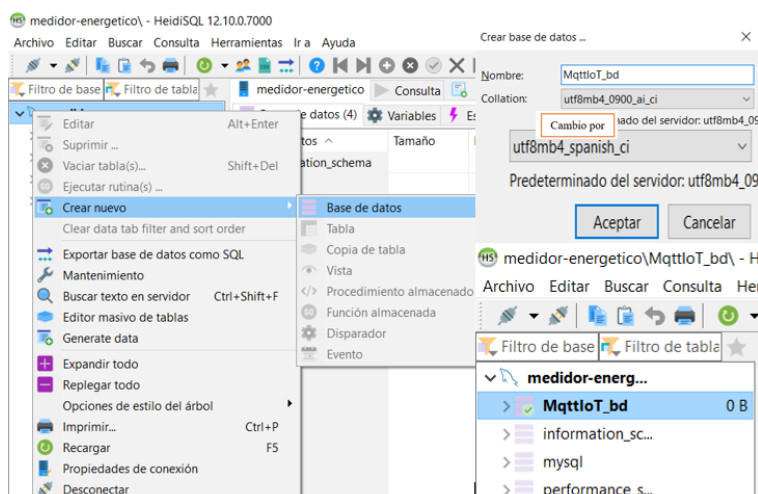
Nota: Esta imagen muestra el resultado de la modificación de la opción `allow_anonymous`, donde se configura a `false` para asegurar que solo los usuarios autenticados puedan acceder a EMQX. Fuente: Elaboración propia.

Creación de tablas MySQL con HeidiSQL

Para establecer la conexión entre MySQL y EMQX y permitir la comunicación entre los dispositivos IoT y el servidor, primero se accede a HeidiSQL con el usuario `sqlheidy` y la contraseña correspondiente. Una vez dentro, se visualizan las bases de datos preinstaladas en el servidor MySQL. A continuación, se procede a la creación de dos nuevas bases de datos denominadas `MqttIoT_bd` y `EspIoT_bd`. Para ello, se hace clic derecho sobre la sección de bases de datos, se selecciona "Crear nuevo" y luego "Base de datos". Se ingresa el nombre correspondiente, asegurándose de que no contenga espacios en blanco al final, y se selecciona la collation `utf8mb4_spanish_ci` antes de aceptar la configuración.

Figura 101

Creación de base de Datos MqttIoT_bd.

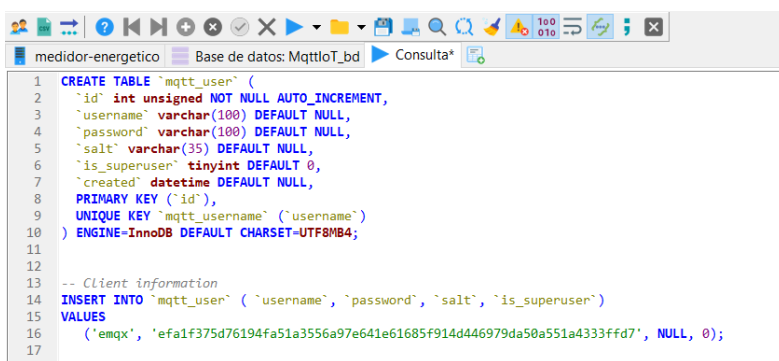


Nota: Esta imagen muestra la creación de la base de datos, secuencia de pestañas y cambio de collation. Fuente: Elaboración propia.

Con la base de datos creada, se accede a la pestaña de consultas y se ejecuta el comando SQL para generar la tabla mqtt_user, la cual contendrá los campos username, password, salt, is_superuser y la fecha de creación. Posteriormente, se insertan valores en esta tabla mediante el comando INSERT INTO, registrando un usuario de prueba llamado emqx con una contraseña encriptada.

Figura 102

Creación de Tabla mqtt_user

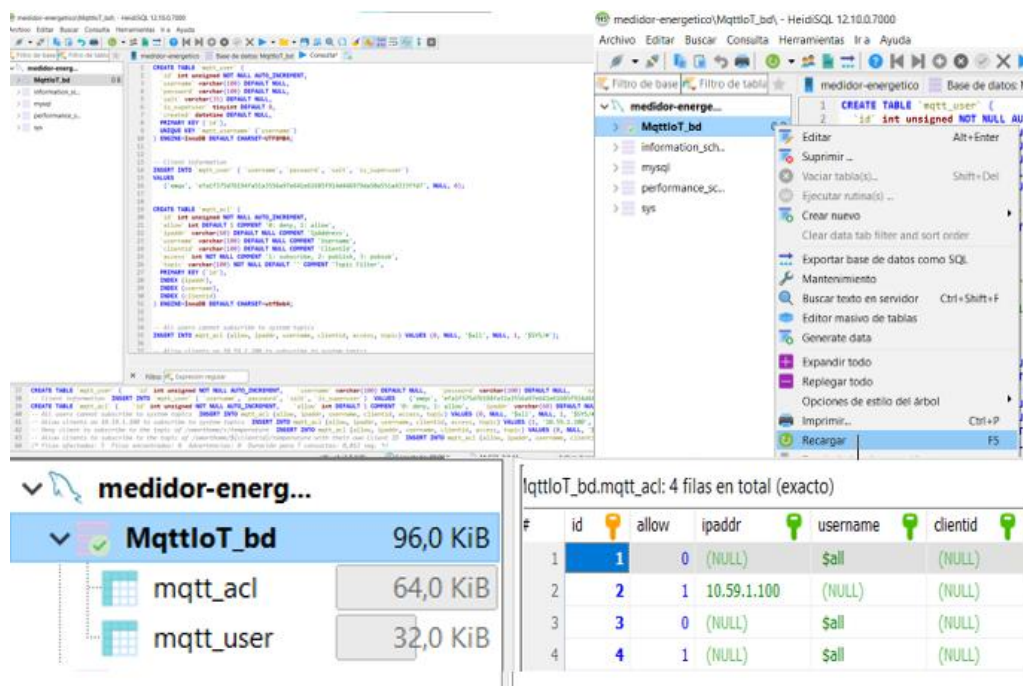


Nota: En esta imagen se puede observar los comandos insertados para la tabla, incorporando el nombre de usuario, contraseña, etc. y la información englobada. Fuente: Elaboración propia.

A continuación, se crea una segunda tabla denominada `mqtt_acl`, encargada de gestionar los permisos de acceso de los clientes MQTT. Esta tabla incluye los campos `ipaddr`, `username`, `clientid`, `access` y `topic`, permitiendo definir reglas de acceso para suscripción y publicación de mensajes. Una vez creada la estructura de la base de datos, se ejecutan los comandos SQL en secuencia y se recargan las tablas para visualizar los registros insertados.

Figura 103

Creación de Tabla `mqtt_acl` y ejecución de comandos.

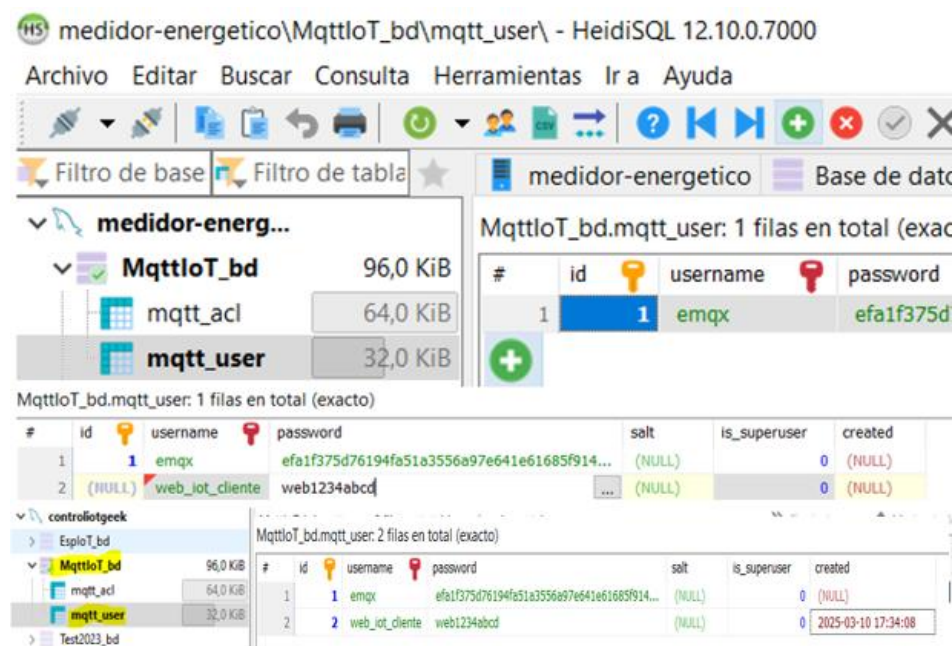


Nota: En esta imagen se puede observar los comandos insertados para la tabla, la ejecución en secuencia de los comandos, el procedimiento de recargar y finalmente las tablas de datos creadas con su información. Fuente: Elaboración propia.

Finalmente, se adiciona un nuevo usuario denominado `web_iot_cliente` con la contraseña `web1234****`. Para ello, se selecciona la opción de agregar un nuevo registro en la tabla `mqtt_user`, se ingresan los valores correspondientes y se guarda la configuración. Se verifica que los campos obligatorios no contengan valores NULL en este caso se selecciona la celda hasta que nos aparezca la fecha de creación, damos click fuera de las celdas para refrescar. Con estos pasos, se completa la configuración de la base de datos MySQL para su integración con EMQX, asegurando que la información de autenticación y permisos quede correctamente almacenada.

Figura 104

Adición de Nuevo Usuario Web_iot_cliente.



Nota: En esta imagen se puede observar el procedimiento para agregar una nueva celda a la tabla mqtt_user, colocando datos y observando la fecha de creación Fuente: Elaboración propia.

Configurando MySQL en EMQX

Para establecer la comunicación entre MySQL y EMQX, primero se debe acceder al servidor VPS utilizando MobaXterm e ingresar al directorio de configuración de plugins de EMQX con el comando `cd emqx/etc/plugins`. Luego, se edita el archivo `emqx_auth_mysql.conf` con un editor de texto como nano (`nano emqx_auth_mysql.conf`). En este archivo, se deben modificar los parámetros de conexión a la base de datos, reemplazando el valor predeterminado de `auth.mysql.database` por el nombre de la base de datos creada en MySQL, por ejemplo, `MqttIoT_bd`. Además, es necesario configurar las credenciales de acceso, eliminando los caracteres `#` de las líneas correspondientes y estableciendo el usuario y contraseña previamente creados en MySQL. También se debe cambiar el método de almacenamiento de contraseñas de `sha256` a `plain`. Una vez realizados los cambios, se guardan con `Ctrl + O`, se sale del editor con `Ctrl + X` y se reinicia EMQX ejecutando `./bin/emqx restart`. Para garantizar una comunicación segura, se habilita el puerto 8084 con el comando `sudo ufw allow 8084` y se reinicia Apache con `systemctl restart apache2`. Finalmente, desde la interfaz web de EMQX, en la pestaña de plugins, se activa el plugin correspondiente, permitiendo la

Figura 106

Certificados en EMQX.

```
root@vps-servicio-medidor:/etc/letsencrypt/live# ls
README  medidor-energetico.com
root@vps-servicio-medidor:/etc/letsencrypt/live#
root@vps-servicio-medidor:/etc/letsencrypt/live# cd medidor-energetico.com
root@vps-servicio-medidor:/etc/letsencrypt/live/medidor-energetico.com# ls
README  cert.pem  chain.pem  fullchain.pem  privkey.pem
root@vps-servicio-medidor:/etc/letsencrypt/live/medidor-energetico.com#

root@vps-servicio-medidor:~/emqx/etc/plugins# ls
cl.conf.paho  emqx_auth_mnesia.conf  emqx_auth_redis.conf  emqx_exhook.conf  emqx_management.conf  emqx_retainer.conf  emqx_stomp.conf
emqx_auth_http.conf  emqx_auth_mongo.conf  emqx_bridge_mqtt.conf  emqx_exproto.conf  emqx_prometheus.conf  emqx_rule_engine.conf  emqx_telemetry.conf
emqx_auth_jwt.conf  emqx_auth_mysql.conf  emqx_bridge_tcp.conf  emqx_lang_hook.conf  emqx_psk_file.conf  emqx_sasl.conf  emqx_web_hook.conf
emqx_authldap.conf  emqx_auth_postgres.conf  emqx_dashboard.conf  emqx_lua2m.conf  emqx_recon.conf  emqx_sh.conf

root@vps-servicio-medidor:~/emqx/etc/plugins#

##-----
## HTTP Listener
## The port that the Dashboard HTTP listener will bind.
##
## Value: Port
## Examples: 18083
dashboard.listener.http = 18083
## Modifico colocando ##

## The acceptor pool for external Dashboard HTTP listener.
##
## Value: Number
dashboard.listener.http.acceptors = 4
##
## Maximum number of concurrent Dashboard HTTP connections.
##
## Value: Number
dashboard.listener.http.max_clients = 512
## Set up the socket for IPv6.
##
## Value: false | true
dashboard.listener.http.ipv6 = false
## Listen on IPv4 and IPv6 (false) or only on IPv6 (true). Use
##
## Value: false | true
dashboard.listener.http.ipv6_v6only = false

##-----
## HTTPS Listener
## The port that the Dashboard HTTPS listener will bind.
##
## Value: Port
## Examples: 18084
dashboard.listener.https = 18084
## Modifico quitando ##

## The acceptor pool for external Dashboard HTTPS listener.
##
## Value: Number
dashboard.listener.https.acceptors = 2
##
## Maximum number of concurrent Dashboard HTTPS connections.
##
## Value: Number
dashboard.listener.https.max_clients = 512
## Set up the socket for IPv6.
##
## Value: false | true
dashboard.listener.https.ipv6 = false
## Listen on IPv4 and IPv6 (false) or only on IPv6 (true). Use
##
## Value: false | true
dashboard.listener.https.ipv6_v6only = false
## Path to the file containing the user's private PEM-encoded key.
##
## Value: File
dashboard.listener.https.keyfile = /etc/letsencrypt/live/medidor-energetico.com/privkey.pem
## String containing the private key file password. Only used if the
## private keyfile is password-protected.
##
## Value: String
## dashboard.listener.https.key_password = yourpass
## Modifico

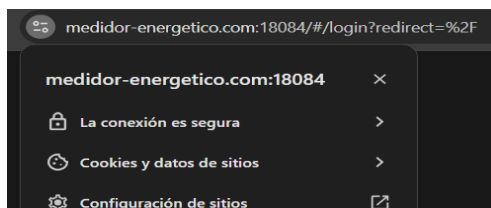
## Path to a file containing the user certificate.
##
## Value: File
dashboard.listener.https.certfile = /etc/letsencrypt/live/medidor-energetico.com/fullchain.pem
```

Nota: Se implementó un certificado de seguridad SSL/TLS utilizando Let's Encrypt para proteger la interfaz web de EMQX. Se modificó el archivo `emqx_dashboard.conf`, deshabilitando puertos no seguros y estableciendo las rutas de los certificados en la configuración del servidor. Fuente: Elaboración propia.

Después de guardar los cambios (Ctrl + O y Ctrl + X), se reinicia el servicio de EMQX con `./bin/emqx restart`. Para acceder al panel de control de manera segura, se debe ingresar a la URL `https://medidor-energetico.com:18084` (Figura 107), donde se validará la autenticidad del certificado y se solicitarán las credenciales de acceso previamente configuradas.

Figura 107

Verificación de Conexión Segura.



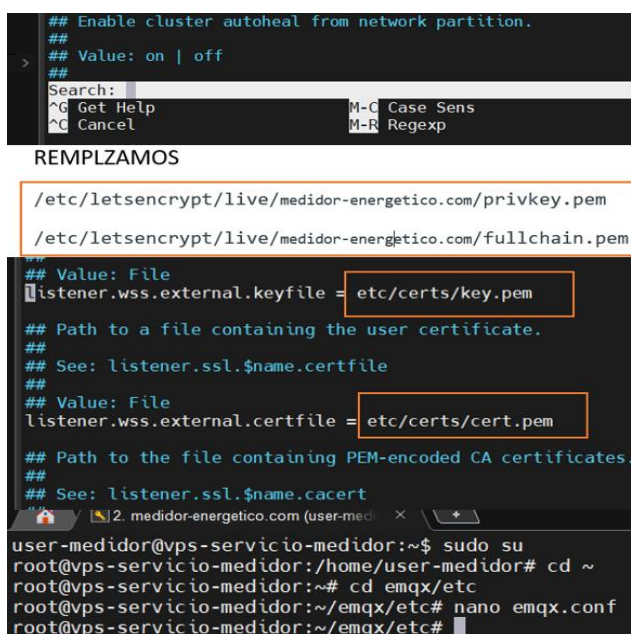
Nota: En la imagen se puede observar el ingreso `https://medidor-energetico.com:18084` y se comprueba que la conexión es Segura al Fuente: Elaboración propia.

Certificado Emqx y WSS

Para garantizar la seguridad en la comunicación entre EMQX y los clientes MQTT mediante WebSockets seguros (WSS), se configuran los certificados en el archivo de configuración de EMQX. Primero, se accede al VPS mediante `mobaxterm` y se edita el archivo `emqx.conf` con `nano emqx.conf`. Dentro del archivo, se utiliza el comando `Ctrl + W` para buscar las líneas `listener.wss.external.keyfile` y `listener.wss.external.certfile`, donde se deben modificar las rutas de los certificados para que coincidan con el dominio configurado.

Figura 108

Certificación en EMQX y WSS.

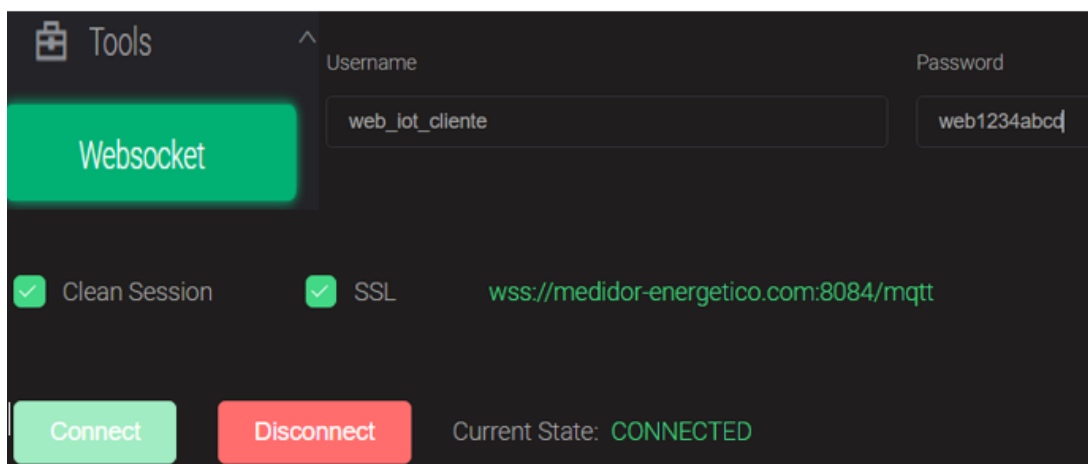


Nota: En la imagen se puede observar la configuración de WSS en EMQX, estableciendo certificados SSL/TLS y autenticación segura en `emqx.conf`. Fuente: Elaboración propia.

Luego, se guardan los cambios (Ctrl + O, Enter, Ctrl + X) y se reinicia EMQX con `./bin/emqx restart`. Una vez configurado, se accede a `https://medidor-energetico.com:18084/#/`, en la pestaña "Tools", dentro de la opción WebSocket. Aquí, se ingresa el usuario y contraseña creados en MySQL para la autenticación, permitiendo la conexión segura mediante WSS y asegurando la protección de los datos transmitidos entre clientes y el servidor EMQX.

Figura 109

Verificación de conexión segura WSS.



Nota: En la imagen se puede observar la conexión segura con MQTT a través de WebSockets, en EMQX se otorga el estado de conectado. Fuente: Elaboración propia.

- **Instalación de Node.js y GIT**

Para la implementación del backend de la plataforma IoT, fue necesario instalar GIT y Node.js, herramientas esenciales para la gestión de versiones y la ejecución de aplicaciones basadas en JavaScript.

