



SISTEMA DE MONITOREO Y DETECCIÓN DE FALLOS EN UNA MICRORRED ELÉCTRICA UTILIZANDO MEDIDORES INTELIGENTES

Autores:

**Nicolás Andrés Fresneda B.
Johan Fernando Gómez O.**

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería, Proyecto curricular de Ingeniería Electrónica
Febrero 2022

**SISTEMA DE MONITOREO Y DETECCIÓN DE FALLOS EN
UNA MICRORRED ELÉCTRICA UTILIZANDO
MEDIDORES INTELIGENTES**

**Nicolás Andrés Fresneda B.
Johan Fernando Gómez O.**

Trabajo de grado como requisito parcial para optar al título de:
Ingeniero Electrónico

Modalidad:
Investigación/Innovación

Director:
Ing. Elvis Eduardo Gaona G., PhD.

Grupo de investigación:
**Grupo de Investigación en Telecomunicaciones de la Universidad
Distrital (GITUD)**

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería, Proyecto curricular de Ingeniería Electrónica
Febrero 2022

Dedicado a mi familia, mis amigos, mis compañeros y todos aquellos que me apoyaron durante este largo trayecto, que me ayudaron a levantarme y no darme por vencido.

Nicolás Andrés Fresneda B.

A mi familia, docentes y compañeros, por hacer de este camino algo posible.

Johan Fernando Gómez O.

Agradecimientos

Este proyecto es destinado directamente a nuestras familias, a la Universidad Distrital Francisco José de Caldas, a nuestros docentes, compañeros, y específicamente a nuestro director de proyecto Elvis Eduardo Gaona, a quienes agradecemos de manera expresa, la oportunidad de brindarnos los conocimientos académicos, formación profesional, personal y un espacio de crecimiento integral a lo largo de estos años para afrontar nuevos retos y proyectos profesionales. Este proyecto se realiza en agradecimiento a ustedes.

ÍNDICE

1. Generalidades	1
1. Introducción	1
2. Justificación	2
2.1. Académica	2
2.2. Investigativa	2
2.3. Social	2
2.4. Económica	2
2.5. Ambiental	3
2.6. Personal	3
3. Planteamiento de problema	3
4. Objetivos	4
2. Marco de referencia	5
1. Marco teórico	5
1.1. Micro Red Inteligente	5
1.2. Medidor inteligente	6
1.3. ONION OS	10
1.4. Fallas eléctricas	12
2. Estado del arte	16
2.1. Medición inteligente y funcionalidades de medidores inteligentes en redes inteligentes: Una revisión	16
2.2. Medición inteligente de energía y control de robo de energía con Arduino y GSM	17
2.3. Medidor de energía inteligente basado en IoT para una utilización eficiente de la energía en la red inteligente	19
2.4. Un sistema de configuración para monitoreo y control en tiempo real del comportamiento del consumo de electricidad	19
2.5. Estudio sobre una red inteligente y el sistema de tecnología clave para impulsar el desarrollo de la generación distribuida de energía.	20
2.6. Modelos de software para una red inteligente	21
2.7. Aplicación de la comunicación por línea eléctrica en consumo de energía inteligente	22
2.8. Smart Grid: una evaluación de oportunidades y desafíos en el despliegue del sistema de energía de Ghana	23

3. Detección de fallas	24
1. Variaciones lentas y fluctuaciones de tensión	24
2. Distorsión armónica y variaciones de frecuencia	25
3. Fluctuaciones de corriente	27
4. Fallas de baja impedancia o cortocircuito	28
4. Medidor inteligente	31
1. Acceso al medidor inteligente	31
1.1. Conexiones físicas	31
1.2. Acceso al sistema operativo del medidor - Acceso a la red WiFi	32
2. Configuración del medidor	37
2.1. Configuración técnica inicial	37
2.2. Instalación de complementos y librerías	39
3. Códigos implementados	40
3.1. Captura de los parámetros de manera serial	41
3.2. Detección del estado para los parámetros de la red	43
3.3. Imprimir en pantalla cada parámetro de la red	44
3.4. Estructuración y transferencia del archivo en formato JSON	46
3.5. Comunicación con las bases de datos desde el medidor	48
5. Servidor web	55
1. Características técnicas del servidor web	55
2. Configuración servidor web - APACHE	55
3. MySQL	57
3.1. Configuración adicional de MySQL	57
4. PHP	58
5. PhpMyAdmin	59
5.1. Proceso de instalación PhpMyAdmin	59
6. Bases de datos	61
6.1. Base de datos para los usuarios	62
6.2. Base de datos para los datos capturados	63
7. Habilitación HTTPs	64
7.1. Certificado SSL	64
6. Aplicativo web	66
1. Página de inicio de sesión	66
2. Página de registro de usuarios	68
3. Página de modificación de contraseña	71
4. Página principal para estudiantes	73
5. Página principal para docentes	77
6. Página principal para supervisores	79
7. Implementación, pruebas y resultados	83
8. Conclusiones	90
Referencias	93

Índice de figuras

2.1.	Diagrama simple de una microrred eléctrica. Tomada de: [3]	6
2.2.	Hardware del medidor eGeo Analyzer. Tomada de: [6].	8
2.3.	Diagrama de conexión unifilar. Tomada de: [7].	9
2.4.	Transformador de corriente de núcleo partido Sensor de corriente alterna KCT-16. Tomada de: [9].	10
2.5.	Terminal de Onion OS. Tomada de: [10].	11
2.6.	Hardware de la placa de desarrollo Omega2+. Tomada de: [11].	11
2.7.	Especificaciones técnicas de la placa de desarrollo Omega2+. Tomada de: [11].	12
2.8.	Tipos de fallas en derivación o de baja impedancia. Adaptada de: [14].	15
2.9.	Diagrama de las fallas eléctricas detectables.	15
2.10.	Diagrama de las fallas eléctricas no detectables.	16
2.11.	Sistema de gestión de datos del medidor (MDMS). Tomada de: [16].	17
2.12.	Diagrama de la arquitectura del sistema del medidor de energía inteligente basado en Arduino y GSM. Tomada de: [17].	18
2.13.	Diagrama de bloques funcional del medidor inteligente. Tomada