Figura 110

Instalación Node.js y GIT.

```
root@vps-servicio-medidor:~# sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package curl
root@vps-servicio-medidor:~#
root@vps-servicio-medidor:~# curl -fsSL https://deb.nodesource.com/setup_20.x |
sudo -E bash -
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://deb.nodesource.com/node_20.x nodistro InRelease [12.1 kB]
Get:6 https://deb.nodesource.com/node_20.x nodistro/main amd64 Packages [10.6 kB]
Fetched 22.7 kB in 1s (24.1 kB/s)
Reading package lists... Done
2025-03-11 21:26:31 - Repository configured successfully.
2025-03-11 21:26:31 - To install Node.js, run: apt-get install nodejs -y
2025-03-11 21:26:31 - You can use Njsolid Runtime as a node.js alternative
2025-03-11 21:26:31 - To install Njsolid Runtime, run: apt-get install nsolid -y
root@vps-servicio-medidor:~# apt update && apt install -y nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
nodejs
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 31.9 MB of archives.
After this operation, 108 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_20.x nodistro/main amd64 nodejs amd64 20.18.3-inodesource1 [31.9 MB]
Fetched 31.9 MB in 1s (30.6 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 93189 files and directories currently installed.)
Preparing to unpack .../nodejs_20.18.3-inodesource1_amd64.deb ...
Unpacking nodejs (20.18.3-inodesource1) ...
Setting up nodejs (20.18.3-inodesource1) ...
Processing triggers for man-db (2.9.11-1) ...
root@vps-servicio-medidor:~# node -v && npm -v
v20.18.3
10.8.2
root@vps-servicio-medidor:~#
```

Nota: Desde MobaXterm, se accedió al VPS y se ejecutaron los comandos de instalación. Luego, se verificó que ambas herramientas estuvieran correctamente configuradas mediante la consulta de sus respectivas versiones. Tomado de: Autoría propia.(22 febrero 2025).

- **Node-RED**

Es un entorno basado en flujos diseñado para integrar dispositivos IoT y facilitar la automatización de procesos. Para ello, se accedió al VPS como superusuario y se ejecutó la instalación, junto con un administrador de procesos para su gestión.(Figura 111).

Figura 111

Instalación de Node-red

```
root@vps-servicio-medidor:~# sudo su
root@vps-servicio-medidor:~# cd ~
root@vps-servicio-medidor:~#
root@vps-servicio-medidor:~# sudo npm install -g --unsafe-perm node-red
added 311 packages in 27s

63 packages are looking for funding
  run `npm fund` for details

npm notice
npm notice New major version of npm available! 10.8.2 -> 11.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.2.0
npm notice To update run: npm install -g npm@11.2.0
npm notice
```

Nota: Mediante MobaXterm ingresamos al VPS se instala Node-red. Tomado de: Autoría propia. (22 febrero 2025).

Inicialmente, se habilitó el acceso a Node-RED a través del puerto 1880 en el VPS, permitiendo la visualización del entorno desde el navegador.(Figuras 112-113) Luego, se procedió a configurar el acceso seguro mediante la creación de un usuario y contraseña.(Figura 114) El sistema generó un hash de clave, el cual fue añadido al archivo de configuración, asegurando la autenticación de los usuarios. (Figura 114).

Figura 112

Iniciar Node-red.

```
root@vps-servicio-medidor:~# node-red
Welcome to Node-RED
=====
11 Mar 22:00:14 - [info] Node-RED version: v4.0.9
11 Mar 22:00:14 - [info] Node.js version: v20.18.3
11 Mar 22:00:14 - [info] Linux 5.15.0-1078-gcp x64 LE
11 Mar 22:00:14 - [info] Loading palette nodes
11 Mar 22:00:15 - [info] Settings file : /root/.node-red/settings.js
11 Mar 22:00:15 - [info] Context store : 'default' [module=memory]
11 Mar 22:00:15 - [info] User directory : /root/.node-red
11 Mar 22:00:15 - [warn] Projects disabled : editorTheme.projects.enabled=false
11 Mar 22:00:15 - [info] Flows file : /root/.node-red/flows.json
11 Mar 22:00:15 - [info] Creating new flow file
11 Mar 22:00:15 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

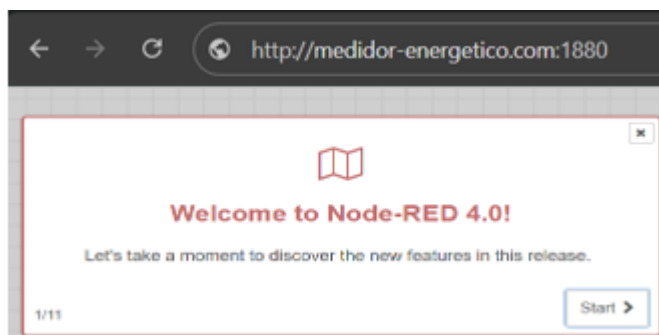
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

11 Mar 22:00:15 - [info] Server now running at http://127.0.0.1:1880/
11 Mar 22:00:15 - [warn] Encrypted credentials not found
11 Mar 22:00:15 - [info] Starting flows
11 Mar 22:00:15 - [info] Started flows
```

Nota: Dentro del VPS iniciamos Node-red. Lo que permite que al ir al navegador podamos acceder a Node-red. Tomado de: Autoría propia.(22 febrero 2025).

Figura 113

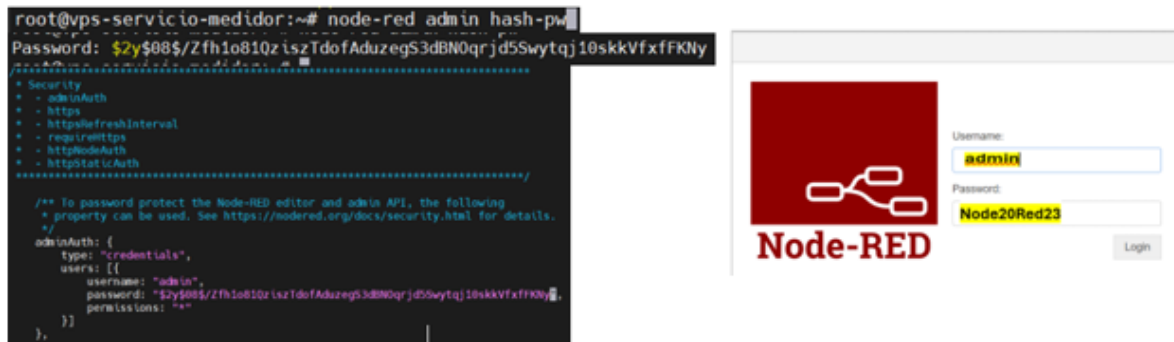
Ingresa a Node-red.



Nota: En esta imagen se ve el ingreso a Node-red, pero con una conexión segura. Tomado de Autoría propia. (22 febrero 2025).

Figura 114

Creación usuario y contraseña.

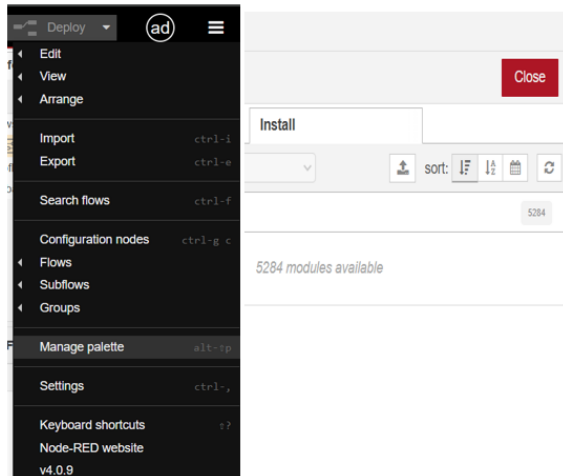


Nota: A través de Mobaxterm ingresamos un comando que nos va generar un código que permitirá modificar el archivo para poder autenticar usuario y contraseña. Tomado de: Autoría propia. (22 febrero 2025).

Se instalaron nodos adicionales para ampliar su funcionalidad con otros complementos.

Figura 115

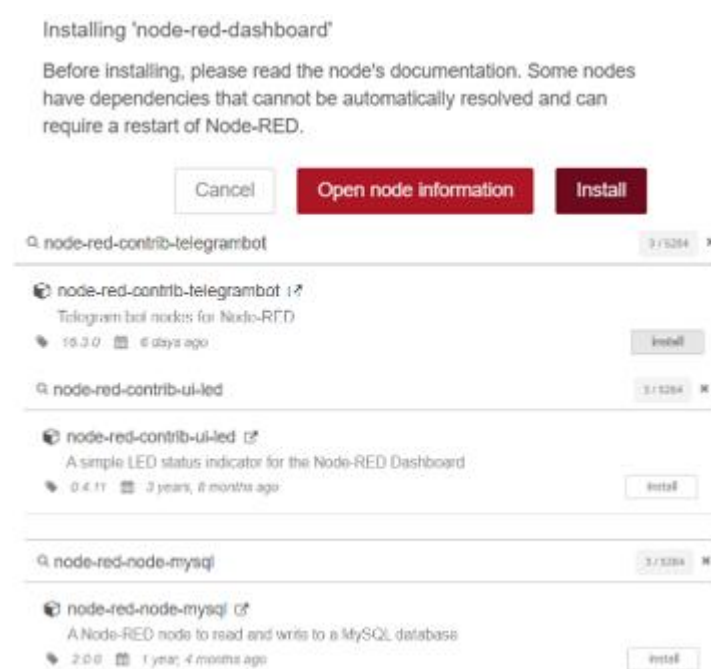
Acceso a menú para agregar nodos.



Nota: Nos colocamos sobre las 3 líneas en el lado derecho y a MANAGE PALETTE y vamos a INSTALL para agregar unos nodos. Tomado de: Autoría propia. (22 febrero 2025).

Figura 116

Instalar nodos.



Nota: Se instalaron nodos en Node-RED para ampliar su funcionalidad, incluyendo node-red-dashboard para la visualización de datos, integración con Telegram para mensajería, control de dispositivos mediante LED indicador y conexión con MySQL para la gestión de la base de datos. Tomado de: Autoría propia.(22 febrero 2025).

Finalmente, se habilitó el acceso seguro mediante certificados SSL/TLS, vinculando los archivos privkey.pem y fullchain.pem de Let's Encrypt al sistema.(Figuras 117-118) Tras reiniciar Node-RED, el acceso quedó configurado bajo HTTPS, asegurando la integridad de la comunicación.(Figura 119).

Figura 117

Ingreso al archivo de configuración.

```
clear
sudo su
cd ~
nano /root/.node-red/settings.js
```



Nota: Se ingresa como super usuario para poder modificar el archivo de configuración, se habilitan las líneas necesarias borrando los “*” Tomado de: Autoría propia.

Figura 118

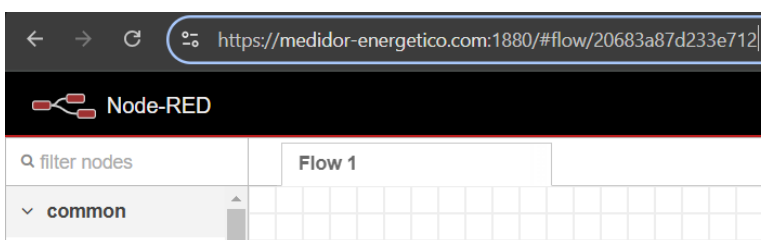
Certificado https.



Nota: En esta sección se realizaron las modificaciones necesarias para poder habilitar los certificados y tener acceso seguro en Node-red. Tomado de: Autoría propia.

Figura 119

Ingreso seguro Node-red.



Nota: Se pudo acceder de manera segura con el certificado https habilitado. Además, es necesario hacerlo a través dl puerto 1880. Tomado de: Autoría propia.

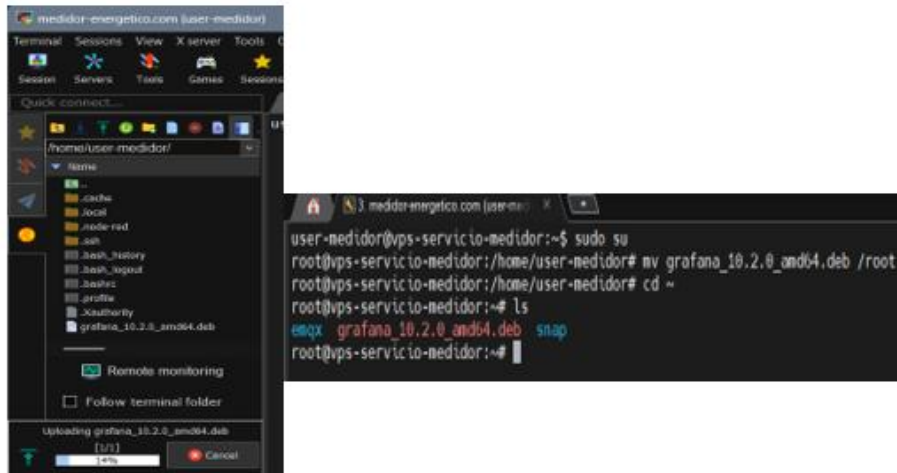
- **Instalación y Configuración de Grafana**

Se integró Grafana, una herramienta para la visualización y análisis de datos en tiempo real. El archivo de instalación fue transferido al VPS y, tras acceder como

superusuario, se movió a la carpeta raíz para proceder con su instalación.(Figuras 120-121).

Figura 120

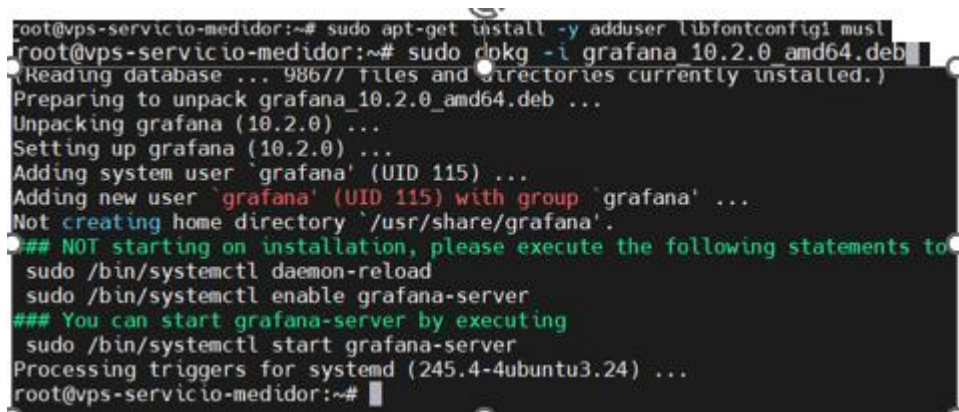
Agregar archivo de instalación a MobaXterm.



Nota: El instalador lo descargamos desde la página oficial, ese archivo lo arrastramos hacia la ventana de MobaXterm e ingresamos como super usuario y movemos el archi a la carpeta raíz para proceder con la instalación. Tomado de: Autoría propia.

Figura 121

Instalación de Grafana.

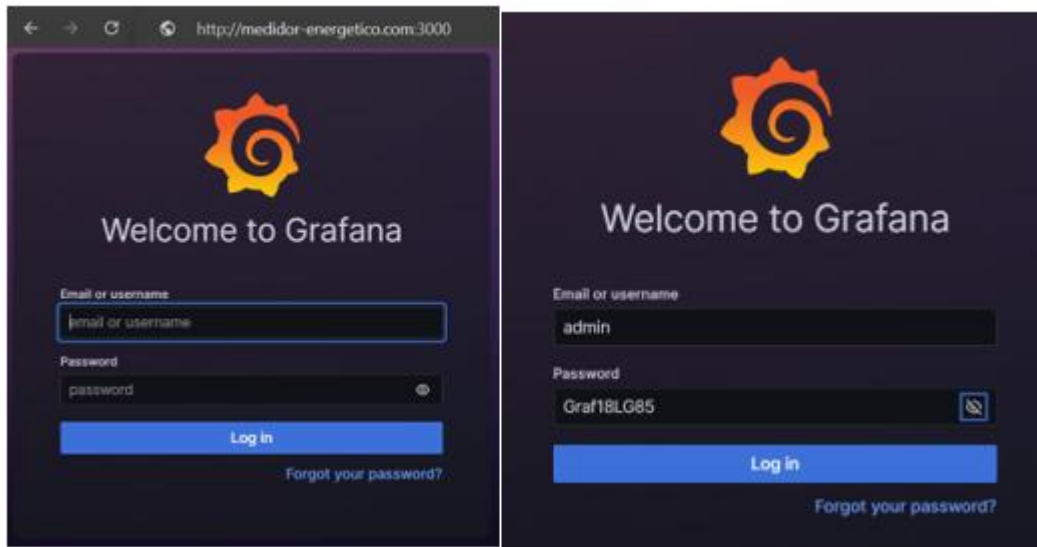


Nota: En la consola se ingresan los comandos para instalar la versión de Grafana requerida. Tomado de: Autoría propia.

Con el puerto 3000 habilitado en el servidor, se accedió a la interfaz web de Grafana mediante el dominio configurado, donde se utilizó la autenticación por defecto (usuario: admin / contraseña: admin), la cual fue modificada posteriormente por razones de seguridad.

Figura 122

Acceso Grafana por primera vez.



Nota: Cuando ingresamos por primera vez tanto Usuario y Password son “**admin**” pero nos solicita cambiar la contraseña y se procede con el cambio. Tomado de: Autoría propia.

- **Configuración de Certificados SSL en Grafana**

Para habilitar el acceso seguro a Grafana, se configuraron certificados SSL utilizando los archivos generados previamente por Let's Encrypt.

Figura 123

Modificación archivo grafana.ini.

```
grafana.ini
ldap.toml
provisioning
root@vps-servicio-medidor:/etc/grafana#
```

Nota: Este paso es el que permitió vincular nuestro dominio con Grafana. Tomado de: Autoría propia.

Los certificados fueron copiados desde la carpeta de Apache a la ubicación correspondiente dentro del sistema de Grafana, y se ajustaron los permisos de acceso.

Figura 124

Mover archivos desde apache a Grafana.

```
root@vps-servicio-medidor:/etc/grafana# cp /etc/letsencrypt/live/medidor-energetico.com/*.pem /etc/grafana/
root@vps-servicio-medidor:/etc/grafana# ls
cert.pem chain.pem fullchain.pem grafana.ini ldap.toml privkey.pem provisioning
root@vps-servicio-medidor:/etc/grafana#
root@vps-servicio-medidor:/etc/grafana# chown :grafana /etc/grafana/privkey.pem
root@vps-servicio-medidor:/etc/grafana# chmod 640 /etc/grafana/fullchain.pem
root@vps-servicio-medidor:/etc/grafana# chmod 640 /etc/grafana/privkey.pem
root@vps-servicio-medidor:/etc/grafana#
```

Nota: En la consola mediante las líneas de comando se movieron los archivos de certificados de apache a Grafana y se concedieron los permisos de acceso. Tomado de: Autoría propia.

Luego, se editó el archivo grafana.ini, habilitando HTTPS y definiendo las rutas correctas para los archivos fullchain.pem y privkey.pem.

Figura 125

Ingreso al archivo grafan.ini.

```
root@vps-servicio-medidor:/etc/grafana# sudo nano grafana.ini
GNU nano 4.8 grafana.ini
##### Grafana Configuration Example #####
#
# Everything has defaults so you only need to uncomment things you want to
# change
#
# possible values : production, development
;app_mode = production
#
# instance name, defaults to HOSTNAME environment variable value or hostname if HOSTNAME var is empty
;instance_name = ${HOSTNAME}
#
# force migration will run migrations that might cause dataloss
;force_migration = false
#
##### Paths #####
[paths]
# Path to where grafana can store temp files, sessions, and the sqlite3 db (if that is used)
;data = /var/lib/grafana
```

Nota: En este archivo se van habilitar ciertas líneas de comandos para habilitar los cambios requeridos. Tomado de: Autoría propia.

Figura 126

Habilitar protocolo https.

```
[server]
# Protocol (http, https, h2, socket)
;protocol = http

[server]
# Protocol (http, https, h2, socket)
protocol = https
```

Nota: En la sección de SERVER eliminamos “;” para poder editar y habilitar el protocolo https.
Tomado de: Autoría propia.

Figura 127

Establecer rutas para los archivos.

```
https certs & key file
;cert_file =
;cert_key =

# https certs & key file
cert_file = /etc/grafana/fullchain.pem
cert_key = /etc/grafana/privkey.pem
```

Nota: Ubicamos los certificados y llaves de seguridad en las rutas adecuadas y que el proceso no tenga ningún error. Tomado de: Autoría propia.

Una vez aplicados los cambios, se reinició Grafana (Figura 128) y se validó el acceso seguro mediante HTTPS, asegurando la protección de la comunicación con la plataforma.

Figura 128

Reinicio Grafana.

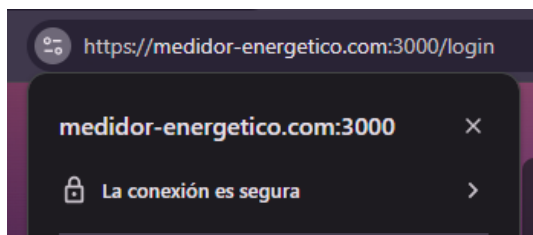
```
sudo service grafana-server restart
sudo service grafana-server status

● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-03-12 03:51:48 UTC; 8s ago
     Docs: http://docs.grafana.org
    Main PID: 4982 (grafana)
      Tasks: 7 (limit: 2342)
    Memory: 87.1M
    CGroup: /system.slice/grafana-server.service
            └─4982 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:dev
```

Nota: Para que los cambios se hagan efectivos se reinicia Grafana y se verifica el estado.
Tomado de: Autoría propia.

Figura 129

Verificar acceso seguro.



Nota: Se pudo confirmar que el acceso a Grafana es seguro mediante los certificados http. Tomado de: Autoría propia.

- **ARDUINO CON MQTT**

El protocolo MQTT (Message Queuing Telemetry Transport) es una herramienta clave en el Internet de las Cosas (IoT), permitiendo la comunicación eficiente entre dispositivos. En este caso, el ESP32 se conecta al broker EMQX utilizando Arduino IDE, estableciendo una red WiFi con credenciales definidas y configurando parámetros de MQTT como el servidor `mqtt_user = "web_iot_cliente"` que sería el usuario y `mqtt_pass = "web1234*****"`, que sería la contraseña. A través de la suscripción y publicación de mensajes en los tópicos "MyCasa/DataOut" y "MyCasa/DataIn", el ESP32 puede enviar y recibir información en tiempo real, en este caso por la simulación de valores de voltaje, corriente y potencia. La conexión al broker se mantiene activa mediante una función de reconexión, asegurando la transmisión continua de datos.

Figura 130

Código ejemplo para Arduino IDE – ESP32-EMQX.

```
const char* ssid = "dlink-E0EC";
const char* password = "okxfq887066";
// const char *mqtt_server = "SuDominio.com";
const char *mqtt_server = "medidor-energetico.com";
```

☐	Nombre	Tipo	Destinos	Filtros	Protocolos/puertos
☐	puertos-habilitados	Entrada	Aplicar a	Intervalos	tcp:22, 53, 80, 443, 1880, 1883,

```
const int mqtt_port = 1883;
const char *mqtt_user = "web_iot_cliente";
const char *mqtt_pass = "web1234abcd";
//
const String Topic_Init = "MyCasa";
const String DataOut = "/DataOut";
const String DataIn = "/DataIn";

if (!client.connected()) {
    reconnect();
}

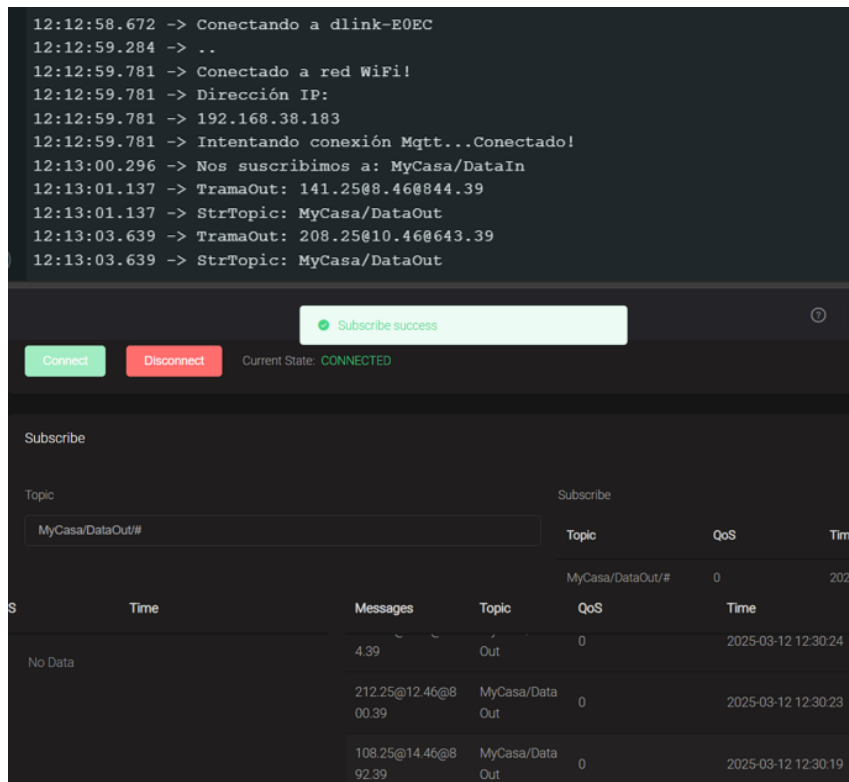
client.loop();
```

Nota: En esta imagen podemos observar las líneas de código importantes, además de la habilitación del puerto 1883 para la comunicación entre la ESP32 y EMQX. Elaboración propia.

El código implementa una lógica sencilla pero efectiva para manejar la conexión WiFi, configurar el cliente MQTT y gestionar la comunicación bidireccional. Cada 2.5 segundos, el ESP32 genera valores aleatorios que representan mediciones eléctricas, los empaqueta en una cadena y los envía al tópico correspondiente. Por otro lado, cuando el ESP32 recibe un mensaje a través del tópico "MyCasa/DataIn", interpreta el comando y responde adecuadamente, esta información puede ser visualizada en la interfaz del EMQX ingresando los datos de información a la suscripción del topic correspondiente.

Figura 131

Conexión de ESP 32, Topic, Visualización de datos.

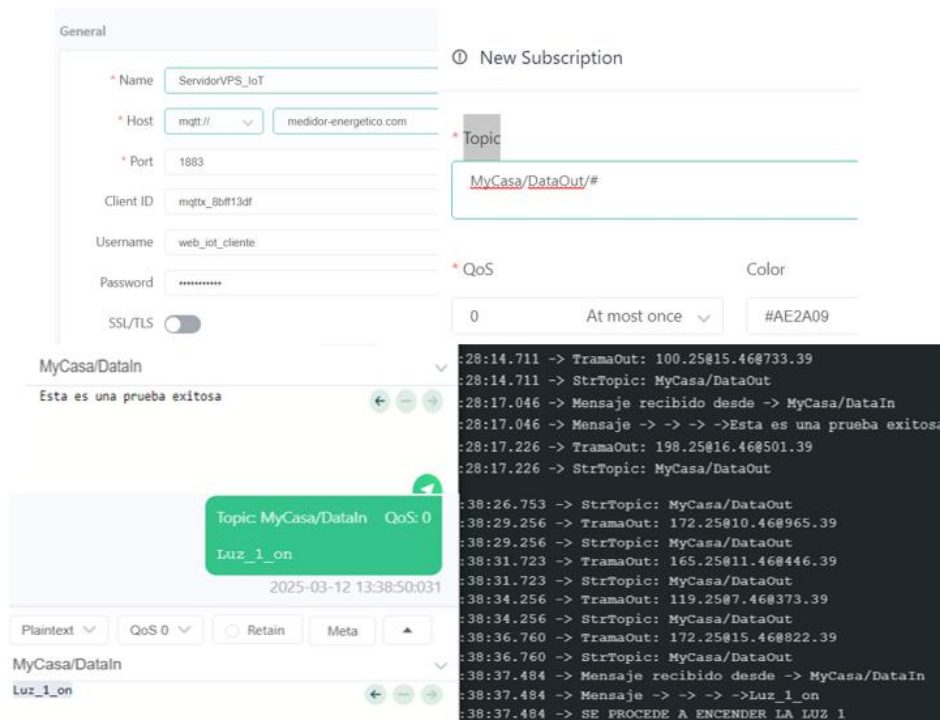


Nota: En esta imagen podemos observar el monitor serial, la conexión de la ESP32, el StrTopic y la visualización de datos en la página web EMQX. Elaboración propia.

Finalmente, la interacción con EMQX mediante la aplicación MQTTX permite monitorear y verificar la comunicación desde una interfaz más amigable, facilitando la supervisión de los datos enviados y recibidos. Al suscribirse a los tópicos adecuados, es posible visualizar los valores generados en el ESP32 y, a su vez, enviar comandos de prueba al microcontrolador. Este enfoque ofrece una base sólida para desarrollar aplicaciones más avanzadas de monitoreo y automatización en entornos conectados, optimizando la gestión energética y el control de dispositivos en tiempo real.

Figura 132

Interacción entre MQTTX y Arduino.



Nota: En esta imagen se puede observar la interacción entre la aplicación MQTTX y el monitor serial, la visualización de los datos y el envío de datos, se puede enviar Elaboración propia.

4.2.5. Etapa5: Integración del Prototipo de medición con la Pagina Web

En esta tapa se abordará la integración de los 4 prototipos IoT, que se encargan de medir parámetros como voltaje, corriente, potencia y energía además de estimar un valor en dólares del consumo energético que total, para esto se debe tener en cuenta los siguientes puntos.

- **Modificación de Topic**

Para la integración de los 4 dispositivos con la página web IoT se modifica en el código los siguientes parámetros para cada casa las credenciales de la red como el usuario u contraseña wifi de cada casa, además del Topic_Out que sería mi publicador y el Topic_In que sería mi suscriptor en el apartado /casa #/ se coloca en número de prototipo esto para que se pueda identificar en MQTT Explorer.

Figura 133

Configuración de código Prototipos IoT.

```
const char* ssid      = "CELERITY_NO_MAS_GRATIS"; // WIFI  USUARIO
const char* password  = "1715998231"; // WIFI  CONTRASEÑA

// *****
// ***   CONFIGURACION MQTT   ***
// *****
const char *mqtt_server = "monitor-casas.com";
const int  mqtt_port  = 1883;
const char *mqtt_user  = "my_user_mqtt_casas";
const char *mqtt_pass  = "rasp_casas_mqtt_2024";
// *****
const String Topic_Out = "Monitoreo/Casa_2/Out"; // CAMBIO NUMERO DE CASA
const String Topic_In  = "Monitoreo/Casa_2/In";  // CAMBIO NUMERO DE CASA
```

Nota: En esta imagen se puede observar el cambio de las credenciales Wifi y el Topic publicador y suscriptor en el apartado casa según el número de prototipo. Elaboración propia.

- **Conexión de Prototipos**

Con los la configuración del código para cada prototipo, se conectan los 4 a la red eléctrica otorgándoles la alimentación para el funcionamiento de la placa electrónica que alberga la ESP32, inicializando la misma para que se conecte a la red wifi y comience a enviar datos, para verificar que los 4 prototipos se encuentran enviando datos se utilizó MQTT Explorer donde se visualizó la vinculación de cada uno.

Figura 134

Prototipos Conectados.



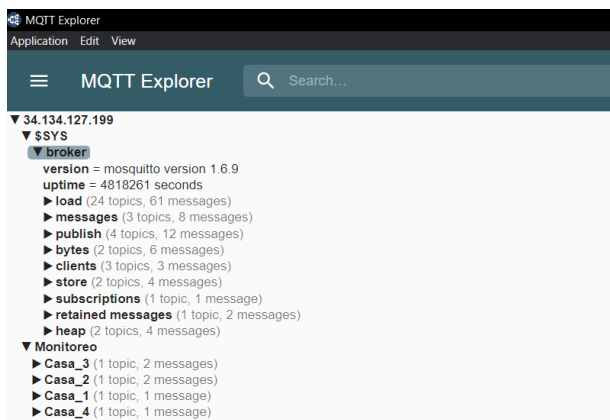
Nota: En esta imagen se puede observar los 4 prototipos conectados y en funcionamiento. Elaboración propia.

- **Identificación de prototipos**

Para saber si los prototipos se encuentran enviando información es decir comunicando los datos obtenidos por el sensor Pzem-004T al broker se hace uso de la aplicación MQTT Explorer la cual permite conectarse a un broker MQTT para enviar, recibir y monitorear mensajes en los topics.

Figura 135

MQTT Explorer.



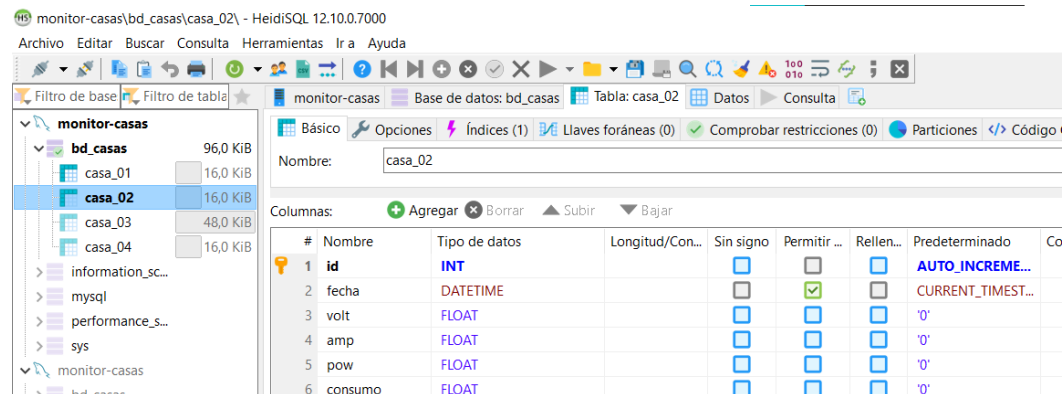
Nota: En esta imagen se observa el monitoreo de cada prototipo (Casa_1, Casa_2 y Casa_3), se muestra la conexión a un servidor con la IP 34.134.127.199. Elaboración propia.

- **Tablas de bases de datos**

Previamente se revisa en HeidiSQL, que cada tabla para cada casa fue creada en la base de datos, con los datos que esta va albergar como son, el nombre, la fecha, el voltaje, la corriente, la potencia y el consumo. Esto se ejecuta para permitir que node red envíe la información a la tabla correspondiente.

Figura 136

Tablas creadas en Base de Datos.



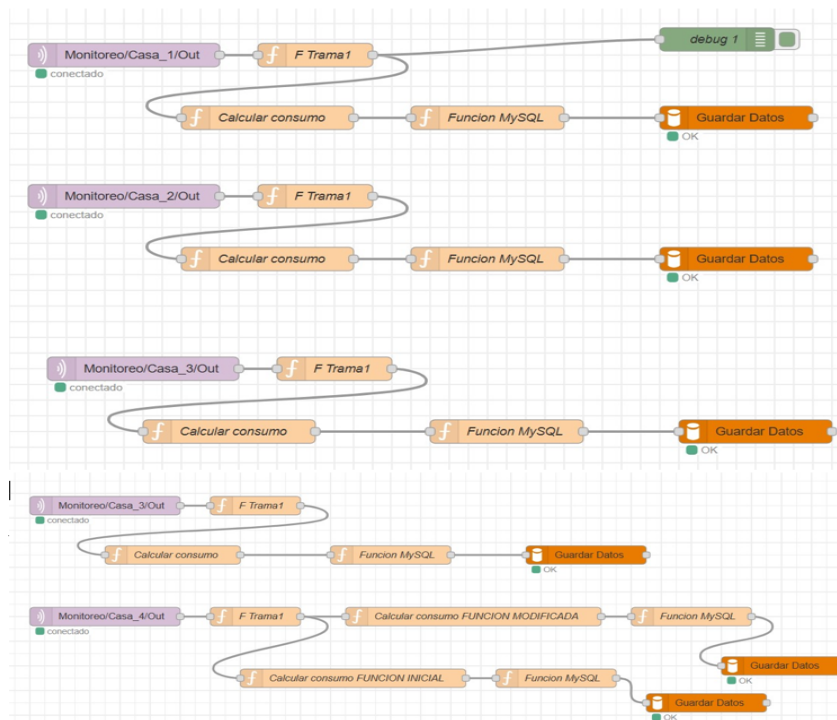
Nota: En esta imagen se puede observar las tablas creadas en la base de datos, cada tabla lleva su número correspondiente y sus tipos de datos. Elaboración propia.

- **Node red**

En este caso mediante Node Red que es una herramienta visual que permite la interconexión de varios sistemas, se ha diseñado un flujo para la recopilación, procesamiento y almacenamiento de datos de monitoreo energético utilizando MQTT y MySQL. Comenzando por el nodo "Monitoreo/Casa_1/Out" el cual recibe los datos desde un dispositivo IoT (ESP32), suscribiéndose a un tema MQTT. Luego, los datos pasan por "F Trama1", que filtra la información. A continuación, la información se bifurca: una parte va directamente a "Calcular consumo", donde se procesa el consumo energético, y la otra sigue su flujo a "Función MySQL", que estructura la información para su almacenamiento en la base de datos. Finalmente, los datos llegan a "Guardar Datos", donde se insertan en MySQL, y también se envían a un nodo "debug 1" para visualizar el resultado y verificar que el proceso se ejecuta correctamente.

Figura 137

Node Red, Nodos de Casa 1 a Casa 4.



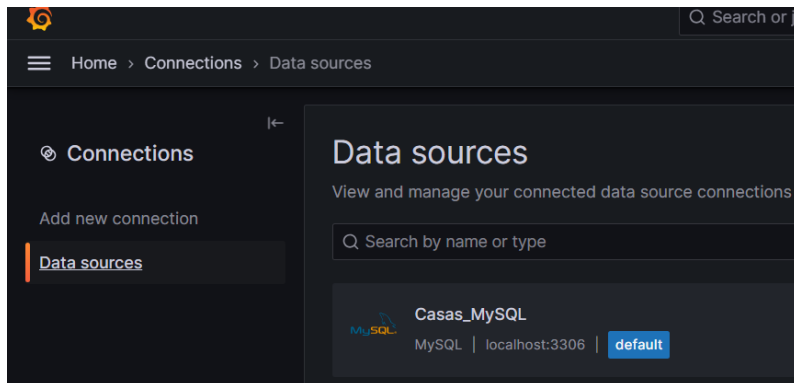
Nota: En la imagen se puede observar el flujo de procesamiento de datos en Node-RED para el monitoreo y registro de consumo de energía, el flujo comienza con la recuperación y procesamiento de datos (voltaje, amperaje y fecha) en el primer nodo de tipo *function*. Luego, se calculan la potencia y el consumo de energía en un segundo nodo *function*. Finalmente, los datos procesados son insertados en una base de datos MySQL mediante un nodo de tipo *mysql*. Elaboración propia.

- **Integración con Grafana**

Para configurar una fuente de datos en Grafana, primero se debe acceder a la plataforma a través de la URL correspondiente, como <http://localhost:3000>, e iniciar sesión con las credenciales de administrador. Luego, en el panel de control, se debe ir a la sección "Connections" → "Data sources", donde se añade una nueva fuente de datos seleccionando MySQL. A continuación, se configuran los parámetros de conexión, ingresando el host (localhost:3306 o la dirección IP del servidor), el nombre de la base de datos (Casas_MySQL), y las credenciales de usuario y contraseña. Finalmente, se guarda la configuración y se prueba la conexión mediante la opción "Save & Test".

Figura 138

Conexión de Base de Datos con Grafana.

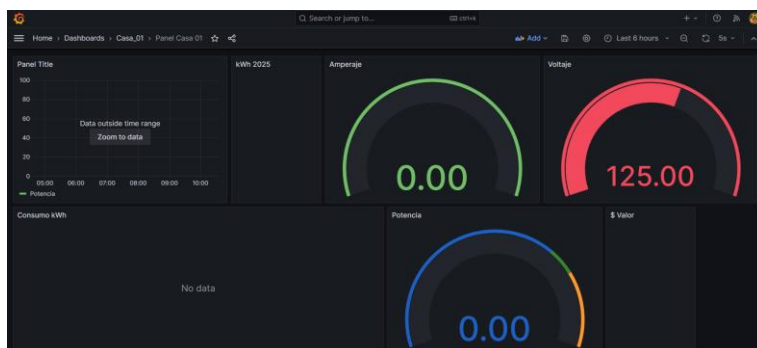


Nota: En esta imagen se puede observar la base de datos añadida en grafana la cual es Casas_MySQL. Elaboración propia.

Una vez conectada la base de datos, se procede a la creación del dashboard principal. Para ello, se debe ir a "Dashboards" → "Manage", hacer clic en "New Dashboard", y luego seleccionar "Add a new panel". En esta sección, se elige MySQL como fuente de datos y se escribe una consulta SQL para extraer los datos en tiempo real, se puede usar la siguiente consulta: `SELECT voltage, current, power, timestamp FROM Casa_01 WHERE timestamp >= NOW() - INTERVAL 1 DAY;` Luego, se configuro el tipo de visualización, en forma de gauge, gráfico de líneas, y se ajustan las leyendas, unidades y el rango de tiempo. Finalmente, se aplican los cambios guardando el panel dentro del dashboard.

Figura 139

Creación de Dashboards.

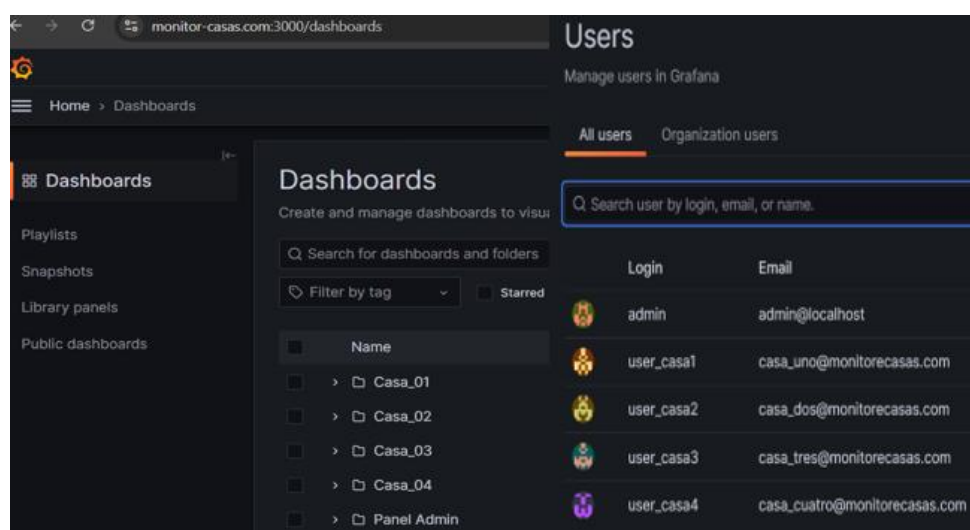


Nota: En esta imagen se puede observar el panel principal creado, con sus paneles de visualización en los que se encuentran el voltaje, la corriente, la potencia, el consumo energético y un panel para la aproximación del valor en dólares. Elaboración propia.

Se replicó el dashboard principal para las demás casas, agregando una nueva carpeta que contenga cada dashboard en cada una de las casas, se copia y pega el Json model del panel principal a los de las demás casas configurando el número de casa, el número de panel y el UID el cual es un número de identificación único para cada panel. Finalmente se crearon los usuarios y contraseñas para el ingreso a cada panel de cada uno de los prototipos.

Figura 140

Creación de Dashboards y Usuarios.



Nota: En esta imagen se puede observar los paneles y usuarios creados para cada casa. Elaboración propia.

4.3. Pruebas

Para garantizar el correcto funcionamiento del sistema IoT y su impacto en la medición del consumo eléctrico, se realizaron diversas pruebas en diferentes etapas. Estas pruebas evaluarán la precisión del sistema, su facilidad de uso y la satisfacción de los usuarios.

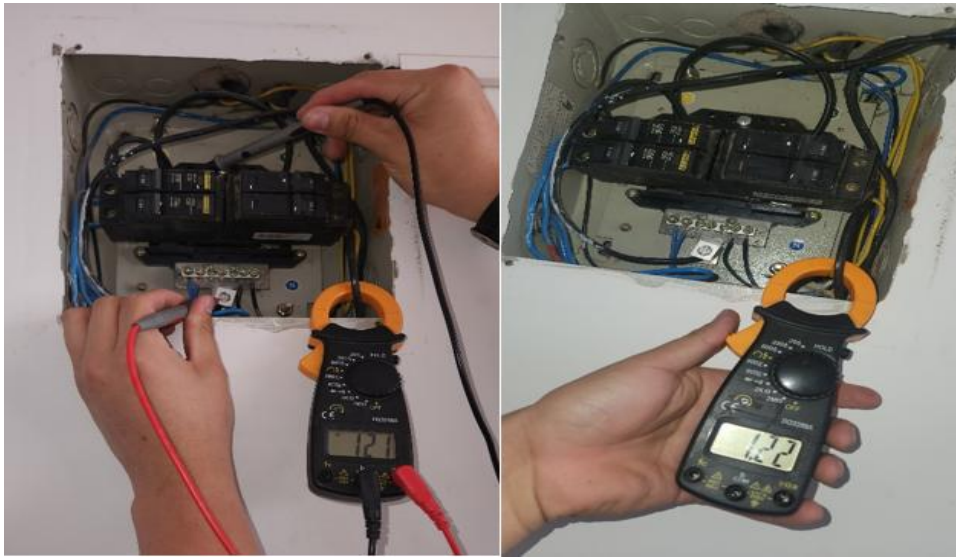
4.3.1. Etapa1: Verificación de acceso correcto a la plataforma.

- **Conexión de prototipos.**

Los cuatro prototipos IoT se instalan en diferentes viviendas, seleccionando un área adecuada cercana a la caja de breakers. Para ello, se retira la tapa de protección y se identifican la fase y el neutro. Se mide con el multímetro en la escala de voltaje alterno se mide el voltaje entre la fase y el neutro, por otro lado, con una pinza amperimétrica colocada en la fase se mide la corriente.

Figura 141

Medición de Variables eléctricas.

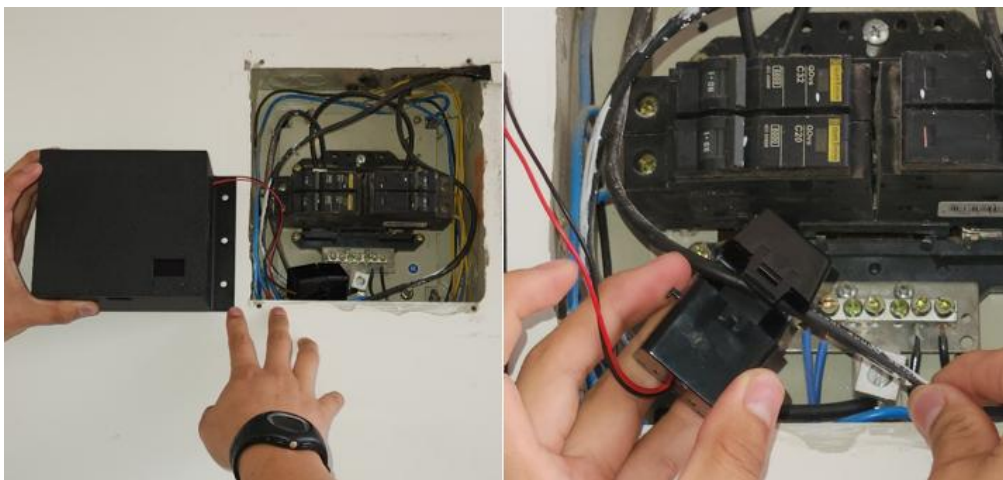


Nota: En la imagen podemos observar la medición del voltaje otorgándonos una lectura de 122 V AC, mientras que la corriente una lectura de 1.23 A. Fuente: Elaboración Propia .

Se utiliza la pinza amperimétrica en la fase de la caja de distribución para determinar la distancia óptima a la que debe empotrarse o asegurarse el prototipo. Una vez colocado cerca de la caja de breakers.

Figura 142

Instalación de Prototipo IoT y colocación de pinza amperimétrica.

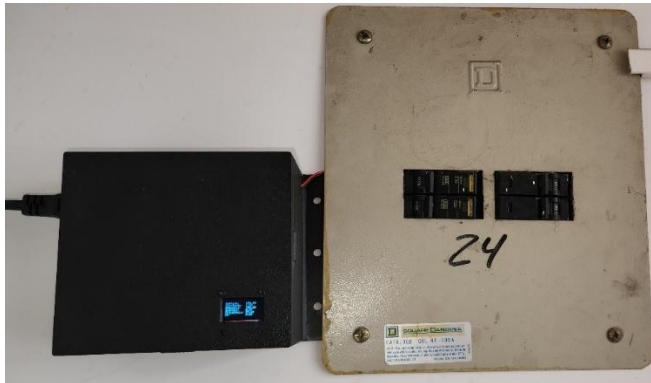


Nota: En la imagen podemos observar la medición del voltaje otorgándonos una lectura de 122 V AC, mientras que la corriente una lectura de 1.23 A. Fuente: Elaboración Propia.

Finalmente se procede a alimentación el prototipo IoT, verificando que inicie correctamente y muestre la pantalla de inicio, seguida de los valores de medición en la pantalla OLED de 1.3". Este procedimiento se repite para cada prototipo instalado.

Figura 143

Funcionamiento del Prototipo IoT.

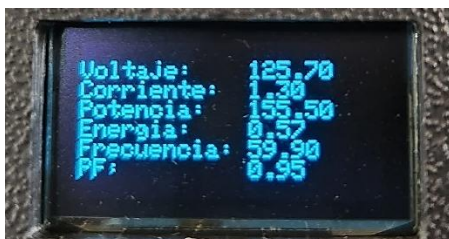


Nota: En la imagen se puede observar la el prototipo correctamente instalado además de la tapa de protección para la caja de breakers. Fuente: Elaboración Propia.

Con el prototipo correctamente instalado y funcionando con la red wifi correspondiente, el oled de 1.3 pulgadas nos permite visualizar los parámetros de medidos como son el voltaje, la corriente, la potencia, la energía consumida, la frecuencia y el factor de potencia de la vivienda.

Figura 144

Mediciones en Oled de 1.3"



Nota: En la imagen se puede observar la el prototipo nos entrega las lecturas de Voltaje 126. V AC, la corriente de 1.30 A, la potencia de 156 W, la energía de 0.57 kWh, la frecuencia de 59.90 Hz y el factor de potencia de 1. Fuente: Elaboración Propia.

- **Acceso a la plataforma Web IoT.**

Se verifica el acceso a la plataforma Web IoT a través del dominio <https://monitor-casas.com/>. Al ingresar, se identifica la caratula principal y se hace clic para acceder al panel de inicio de sesión, donde cada usuario introduce su nombre de usuario y contraseña según el número de su vivienda. Una vez dentro de la interfaz de Grafana, se navega por la plataforma identificando la pestaña de dashboard y seleccionando el dashboard principal de la casa. Finalmente, se puede visualizar en tiempo real los datos enviados por el prototipo IoT.

Figura 145

Monitoreo en Grafana.



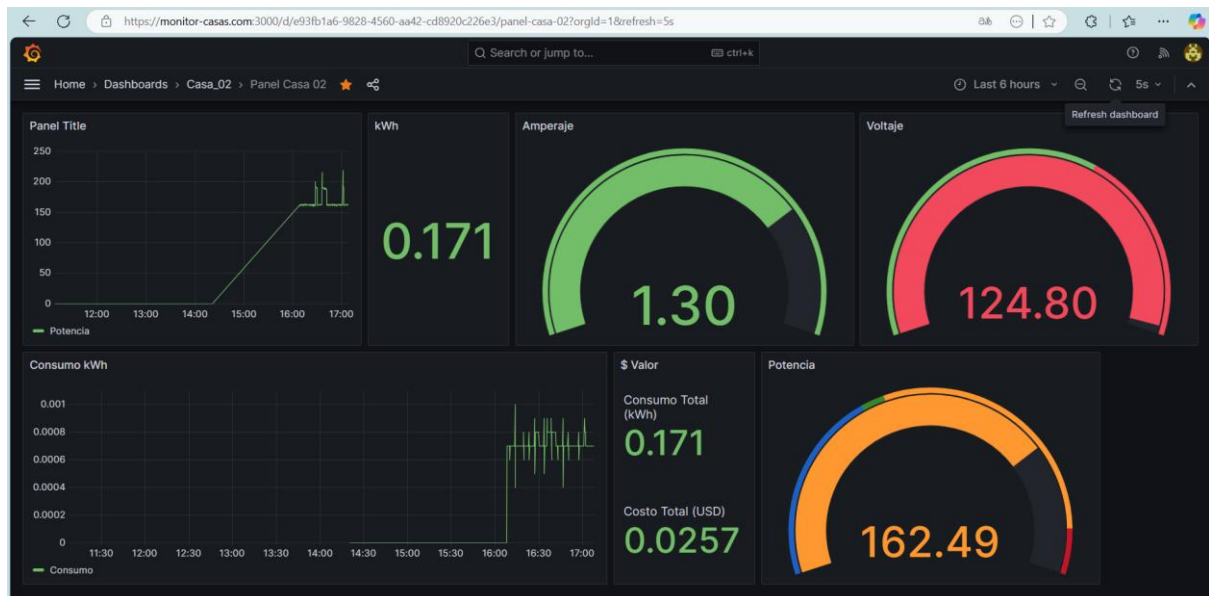
Nota: Caratula de inicio de la página de la página web IoT. Fuente: Elaboración Propia.

- **Visualización de los datos en tiempo real**

La visualización de los datos de cada prototipo varía según las condiciones eléctricas de cada vivienda. Estos parámetros son presentados en el panel principal de cada casa mediante Grafana, donde se pueden observar valores de corriente, voltaje, potencia y consumo eléctrico. Además, se muestra un valor aproximado del consumo eléctrico en dólares. Cada widget proporciona una visualización general del valor medido en tiempo real, con una actualización de los datos cada 15 segundos. Esto permite monitorear el comportamiento energético de cada hogar de manera continua y detallada.

Figura 146

Ingreso a panel principal de Casa número 2.



Nota: Panel principal casa número 2 en este se puede visualizar los parámetros medidos, además 2 paneles muestran graficas del consumo y la energía eléctrica consumida en función al tiempo transcurrido. Fuente: Elaboración Propia.

4.3.2. Etapa2: Inducción sobre el manejo de la plataforma

- **Explicación del acceso y navegación en la interfaz web**

A cada usuario se le realizó una inducción detallada, en la cual se explicaron los pasos necesarios para acceder a la plataforma web de manera correcta y segura. Durante esta sesión, se brindaron instrucciones precisas sobre el procedimiento de inicio de sesión, la navegación dentro del sistema y el uso de las diferentes funciones disponibles. Además, se enfatizó la importancia de la seguridad en el manejo de credenciales, recomendando a cada usuario mantener en confidencialidad su información de acceso. Como parte de este proceso, se asignaron usuarios y contraseñas personalizadas para cada persona, garantizando así un acceso seguro y controlado. A continuación, se detallarán los pasos específicos para ingresar a la plataforma y utilizar sus herramientas de manera óptima.

Ingreso al dominio mediante cualquier navegador

Para ingresar al dominio se escribe en el buscador la siguiente dirección <https://monitor-casas.com/> la cual le muestra una caratula y se da clic en la misma para acceder al pestana de credenciales.

Figura 147

Inducción al manejo de la página Web IoT.



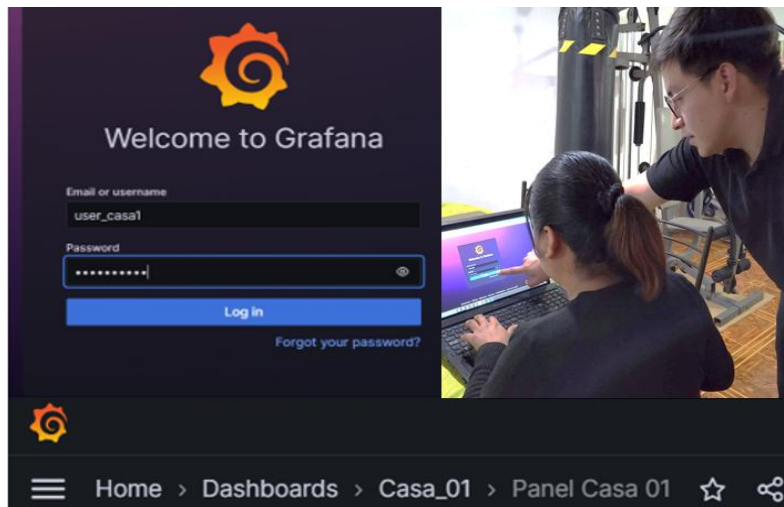
Nota: Ingreso a dominio por medio de computadora, identificación de caratula y pasos a seguir
Elaboración Propia.

Colocación de usuario y contraseña

Una vez en la pestaña de ingreso a grafana el usuario colocara su usuario y contraseña para acceder al interfaz de grafana. Una vez adentro el usuario seguirá una secuencia sencilla en la debe identificar la pestaña Dashboards y dar clic, seguidamente seleccionar su Casa y su panel de visualización.

Figura 148

Ingreso de credenciales por parte del usuario.



Nota: Ingreso de credenciales por parte del usuario además de secuencia de para abrir el panel principal. Fuente: Elaboración Propia.

El usuario se encuentra dentro de la interfas y podra vizualir el siguiente panel con las mediciones otorgadas por el prototipo IoT en la pagina Web mediante Grafana.

Figura 149

Visualización de Panel principal por parte de usuario



Nota: Visualización e interacción con el panel principal por parte del usuario, monitore de parámetros de otorgados por el prototipo IoT. Fuente: Elaboración Propia.

- **Interpretación de gráficos y datos de consumo eléctrico**

Este panel de monitoreo permite al usuario visualizar en tiempo real los valores eléctricos clave de su hogar, incluyendo voltaje, amperaje, potencia y consumo de energía en kWh. A través de gráficos e indicadores, el usuario puede observar el comportamiento del consumo energético, identificando posibles variaciones o anomalías. En la parte superior, se muestra el voltaje actual, mientras que en el centro se encuentran las mediciones de amperaje y potencia, reflejando el flujo de electricidad utilizado. Además, el consumo acumulado en kWh y su costo estimado en dólares facilitan un control más eficiente del gasto energético. La interfaz dinámica y actualizable cada poco segundo permite un monitoreo preciso, brindando al usuario la posibilidad de tomar decisiones informadas para optimizar su consumo eléctrico.

Figura 150

Interpretación de parámetros y graficas de W y kWh.



Nota: Explicación de tablas de gráficos mostradas en el panel principal al usuario, se distingue como la potencia y el consumo eléctrico en kWh va fluctuando en función al tiempo. Fuente: Elaboración Propia.

El panel de monitoreo de consumo eléctrico en tiempo real de la "Casa_02", presenta diferentes indicadores y gráficos que permiten al usuario analizar el comportamiento del sistema eléctrico.

1. **Voltaje (124.80 V):** Representa el nivel de tensión suministrado al hogar. Se muestra en un medidor tipo gauge con una escala de colores para visualizar si está dentro de un rango adecuado.
2. **Amperaje (1.30 A):** Indica la corriente eléctrica que fluye en el sistema. Un mayor amperaje sugiere un mayor consumo de energía.
3. **Potencia (162.49 W):** Es el resultado del producto del voltaje y el amperaje. Se muestra en un medidor semicircular con una escala de colores para indicar niveles de consumo.
4. **Consumo en kWh (0.171 kWh):** Refleja la energía total consumida en el período analizado.
5. **Costo total (0.0257 USD):** Muestra el costo acumulado en dólares basado en el consumo energético registrado.
6. **Gráficos de tendencia:**
 - El gráfico superior izquierdo muestra la evolución de la potencia a lo largo del tiempo.
 - El gráfico inferior izquierdo detalla el consumo de energía en kWh con el paso de las horas, permitiendo identificar patrones de uso.

Figura 151

Descripción al Usuario.



Nota: En conjunto, este panel ofrece al usuario una visión detallada y actualizada de su consumo energético. Fuente: Elaboración Propia.

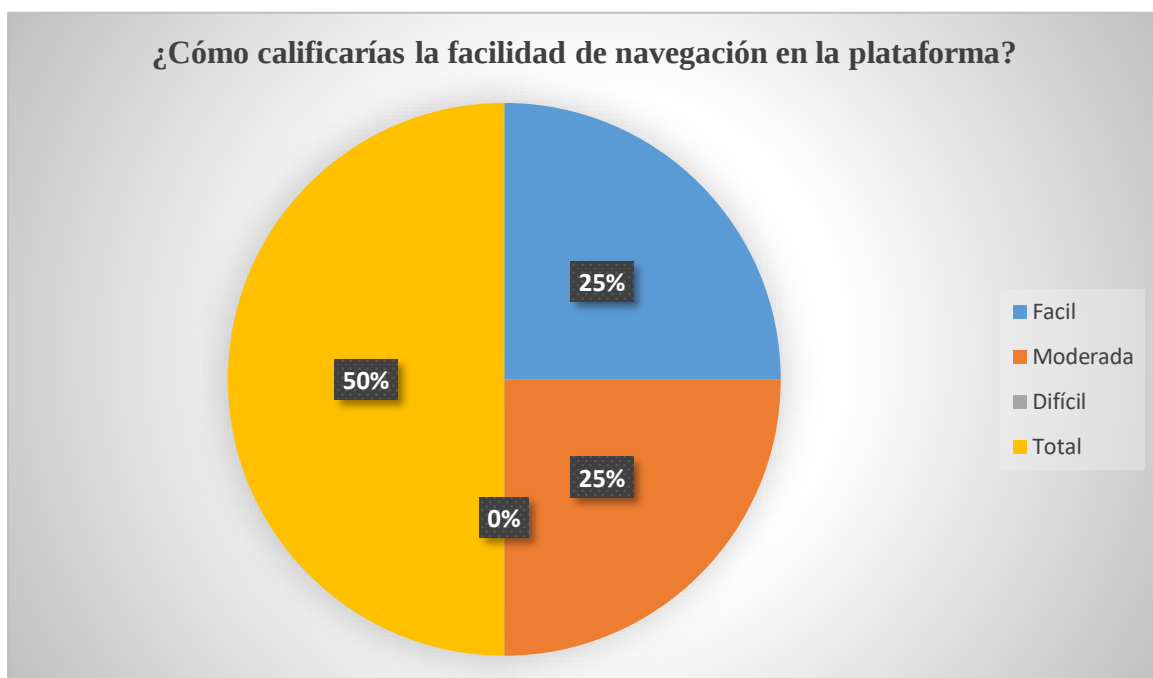
4.3.3. Etapa3: Retroalimentación por parte de los usuarios (entorno de la plataforma satisfacción con la información obtenida).

- **Encuestas de sobre la facilidad de uso y utilidad de la plataforma.**

Se realizó una encuesta compuesta por 4 preguntas a los 4 usuarios, con el fin de identificar aspectos clave sobre la interfaz de la página web, la utilidad de la información mostrada y posibles áreas de mejora. Las respuestas permitieron evaluar la percepción de los usuarios en cuanto a la facilidad de uso y la efectividad de los datos proporcionados, además de recopilar sugerencias que podrían contribuir a futuras actualizaciones y mejoras en la plataforma.

Figura 152

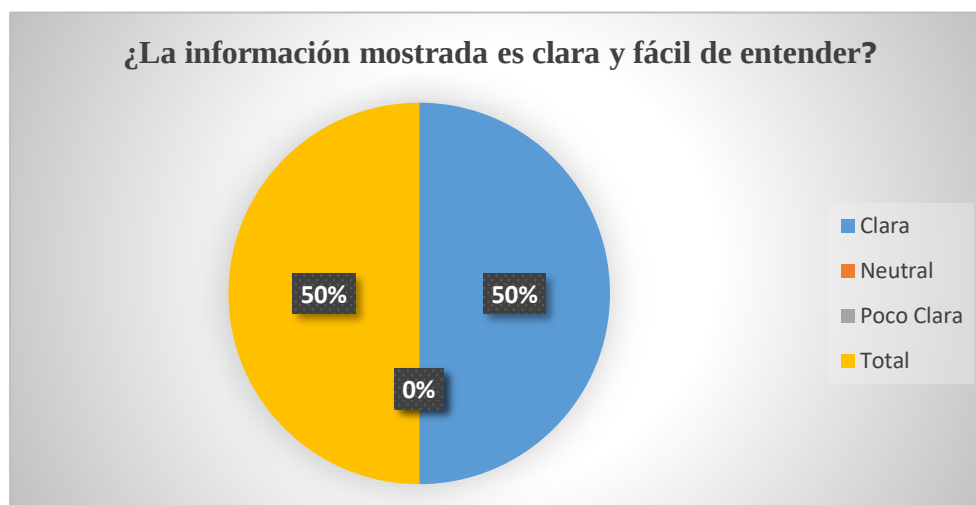
Pregunta # 1



Nota: El 50 % de los usuarios calificó la navegación como "Fácil", mientras que el otro 50 % la consideró "Moderada". Esto indica que la interfaz de la plataforma es accesible, pero algunos usuarios perciben cierta dificultad en su uso. Fuente: Elaboración Propia.

Figura 153

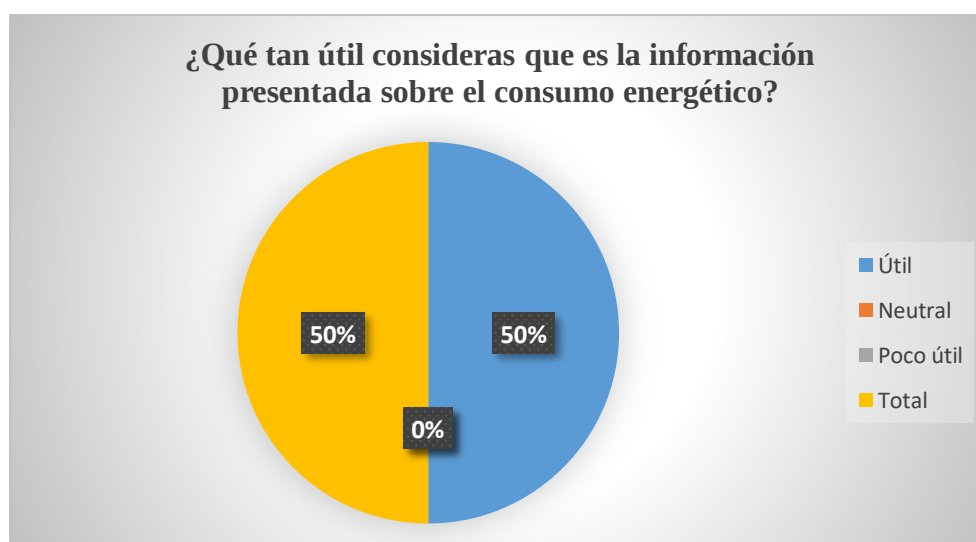
Pregunta #2



Nota: El 100 % de los encuestados opinó que la información mostrada es clara y fácil de entender. Esto refleja que la plataforma presenta los datos de manera intuitiva y comprensible para los usuarios. Elaboración Propia.

Figura 154

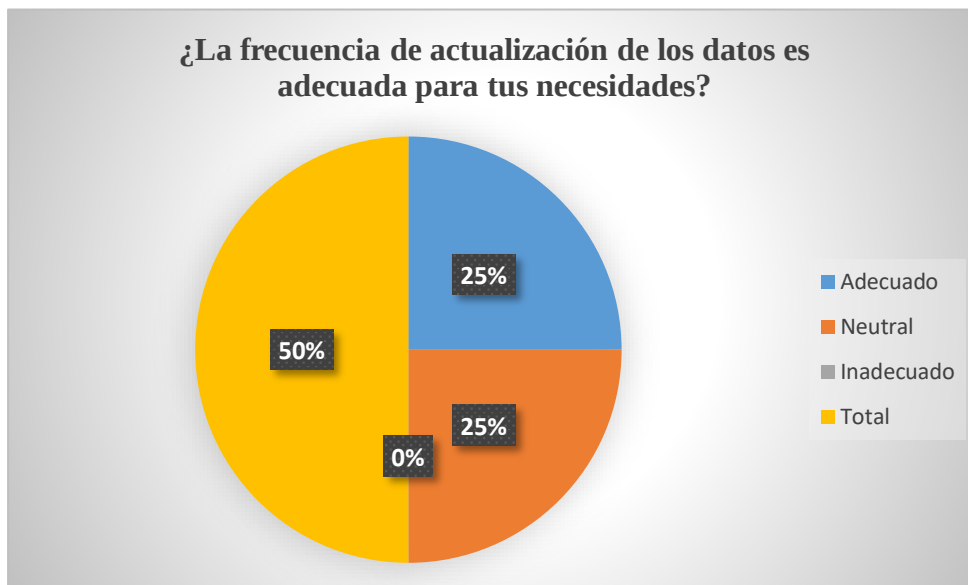
Pregunta #3



Nota: Fuente: El 100 % de los participantes consideró que la información sobre el consumo energético es útil. Esto sugiere que los datos proporcionados cumplen con las expectativas y necesidades de los usuarios. Elaboración Propia.

Figura 155

Pregunta #4



Nota: Fuente: El 50 % de los usuarios consideró que la actualización de los datos es "Adecuada", mientras que el otro 50 % la calificó como "Neutral". Esto sugiere que, aunque la frecuencia es aceptable, podría optimizarse para mejorar la experiencia del usuario. Elaboración Propia.

Figura 156

Pregunta # 5



Nota: Fuente: E El 100 % de los encuestados expresó una satisfacción "Satisfecho" con la plataforma. Esto indica que los usuarios tienen una percepción positiva del sistema y que cumple con sus expectativas en cuanto a funcionalidad y usabilidad. Fuente: Elaboración Propia.

4.3.4. Etapa4: Evaluación de Datos visualizados en los Dashboards de la Plataforma Web IoT.

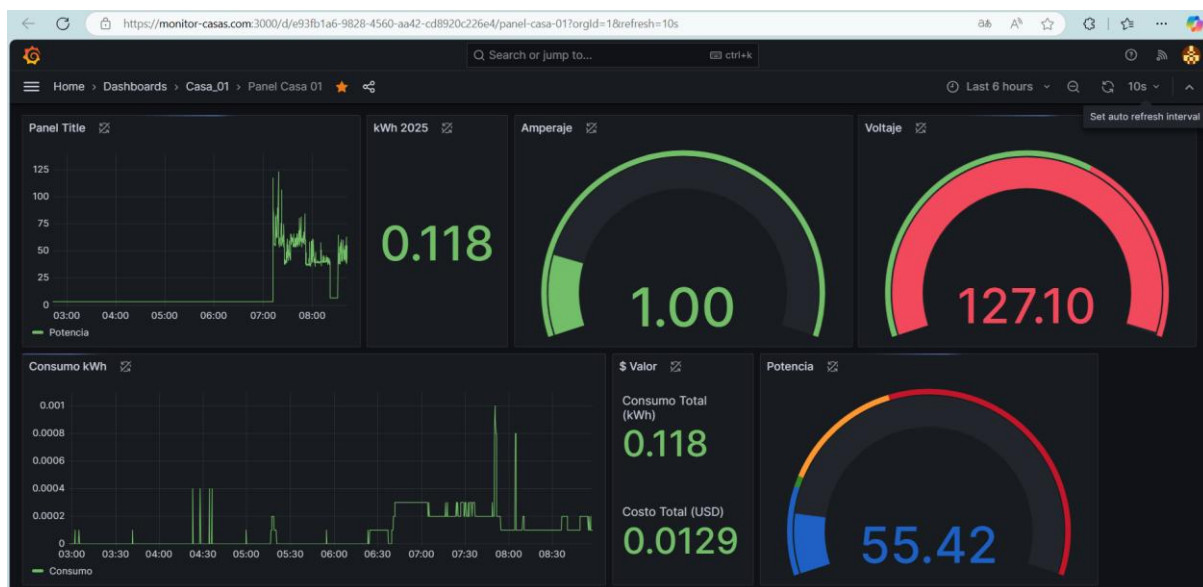
En esta etapa se analizarán los datos visualizados en los Dashboards de la plataforma web IoT implementada, evaluando los patrones de consumo eléctrico en tiempo real. Esto permitirá validar la efectividad del sistema IoT en la optimización del uso de la energía en los hogares.

Vista general del dashboard de la casa número 1

En esta imagen se observa el panel principal del sistema de monitoreo de consumo energético en la casa número 1. Se presentan mediciones en tiempo real de variables clave como voltaje (127.10V), corriente (1.00A), potencia (55.42W) y consumo acumulado en kWh (0.118 kWh). Además, se incluye el costo estimado del consumo eléctrico. La visualización gráfica permite interpretar patrones de uso de energía, lo que facilita la toma de decisiones para mejorar la eficiencia energética del hogar.

Figura 157

Panel Principal Casa 1.



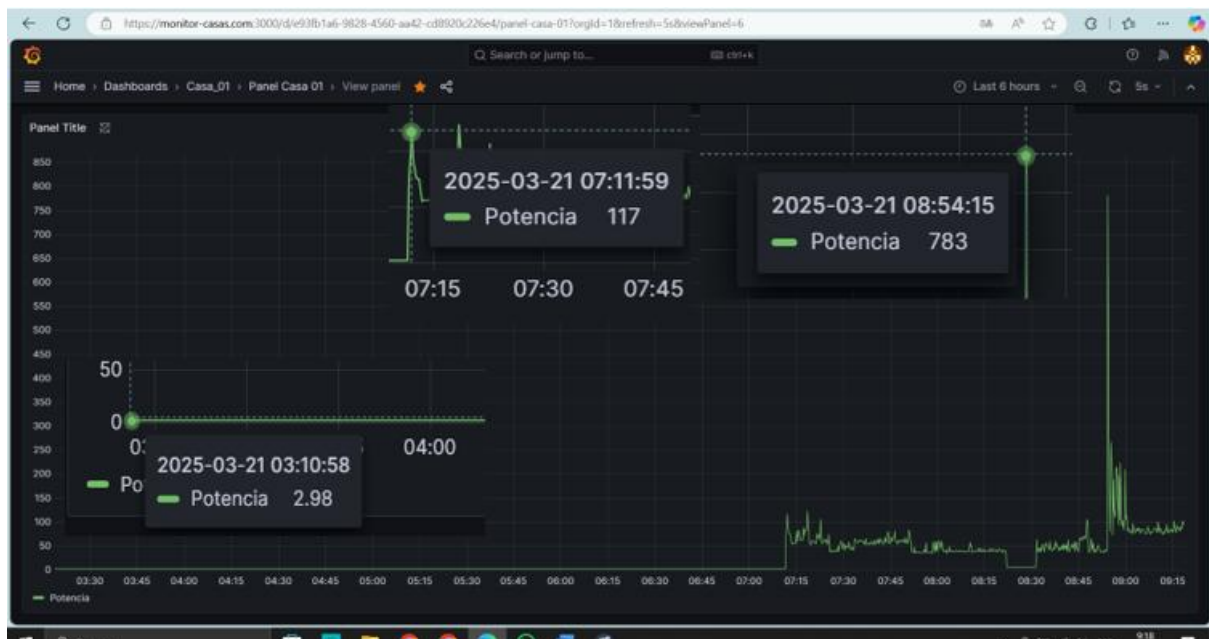
Nota: La imagen muestra el monitoreo en tiempo real de la casa número 1, el panel principal engloba varios gauges y graficas que muestran las mediciones tomadas por el prototipo IoT y visualizadas en la página web IoT. Fuente: Elaboración Propia.

- **Gráfico de Potencia en el Tiempo**

Este gráfico muestra la variación de la potencia consumida en función del tiempo, permitiendo identificar momentos de mayor y menor demanda eléctrica. Se observa que entre las 2:45 a. m. y las 7:10 a. m. la potencia se mantuvo relativamente baja, oscilando entre 1 y 3 W. A partir de las 7:11 a. m., la potencia comenzó a aumentar gradualmente, y a las 7:12 a. m. tuvo un incremento repentino, lo que podría indicar el encendido de electrodomésticos de alto consumo. Posteriormente, los valores fluctúan entre 123 W y 40 W. Sin embargo, a las 8:54 a. m. se registra un pico de consumo de 783 W, lo que indica que en ese momento un equipo con alta demanda energética entró en funcionamiento. Tras este pico, la potencia se estabiliza y vuelve a los valores mencionados anteriormente.

Figura 158

Grafica de Potencia casa número 1.



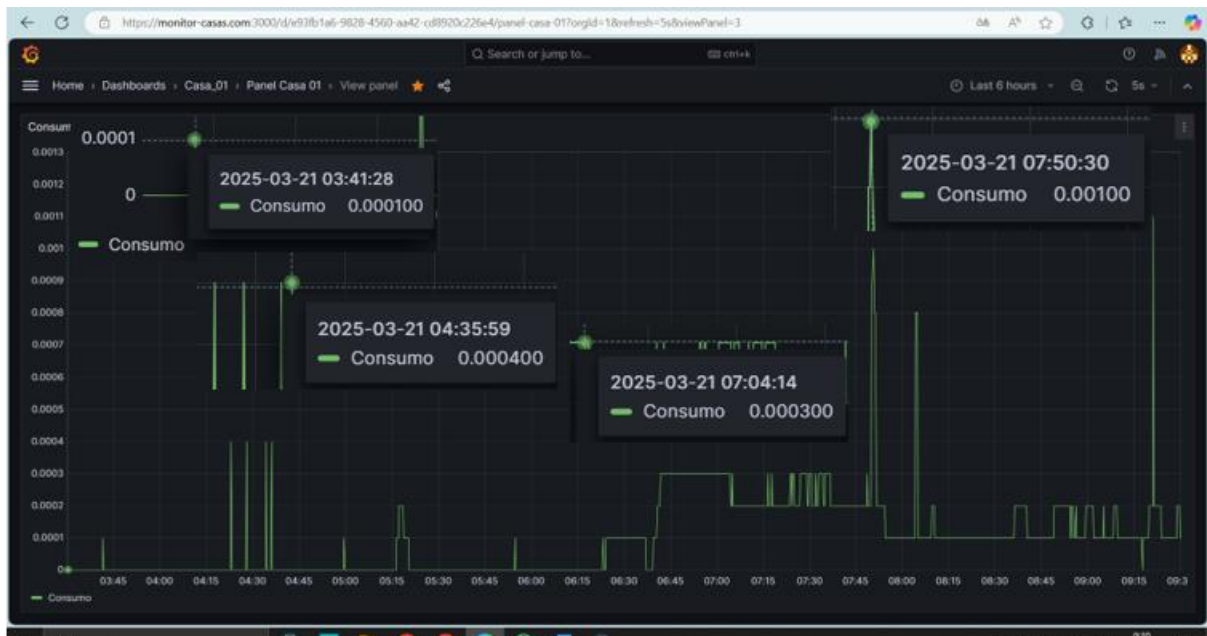
Nota: La imagen muestra la variación de la potencia consumida a lo largo del tiempo, permitiendo identificar picos de demanda y momentos de menor consumo eléctrico. Fuente: Elaboración Propia.

- **Registro del Consumo Energético (kWh)**

El gráfico muestra la variación del consumo eléctrico en una vivienda durante un período de tiempo específico. Se observan picos en momentos clave, como a las 03:41:28 con 0.000100, a las 04:35:59 con 0.000400 y a las 07:04:14 con 0.000300. El punto con mayor consumo ocurre a las 07:50:30 con 0.00100, lo que sugiere un aumento en la actividad del hogar, posiblemente por el uso de electrodomésticos. Entre estos picos, el consumo se mantiene en niveles bajos o cercanos a cero, indicando periodos de inactividad.

Figura 159

Grafica de Consumo Eléctrico casa número 1.



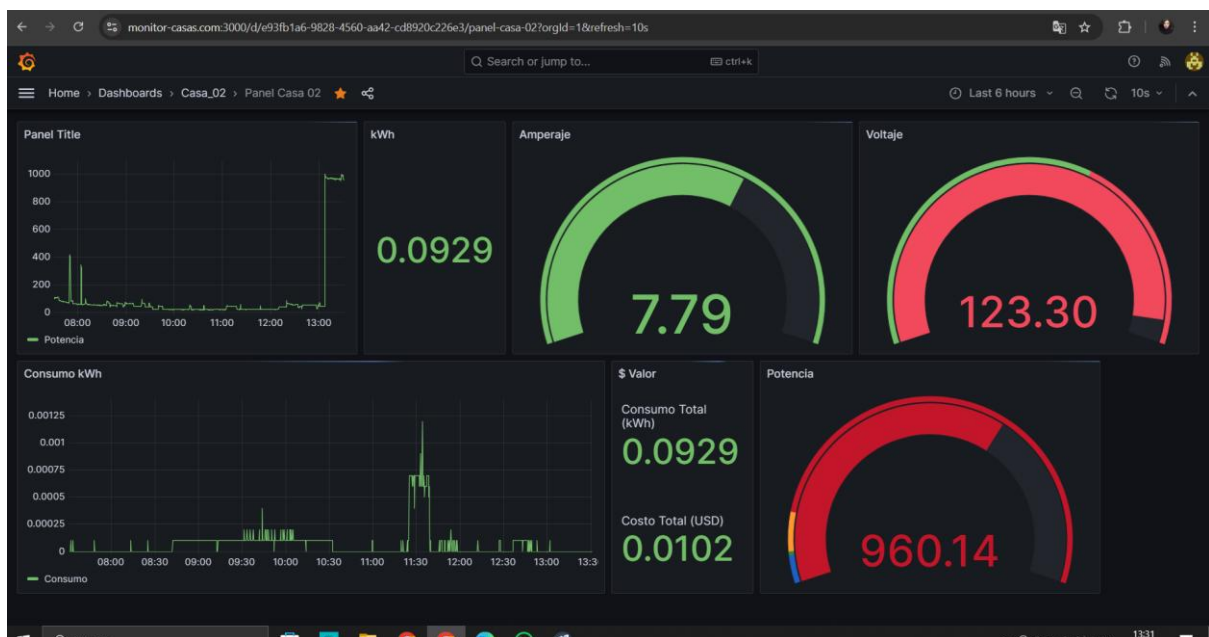
Nota: La imagen presenta el consumo eléctrico en kilovatios-hora (kWh) en el transcurso del día, facilitando el análisis del gasto energético en función del tiempo Fuente: Elaboración Propia.

- **Vista general del dashboard de la casa número 2**

En esta imagen se observa el panel principal del sistema de monitoreo de consumo energético en la casa número 1. Se presentan mediciones en tiempo real de variables clave como voltaje (123.30V), corriente (7.79 A), potencia (960.14W) y consumo acumulado en kWh (0.0929 kWh). Además, se incluye el costo estimado del consumo eléctrico. La visualización gráfica permite interpretar patrones de uso de energía, lo que facilita la toma de decisiones para mejorar la eficiencia energética del hogar.

Figura 160

Panel Principal Casa 2.



Nota: La imagen muestra el monitoreo en tiempo real de la casa número 2, en la Dashboard se puede visualizar los parámetros medidos de voltaje, amperaje, potencia y consumo eléctrico, contrario de la casa número uno en esta se puede distinguir que los parámetros medidos son mucho mayores en amperaje y potencia. Fuente: Elaboración Propia.

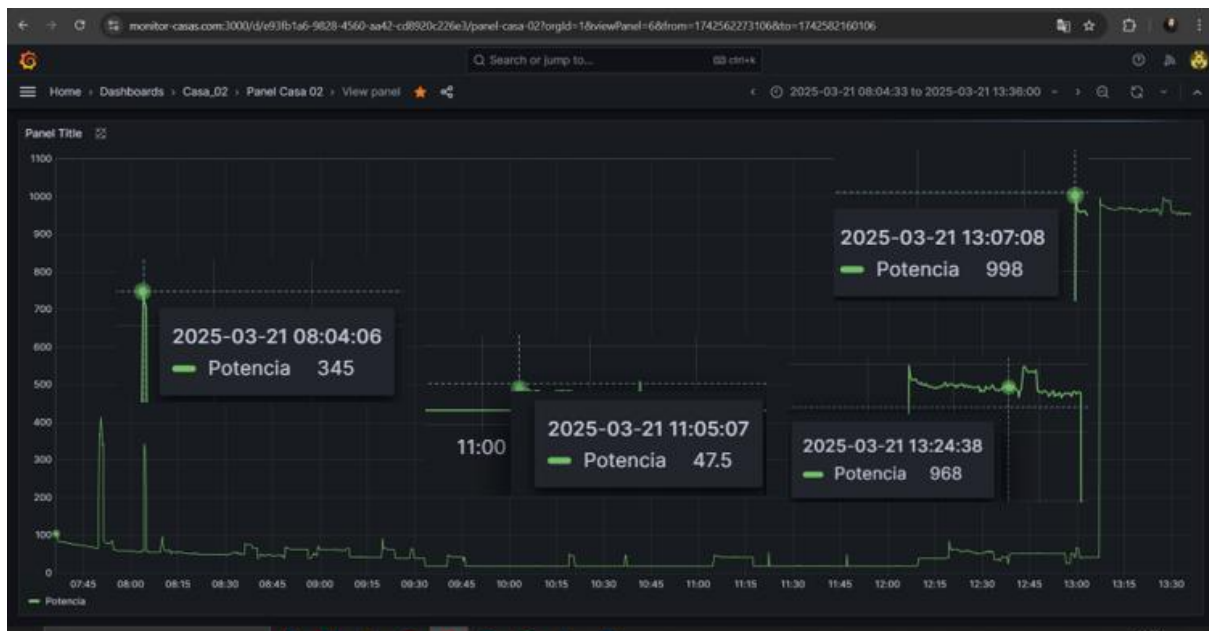
- **Gráfico de Potencia en el Tiempo**

El grafico muestra el registro de la potencia consumida a lo largo del tiempo mediante una gráfica de líneas en color verde. En el análisis de los datos, se observan fluctuaciones en la potencia con picos significativos en ciertos momentos del día. A las 08:04 am se registró un valor de 345 W, seguido de una disminución a 47.5 a las 11:05 am, la cual se mantuvo en ese rango hasta la 13.6 pm sin picos altos de potencia. Posteriormente, se observa un aumento considerable en la potencia a partir de las 13:07,

alcanzando 998, y un leve descenso a 968 a las 13:24. Estos datos reflejan patrones de consumo variables que podrían estar relacionados con el uso de dispositivos de alto consumo energético en determinados momentos del día.

Figura 161

Grafica de Potencia casa número 2.



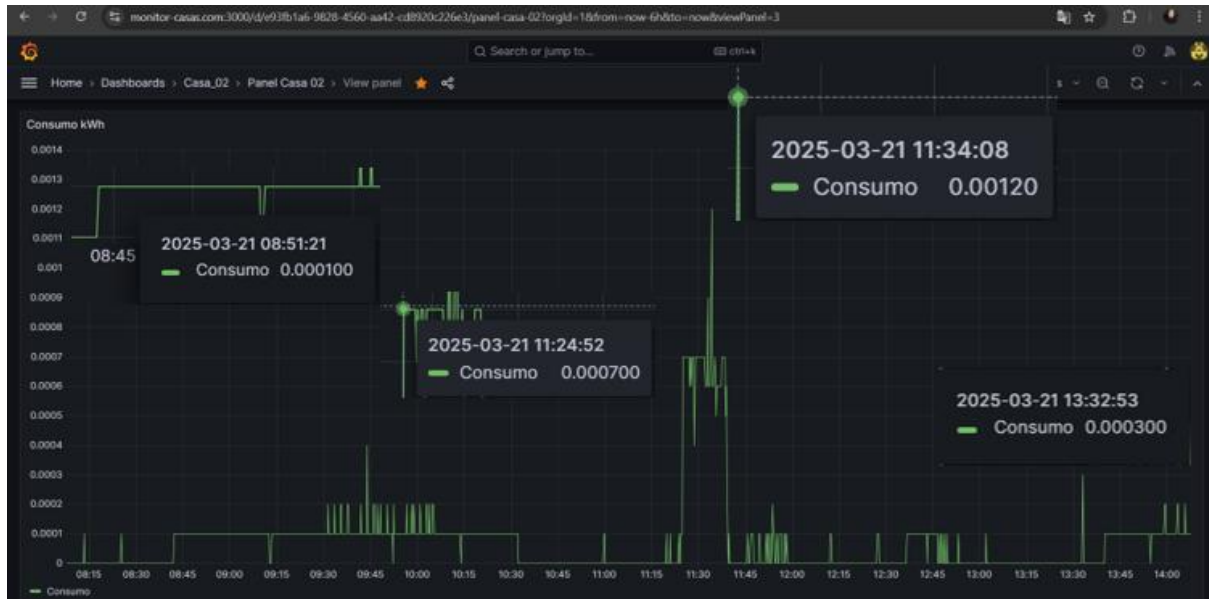
Nota: La imagen muestra la variación de la potencia consumida a lo largo del tiempo, se tomó 4 mediciones en distintos puntos de la gráfica detallando la fecha, hora y la potencia que se reflejó en ese lapso de tiempo. Fuente: Elaboración Propia.

- **Registro del Consumo Energético (kWh)**

La imagen representa la evolución del consumo de energía en kilovatios-hora (kWh) a lo largo del tiempo, reflejando como varía el consumo eléctrico en la vivienda. Se observan picos significativos en ciertos momentos del día, lo que indica periodos de mayor actividad eléctrica. A las 08:51am, el consumo registrado fue de 0.000100 kWh, el cual fue aumentando a 0.000700 kWh a las 11:24 am, y alcanzando un pico máximo de 0.00120 kWh a las 11:34, lo que podría estar relacionado con el uso simultáneo de varios electrodomésticos. Posteriormente, el consumo disminuye a 0.000300 kWh a las 13:32 pm, lo que sugiere una reducción en la actividad eléctrica. Este tipo de análisis permite comprender los hábitos de consumo energético y evaluar oportunidades para optimizar el uso de la electricidad, reduciendo costos sin afectar el confort del hogar.

Figura 162

Grafica de Consumo Eléctrico casa número 2.



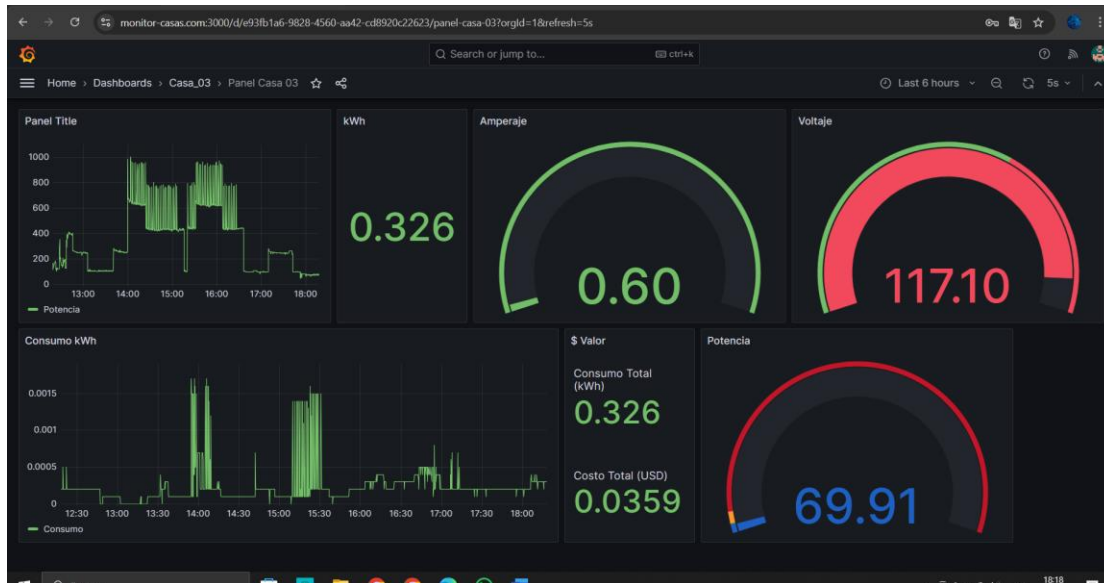
Nota: El gráfico refleja el consumo de energía en kWh a lo largo del día, destacando picos a las 08:51 con 0.000100, a las 11:24 con 0.000700 y alcanzando su punto máximo a las 11:34 con 0.00120, seguido de una reducción a 0.000300 a las 13:32. Estas fluctuaciones pueden estar relacionadas con el uso intermitente de electrodomésticos y permiten evaluar la eficiencia energética del hogar. Fuente: Elaboración Propia.

- **Vista general del dashboard de la casa número 3**

En esta imagen se observa el panel principal del sistema de monitoreo de consumo energético en la casa número 3. Se presentan mediciones en tiempo real de variables clave como voltaje (117.10V), corriente (0.60 A), potencia (69.91 W) y consumo acumulado en kWh (0.326 kWh). Además, se incluye el costo estimado del consumo eléctrico. La visualización gráfica permite interpretar patrones de uso de energía, lo que facilita la toma de decisiones para mejorar la eficiencia energética del hogar.

Figura 163

Panel Principal Casa 3.



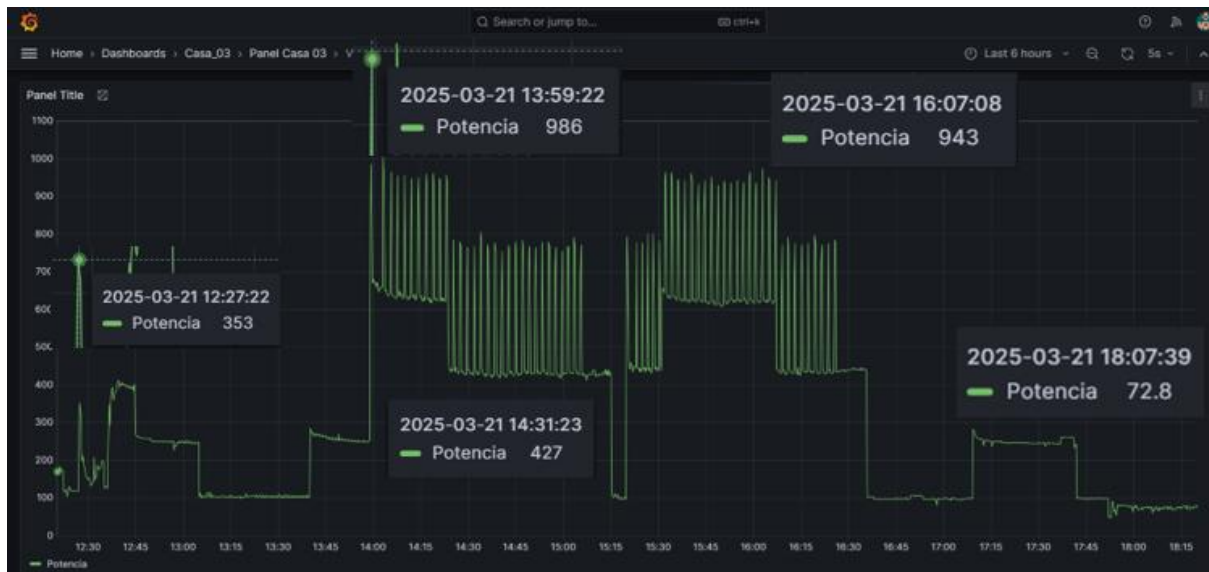
Nota: La imagen muestra el monitoreo en tiempo real de la casa número 3, el panel principal engloba varios gauges y graficas que muestran las mediciones tomadas por el prototipo IoT y visualizadas en la página web IoT. Fuente: Elaboración Propia

- **Gráfico de Potencia en el Tiempo**

Se observan variaciones significativas en la demanda eléctrica, con picos y descensos en distintos momentos. A las 12:27, la potencia registrada fue de 353, seguida de un aumento a 986 a las 13:59, lo que representa el punto de mayor consumo. Posteriormente, se observa una leve reducción a 427 a las 14:31, antes de alcanzar otro pico de 943 a las 16:07, sugiriendo un alto uso de dispositivos eléctricos en ese periodo. Finalmente, la potencia desciende a 72.8 a las 18:07, indicando una menor actividad eléctrica en la vivienda. Este análisis refleja patrones de consumo que pueden estar asociados con rutinas diarias y el uso de electrodomésticos de alto consumo, lo que permite evaluar la eficiencia energética y optimizar la gestión de la electricidad en el hogar.

Figura 164

Grafica de Potencia casa número 3.



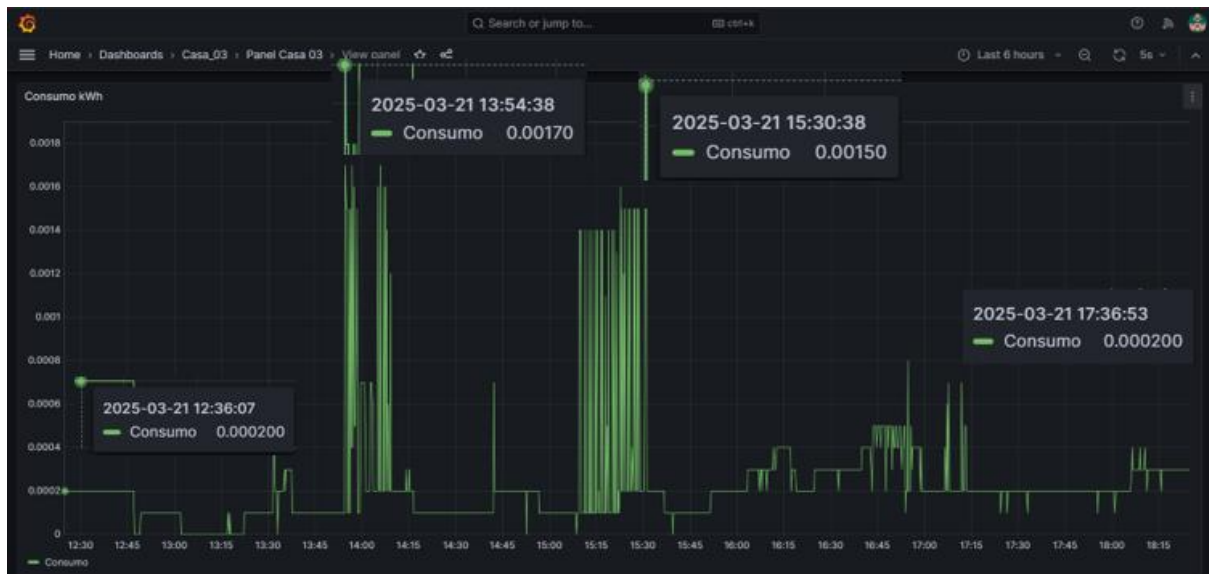
Nota: El gráfico muestra la variación de la potencia eléctrica en una vivienda. Se observan picos de 986 a las 13:59 y 943 a las 16:07, reflejando momentos de alto consumo, seguidos de descensos hasta 72.8 a las 18:07, indicando menor actividad eléctrica.

- **Registro del Consumo Energético (kW h)**

Se identifican picos importantes en distintos momentos del día, con un consumo de 0.000200 kWh a las 12:36, seguido de un aumento significativo a 0.00170 kWh a las 13:54:38, que representa el punto de mayor consumo registrado. Posteriormente, se observa una ligera reducción a 0.00150 kWh a las 15:30, y finalmente, el consumo desciende nuevamente a 0.000200 kWh a las 17:36, indicando una menor actividad eléctrica en el hogar. Estas fluctuaciones sugieren patrones de uso asociados a la activación y desactivación de electrodomésticos, permitiendo analizar tendencias y optimizar la eficiencia energética en la vivienda.

Figura 165

Grafica de Consumo Eléctrico casa número 3.



Nota: La imagen refleja el consumo energético en kWh a lo largo del día. Se identifican picos de 0.00170 kWh a las 13:54 y 0.00150 kWh a las 15:30, con períodos de menor consumo en otros momentos, evidenciando variaciones en el uso de electricidad.

- **Vista general del dashboard de la casa número 4**

En esta imagen se observa el panel principal del sistema de monitoreo de consumo energético en la casa número 1. Se presentan mediciones en tiempo real de variables clave como voltaje (127.10V), corriente (1.00A), potencia (55.42W) y consumo acumulado en kWh (0.118 kWh). Además, se incluye el costo estimado del consumo eléctrico. La visualización gráfica permite interpretar patrones de uso de energía, lo que facilita la toma de decisiones para mejorar la eficiencia energética del hogar.

Figura 166

Panel Principal Casa 4.



Nota: La imagen muestra el monitoreo en tiempo real de la casa número 4, en la Dashboard se puede visualizar los parámetros medidos de voltaje, amperaje, potencia y consumo eléctrico.

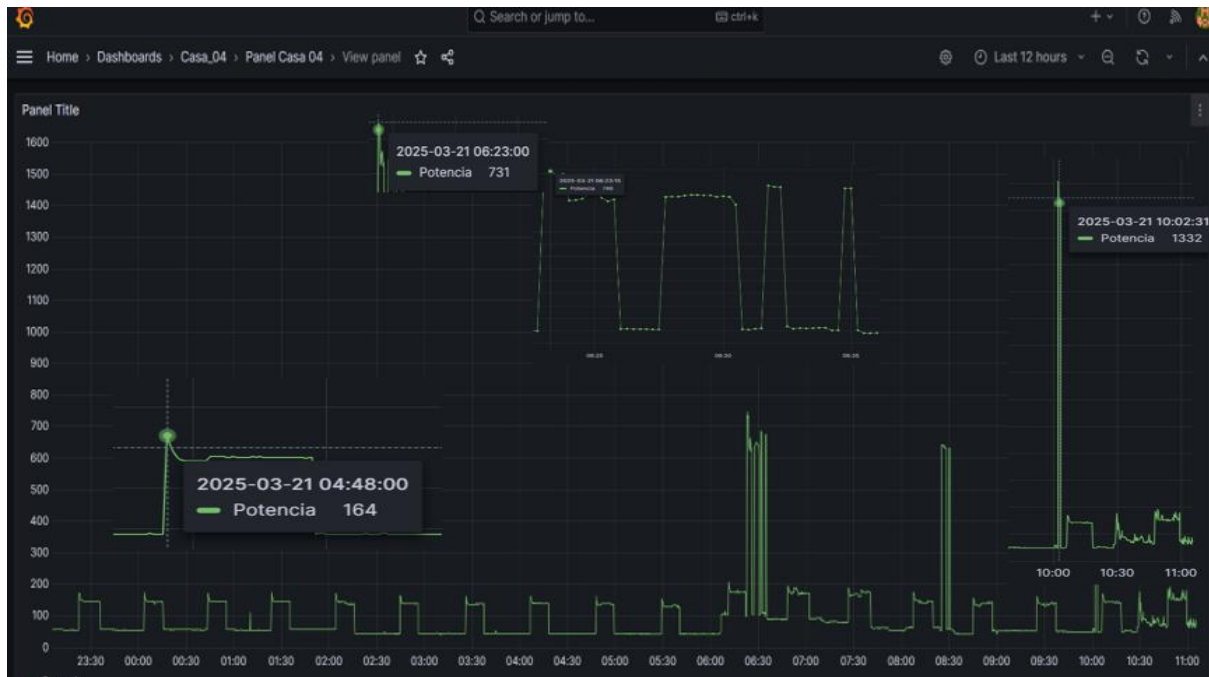
Fuente: Elaboración Propia.

- **Gráfico de Potencia en el Tiempo**

Este gráfico muestra la variación de la potencia consumida en función del tiempo, permitiendo identificar momentos de mayor y menor demanda eléctrica. Se observa que entre las 4:00 a. m. y las 6:05 a. m. la potencia se mantuvo relativamente baja, oscilando entre 55 W y 164 W sabiendo que ese consumo es de los equipos conectados de manera continua en el hogar. A partir de las 6:10 a. m., la potencia comenzó a aumentar gradualmente, y a las 6:23 a. m. tuvo un incremento repentino, lo que podría indicar el encendido de electrodomésticos de alto consumo como el microondas, las luces hasta aproximadamente las 7:00 a. m. Posteriormente, los valores fluctúan entre 80 W y 170 W. Sin embargo, a las 10:02 a. m. se registra un pico de consumo de 1332 W, identificado con el uso de una secadora de cabello. Tras este pico, la potencia se estabiliza y vuelve a los valores mencionados anteriormente.

Figura 167

Grafica de Potencia casa número 4.



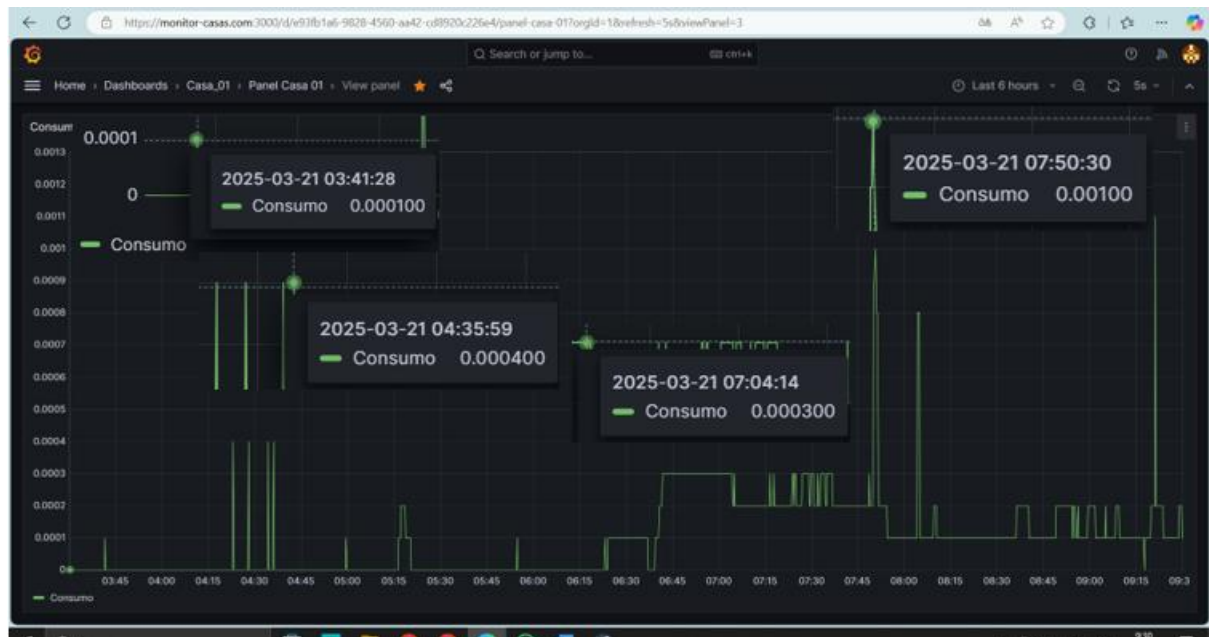
Nota: El gráfico muestra la variación de la potencia eléctrica en una vivienda durante un período de 12 horas. Se observan picos significativos de 164 a las 04:48, 731 a las 06:23 y un máximo de 1332 a las 10:02, reflejando momentos de alto consumo energético, posiblemente asociados al uso de electrodomésticos de gran demanda.

- **Registro del Consumo Energético (kWh)**

El gráfico muestra la variación del consumo eléctrico en una vivienda durante un período de tiempo específico. Se observan picos en momentos clave, como a las 03:41:28 con 0.000100, a las 04:35:59 con 0.000400 y a las 07:04:14 con 0.000300. El punto con mayor consumo ocurre a las 07:50:30 con 0.00100, lo que sugiere un aumento en la actividad del hogar, posiblemente por el uso de electrodomésticos. Entre estos picos, el consumo se mantiene en niveles bajos o cercanos a cero, indicando periodos de inactividad.

Figura 168

Grafica de Consumo Eléctrico casa número 4.



Nota: El gráfico representa el consumo eléctrico acumulado en kWh durante un período de tiempo en una vivienda. Se observan variaciones en el consumo con picos en 03:41 (0.000100 kWh), 04:35 (0.000400 kWh) y 07:04 (0.000300 kWh), alcanzando un máximo de 0.00100 kWh a las 07:50, lo que indica momentos de mayor uso de dispositivos eléctricos.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se logró desarrollar e implementar una plataforma web basada en IoT para la gestión del consumo eléctrico en hogares urbanos, permitiendo a los usuarios visualizar en tiempo real sus datos energéticos y tomar decisiones informadas para mejorar su eficiencia.
- La combinación de sensores PZEM-004T, microcontroladores ESP32, protocolos MQTT, almacenamiento en MySQL y visualización en Grafana demostró ser una solución viable para el monitoreo energético, asegurando una infraestructura escalable y adaptable a futuras mejoras.
- La comparación entre las mediciones del prototipo IoT y un medidor convencional confirmó la precisión del sistema.
- A pesar de los desafíos en costos de implementación y la necesidad de educación en el uso de la plataforma, el sistema demostró ser funcional y con alto potencial de expansión a más hogares, contribuyendo a la sostenibilidad y optimización del consumo energético.
- La accesibilidad a los datos en tiempo real y la posibilidad de visualizar tendencias de consumo generaron un mayor interés por parte de los usuarios en mejorar sus hábitos energéticos, evidenciando que la tecnología no solo facilita el monitoreo, sino que también motiva al cambio en el comportamiento del consumidor.

5.2 Recomendaciones

- Se sugiere expandir la plataforma a más hogares y sectores comerciales, lo que permitirá validar su rendimiento en distintos escenarios y optimizar su operatividad a mayor escala.
- Para facilitar su adopción, es recomendable optimizar la experiencia del usuario, asegurando que la visualización de datos sea clara, intuitiva y accesible desde diferentes dispositivos.
- Se recomienda integrar herramientas de análisis predictivo mediante inteligencia artificial o machine learning para anticipar patrones de consumo y sugerir medidas de ahorro personalizadas para cada usuario.
- Es fundamental generar campañas de capacitación sobre el uso de la plataforma, así como fomentar hábitos de consumo eficiente mediante alertas y reportes personalizados dentro del sistema.
- Para mejorar la estabilidad y escalabilidad del sistema, se recomienda evaluar servidores con mayor capacidad, optimizar el procesamiento de datos en la nube y explorar la posibilidad de implementar redes LoRaWAN para extender el alcance del monitoreo en zonas con menor infraestructura de internet.

Referencias

- Cantos, A. (2021, septiembre 4). *Plataformas IoT: Diferentes tipos de plataformas y cómo elegir la que necesitas* - Barbara. <https://www.barbara.tech/es/blog/tipos-de-plataforma-iot>
- Carrillo Baquerizo, K. G., & León Arias, C. A. (2021). *Diseño e implementación de un sistema SCADA basado en el software libre Node-red para el monitoreo y operación de la planta didáctica MPS PA. Compact Workstation de Festo* [bachelorThesis]. <http://dspace.ups.edu.ec/handle/123456789/21905>
- Dignal Academy. (2024). *Curso Online Plataformas Web IoT | Dignal*. <https://dignal.com/plataformas-web-iot/>
- Eclipse Mosquitto. (2020). *Un corredor de código abierto MQTT*. Eclipse Foundation. <https://mosquitto.org/>
- Electronica Estudio. (2019). *PICmicro® MCU Estudio—¿qué es un microcontrolador?* Electronica Estudio. <https://www.estudioelectronica.com/que-es-un-microcontrolador/>
- Guasch Llobera, J., & Calleja Collado, M. (2019). *Monitorización de sensores con arduino utilizando el protocolo MQTT* [Bachelor thesis, Universitat Politècnica de Catalunya]. <https://upcommons.upc.edu/handle/2117/134193>
- Gupta, B., Mittal, P., & Mufti, T. (2021, marzo 11). *A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services*. Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India. <https://eudl.eu/doi/10.4108/eai.27-2-2020.2303255>
- Harahap, P., Pasaribu, F. I., & Adam, M. (2020). *Prototype Measuring Device for Electric Load in Households Using the Pzem-004T Sensor*. <https://www.semanticscholar.org/paper/Prototype-Measuring-Device-for-Electric-Load-in-the-Harahap-Pasaribu/6631bc1317f4b66220efc085475ef53c25f45bac>
- Hernández Acevedo, F. M. (2016). *Y se hizo la luz Voltaje, corriente y potencia eléctrica* (Primera). <https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/12235/5/imagenes/voltaje-ciencias.pdf>
- IEEE. (2015). *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications* (Vol. 17). <https://ieeexplore.ieee.org/document/7123563>
- Ikiss, J. (2020). *Sistema de adquisición de datos con ESP32* [Bachelor thesis, Universitat Politècnica de Catalunya]. <https://upcommons.upc.edu/handle/2117/344400>
- Khanh, D. B. (2021). *Data Visualization with Integration of Grafana and Quanta Platform* [fi=AMK-opinnäytetyö|sv=YH-examensarbete|en=Bachelor's thesis]. <http://www.theseus.fi/handle/10024/497937>
- Lozano Guamán, S. L. (2020). *Rediseño del sistema eléctrico en baja tensión para la Universidad Politécnica Salesiana Campus El Girón bloque b basado en una campaña de medidas para la ejecución de un plan de mejoras* [bachelorThesis]. <http://dspace.ups.edu.ec/handle/123456789/18800>

- Oracle. (2025). *MySQL :: MySQL Documentation*. <https://dev.mysql.com/doc/>
- Ovando Ortega, D. J. (2019). *Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web* [UNIVERSIDAD POLITÉCNICA DE SINALOA]. <http://repositorio.upsin.edu.mx/Fragmentos/tesinas/142016030030OvandoOrtegaDenzelJavier10843.pdf>
- Pérez Ibarra, S. G., Quispe, J. R., Mullicundo, F. F., & Lamas, D. A. (2021). *Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd*. XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja). <http://sedici.unlp.edu.ar/handle/10915/120476>
- React. (2021). *Inicio rápido – Aprender React*. <https://es.react.dev/learn>
- Recalde, H., & Zambrano, W. (2022). Análisis, diseño, implementación y gestión de un sistema de bases de datos para automatizar los procesos de facturación de la empresa “OKI DOKI”. *NEXOS CIENTÍFICOS - ISSN 2773-7489*, 6(1), Article 1.
- Rivera Berrío, J. G., & Lopera Sánchez, R. A. (2024). *Internet de las cosas*. https://proyectodescartes.org/iCartesiLibri/materiales_didacticos/IoT/index.html?page=2
- Rose, K., Eldridge, S., & Chapin, L. (2015). *La Internet de las Cosas—Una breve reseña*. <https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf>
- Rosencrance, L. (2024, julio 12). Las 5 mejores plataformas IoT gratuitas para 2024. *Techopedia Español*. <https://www.techopedia.com/es/mejores-plataformas-iot-gratuitas>
- Ternium. (2020). *Manual de Contenido del Participante Electricidad Básica.pdf* (Vol. 2). https://www.trabajosocial.unlp.edu.ar/uploads/docs/electricidad_basica_ii.pdf

ANEXOS

Anexo 1. Código Arduino IDE, ESP32, Sensor Pzem-004T, MQTT

```
//*****  
//***   LIBRERÍAS   ****  
//*****  
#include <Arduino.h>  
#include <WiFi.h>  
#include <PubSubClient.h>  
#include <PZEM004Tv30.h>  
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
  
//*****  
//***   CONFIGURACIÓN OLED   ****  
//*****  
#define SCREEN_WIDTH 128 // Ancho del display OLED en píxeles  
#define SCREEN_HEIGHT 64 // Alto del display OLED en píxeles  
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);  
  
//*****  
//***   CONFIGURACIÓN PZEM   ****  
//*****  
#define RXD2 16  
#define TXD2 17  
#define PZEM_SERIAL Serial2  
PZEM004Tv30 sen_pzem(PZEM_SERIAL, RXD2, TXD2);  
  
//*****  
//***   CONFIGURACIÓN WIFI   ****  
//*****  
const char* ssid      = "TP-Link_B5A9"; // Nombre de la red WiFi  
const char* password = "42995142"; // Contraseña de la red WiFi  
  
//*****  
//***   CONFIGURACIÓN MQTT   ****  
//*****  
const char *mqtt_server = "monitor-casas.com";  
const int  mqtt_port = 1883;  
const char *mqtt_user = "my_user_mqtt_casas";  
const char *mqtt_pass = "rasp_casas_mqtt_2024";  
const String Topic_Out = "Monitoreo/Casa_1/Out"; // Topic de salida de Casa  
Monitoreada  
const String Topic_In = "Monitoreo/Casa_1/In"; // Topic de entrada de Casa  
Monitoreada
```

```

//*****
//*** VARIABLES GLOBALES ****
//*****
float voltage, current, power, energy, frequency, pf = 0;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;

//*****
//*** DECLARACIÓN FUNCIONES ***
//*****
void setup_wifi();
void callback(char* topic, byte* payload, unsigned int length);
void reconnect();

//*****
//*** CONFIGURACIÓN INICIAL ***
//*****
void setup()
{
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(callback);

    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println(F("SSD1306 allocation failed"));
        for(;;);
    }
    delay(500);
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0, 16); display.println(" MEDIDOR ");
    display.setCursor(0, 32); display.println(" ELECTRICO ");
    display.display();
    delay(500);
}

//*****
//*** BUCLE PRINCIPAL ****
//*****
void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
    long now = millis();

```

```

if (now - lastMsg > 15000){
    lastMsg = now;
    // Leer datos del sensor
    voltage = sen_pzem.voltage();
    current = sen_pzem.current();
    power = sen_pzem.power();
    energy = sen_pzem.energy();
    frequency = sen_pzem.frequency();
    pf = sen_pzem.pf();

    // Validar datos
    if(isnan(voltage)) { voltage = 0; }
    if(isnan(current)) { current = 0; }
    if(isnan(power)) { power = 0; }
    if(isnan(energy)) { energy = 0; }
    if(isnan(frequency)) { frequency = 0; }
    if(isnan(pf)) { pf = 0; }

    // Mostrar datos en el monitor serial
    Serial.println("<<<<< LECTURA SENSOR PZEM >>>>>");
    Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");
    Serial.print("Current: "); Serial.print(current); Serial.println("A");
    Serial.print("Power: "); Serial.print(power); Serial.println("W");
    Serial.print("Energy: "); Serial.print(energy,3); Serial.println("kWh");
    Serial.print("Frequency: "); Serial.print(frequency, 1);
Serial.println("Hz");
    Serial.print("PF: "); Serial.println(pf);
    Serial.println();

    // Mostrar datos en OLED
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 8); display.println("Voltaje: " +
String(voltage));
    display.setCursor(0, 16); display.println("Corriente: " +
String(current));
    display.setCursor(0, 24); display.println("Potencia: " +
String(power));
    display.setCursor(0, 32); display.println("Energía: " +
String(energy));
    display.setCursor(0, 40); display.println("Frecuencia: " +
String(frequency));
    display.setCursor(0, 48); display.println("PF: " + String(pf));
    display.display();

    // Enviar datos vía MQTT

```

```

        String StrMsg =
String(voltage,2)+"@"+String(current,4)+"@"+String(power,2);
        client.publish(Topic_Out.c_str(), StrMsg.c_str());
    }
}

//*****
//***      CONEXIÓN WIFI      ***
//*****
void setup_wifi()
{
    delay(10);
    Serial.println();
    Serial.print("Conectando a ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nConectado a red WiFi!");
    Serial.println("Dirección IP: ");
    Serial.println(WiFi.localIP());
}

//*****
//***      CALLBACK MQTT      ***
//*****
void callback(char* topic, byte* payload, unsigned int length)
{
    String incoming = "";
    Serial.print("Mensaje recibido desde -> "); Serial.print(topic);
    Serial.println("");
    for (int i = 0; i < length; i++) {
        incoming += (char)payload[i];
    }
    incoming.trim();
    Serial.println("Mensaje -> " + incoming);
}

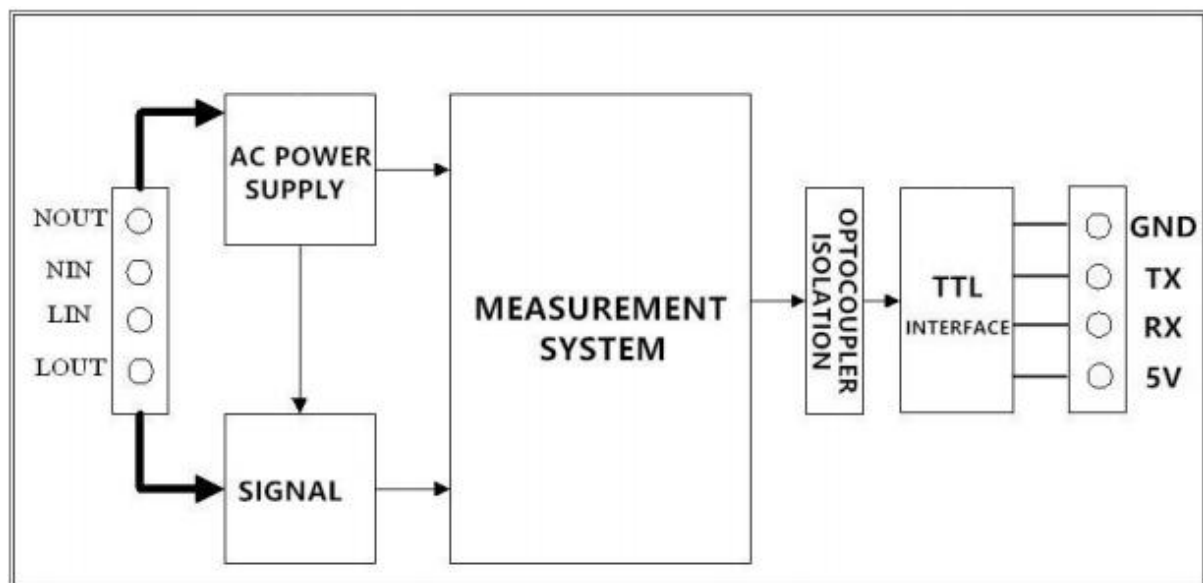
//*****
//***      RECONEXIÓN MQTT      ***
//*****
void reconnect()
{
    while (!client.connected()) {
        Serial.print("Intentando conexión Mqtt...");
        String clientId = "esp32_";

```

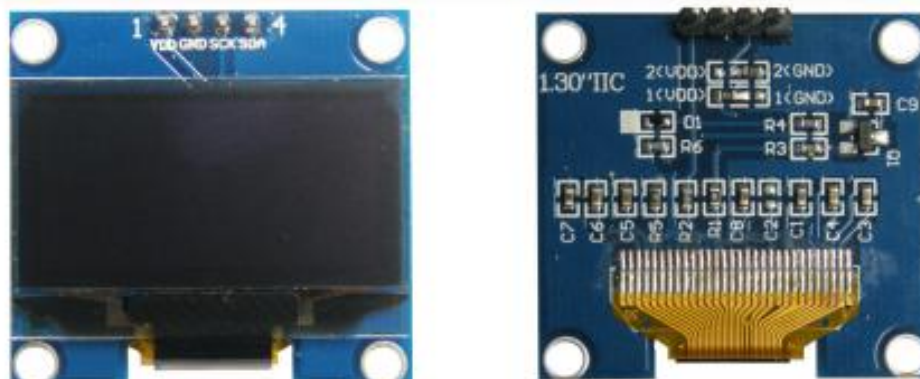
```
clientId += String(random(0xffff), HEX);  
if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {  
    Serial.println("Conectado!");  
    Serial.print("Nos suscribimos a: "); Serial.println(Topic_Out);  
    client.subscribe(Topic_In.c_str());  
} else {  
    Serial.print("falló :( con error -> ");  
    Serial.print(client.state());  
    Serial.println(" Intentamos de nuevo en 5 segundos");  
    delay(5000);  
}  
}  
}
```

Anexo 2. Datasheet modulo sensor Pzem-004T

Function	Measuring range		Starting measure current/power		Resolution	Measure-ment accuracy	Display format
	10A	100A	10A	100A			
Voltage	80~260V				0.1V	0.5%	
Current	0~10A	0~100A	0.01A	0.02A	0.001A	0.5%	
Active power	0~2.3kW	0~23kW	0.4W		0.1W	0.5%	<1000W, it display one decimal, such as: 999.9W; ≥1000W, it display only integer, such as: 1000W
Power factor	0.00~1.00				0.01	1%	
Frequency	45Hz~65Hz				0.1Hz	0.5%	
Active energy (Reset energy: use software to reset)	0~9999.99kWh				1Wh	0.5%	<110kWh, the display unit is Wh(1kWh=1000Wh), such as: 9999Wh; ≥10kWh, the display unit is kWh, such as: 9999.99kWh
Over power alarm	Active power threshold can be set, when the measured active power exceeds the threshold, it can alarm						
Communication interface	RS485 interface						
size	Length * width * height=73.7*30*14.3mm (Bare pager)						
Power Supply	The power supply of single-phase power-frequency network supplies power to the main circuit through resistance-capacitance step-down, TTL output communication interface and Main circuit optocoupler isolation, for passive output, communication needs to provide external 5V power supply						
working temperature	-20°C~+60°C						



Anexo 3. Datasheet pantalla oled i2c iic de 1.3 pulgadas



SPECIFICATIONS

- Use CHIP No.SH1106
- Use 3.3V-5V POWER SUPPLY
- Graphic LCD 1.3" in width with 128x64 Dot Resolution
- White Display is used for the model **OLED 1.3 I2C WHITE** and blue Display is used for the model **OLED 1.3 I2C BLUE**
- Use I2C Interface
- Directly connect signal to Microcontroller 3.3V and 5V without connecting through Voltage Regulator Circuit
- Total Current when running together is 8 mA
- PCB Size: 33.7 mm x 35.5 mm

Table shows name and function of Pin OLED

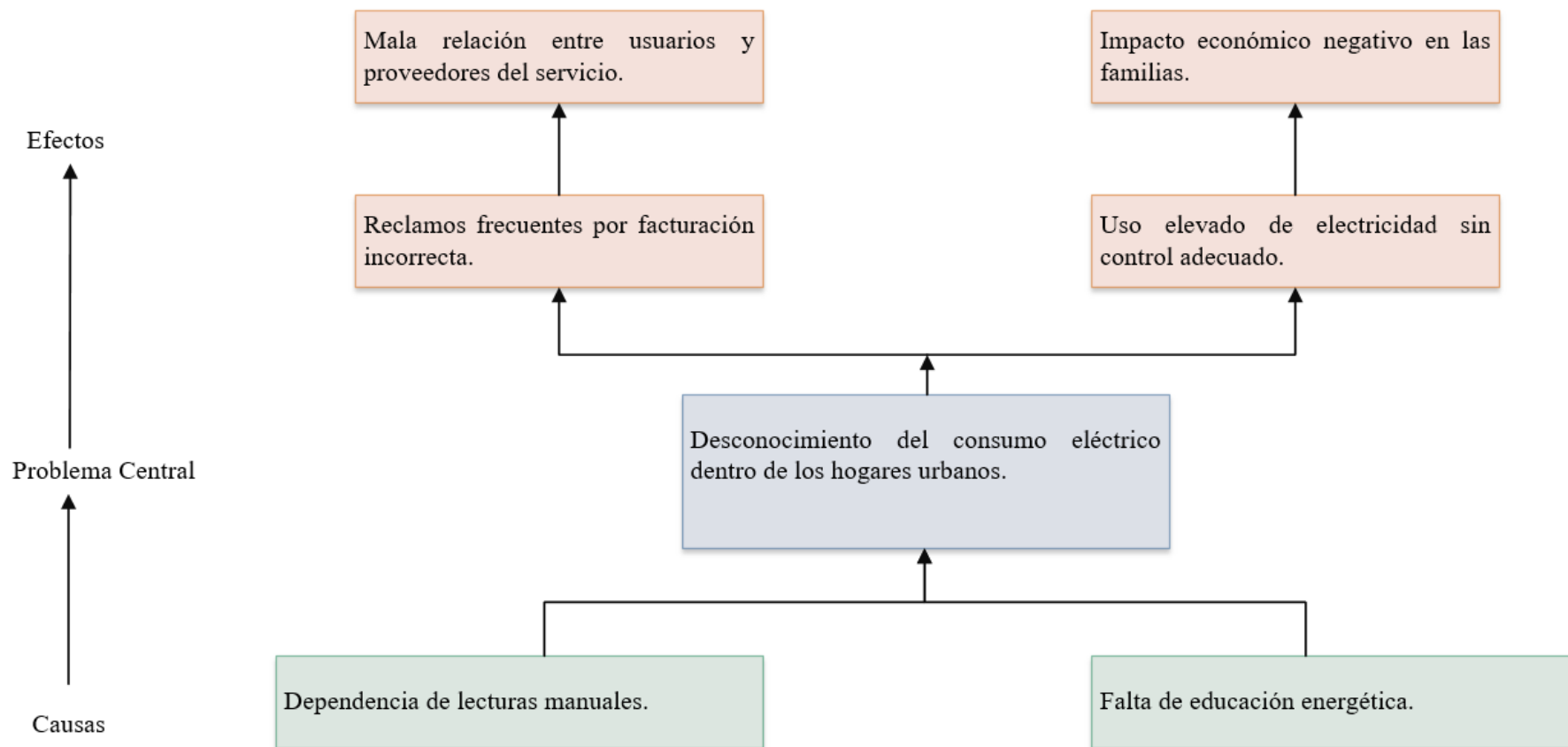
Pin No.	Pin Name	Description
1	VDD	Pin Power Supply for LCD, using 3.3V-5V
2	GND	Pin Ground
3	SCK	Pin SCL of I2C Interface
4	SDA	Pin SDA of I2C Interface

Anexo 4. Datasheet Esp32

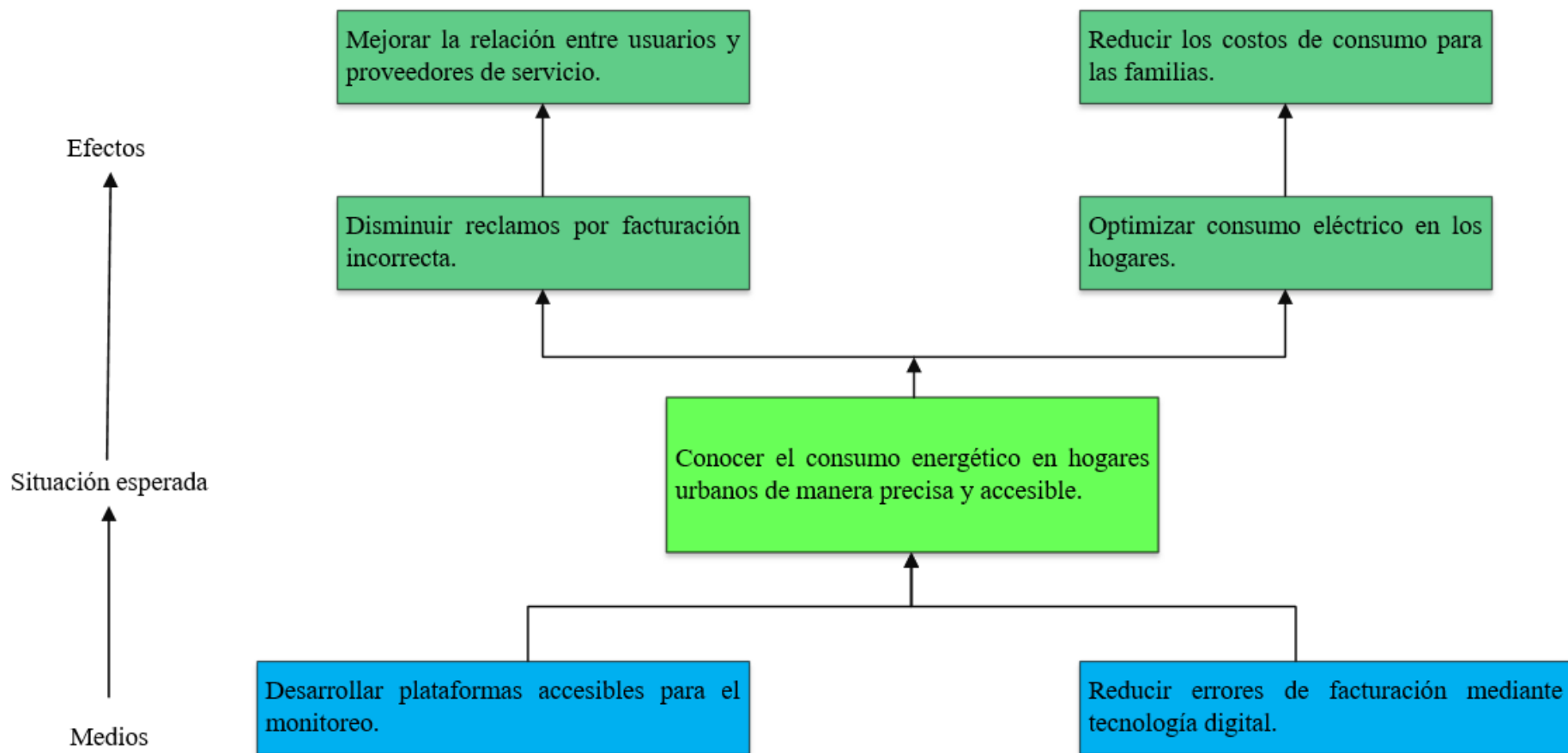
Categories	Items	Specifications
Certification	RF certification	FCC/CE/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green certification	RoHS/REACH
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps) A-MPDU and A-MSDU aggregation and 0.4 μ s guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC

Categories	Items	Specifications
Hardware	Module interface	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR
		GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor, temperature sensor
	On-board clock	40 MHz crystal
	Operating voltage/Power supply	2.7 ~ 3.6V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Operating temperature range	-40°C ~ +85°C
	Ambient temperature range	Normal temperature
	Package size	18±0.2 mm x 25.5±0.2 mm x 3.1±0.15 mm
Software	Wi-Fi mode	Station/SoftAP/SoftAP+Station/P2P
	Wi-Fi Security	WPA/WPA2/WPA2-Enterprise/WPS
	Encryption	AES/RSA/ECC/SHA
	Firmware upgrade	UART Download / OTA (download and write firmware via network or host)
	Software development	Supports Cloud Server Development / SDK for custom firmware development
	Network protocols	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
	User configuration	AT instruction set, cloud server, Android/iOS app

Anexo 5. Árbol de problemas



Anexo 6. Árbol de Objetivos



Anexo 7. Cronograma de Actividades

Nº	ACTIVIDADES	TIEMPOS									
		2024					2025				
		OCTUBRE					NOVIEMBRE				
		semana					semana				
		1	2	3	4	5	1	2	3	4	5
1	Elaboración de tema de investigación.										
2	Planteamiento del Problema.										
3	Objetivos.										
4	Justificación.										
5	Alcance del Proyecto.										
6	Obtención bibliográfica.										
7	Bibliografía.										
8	Marco Teórico.										
9	Formulación de diagrama de fases y etapas.										
10	Cotización.										
11	Presupuesto.										
12	Presentación del plan de proyecto completado.										
13	Estudio viabilidad para el desarrollo de un sistema de monitoreo energético.										
14	Diseño del esquema electrónico del prototipo de monitoreo.										
15	Diseño de PCB.										
16	Diseño 3D de la carcasa protectora del prototipo.										
17	Diseño conceptual de la plataforma web.										
18	Fabricación de la PCB y montaje de los elementos.										
19	Desarrollo del código para ESP32.										
20	Impresión 3D carcasa protectora del prototipo.										
21	Desarrollo implementación de la plataforma web IOT.										
22	Integración de prototipo de medición con la plataforma web.										
23	Pruebas de integración y depuración.										
24	Verificación de acceso correcto a la plataforma.										
25	Comparación de lecturas entre el medidor de la empresa eléctrica y el prototipo IOT.										
26	Inducción sobre el manejo de la plataforma.										
27	Retroalimentación por parte de los usuarios.										
28	Evaluar el impacto del monitoreo IOT sobre el consumo energético en los hogares.										
29	Redacción del informe final.										
30	Revisión escrito final y funcionamiento de la plataforma.										
31	Defensa del proyecto.										

Anexo 8. Presupuesto

PRE SUPUESTO DEL PROYECTO

Tipo de Proyecto: _____

DURACIÓN: 4 MESES

INGRESOS ESTIMADOS	1200.00
Costos directos	1110.17
Costos indirectos	25%
Reserva para riesgos	15%

DEFICIT	395.87
Presupuesto	1387.71
Riesgo	208.18
TOTAL	1595.87

DETALLE DE INGRESOS

Nº	DESCRIPCIÓN	APORTE ECONÓMICO
1	AMAGUADARIO	600.00
2	GAVILANES FERNANDO	600.00
3		
4		
TOTAL DE INGRESOS ESTIMADOS		1200.00

DETALLE DE EGRESOS

RECURSO	TIPO DE RECURSO	TIPO DE UNIDAD	CANTIDAD	PRECIO POR UNIDAD	COSTO TOTAL
HUMANO	IMPRESIÓN 3D	PIEZA	4	40.00	160.00
	FABRICACIÓN PCB	PIEZA	5	6.50	32.50
	ASESORIA PLATAFORMA WEB	REUNIONES	2	35.00	70.00
	CURSO CREACIÓN PÁGINA WEB	SUSCRIPCIÓN	2	120.00	240.00
ECONÓMICO / FINANCIERO	Papel bond	Resma	1	3.50	3.50
	Tinta	Envase	5	5.00	25.00
	Transporte público	Paseo	100	0.35	35.00
	Recarga chip número de cta Gmail.	Saldo	4	5.15	20.60
		Propuesta		422.57	
TECNOLOGICO	Pago mensual VPS Goggle Cloud	Servicio	5	16.00	80.00
	Pago anual reserva de dominio DNS	Servicio	1	12.00	12.00
	Pago anual protección dominio	Cotización	1	9.00	9.00
		Cotización	1	422.57	422.57
TOTAL DE GASTOS ESTIMADOS					1110.17

PRE SUPUESTO DEL PROYECTO

Tipo de Proyecto: _____

DURACIÓN: 4 MESES

INGRESOS ESTIMADOS	1200.00
Costos directos	1110.17
Costos indirectos	25%
Reserva para riesgos	15%

DEFICIT	395.87
Presupuesto	1387.71
Riesgo	208.16
TOTAL	1595.87

DETALLE DE INGRESOS

N°	DESCRIPCIÓN	APORTE ECONÓMICO
1	AMAGUADARIO	600.00
2	GAVILANES FERNANDO	600.00
3		
4		
TOTAL DE INGRESOS ESTIMADOS		1200.00

DETALLE DE EGRESOS

RECURSO	TIPO DE RECURSO	TIPO DE UNIDAD	CANTIDAD	PRECIO POR UNIDAD	COSTO TOTAL
HUMANO	IMPRESIÓN 3D	PIEZA	4	40.00	160.00
	FABRICACION PCB	PIEZA	5	6.50	32.50
	ASESORIA PLATAFORMA WEB	REUNIONES	2	35.00	70.00
	CURSO CREACION PAGINA WEB	SUSCRIPCION	2	120.00	240.00
ECONÓMICO / FINANCIERO	Papel bond	Resma	1	3.50	3.50
	Tinta	Envase	5	5.00	25.00
	Transporte público	Pasaje	100	0.35	35.00
	Recarga chip numero de cta Gmail.	Saldo	4	5.15	20.60
		Propuesta		422.57	
TECNOLÓGICO	Pago mensual VPS Goggle Cloud	Servicio	5	16.00	80.00
	Pago anual reserva de dominio DNS	Servicio	1	12.00	12.00
	Pago anual proteccion dominio	Cotización	1	9.00	9.00
		Cotización	1	422.57	422.57
TOTAL DE GASTOS ESTIMADOS					1110.17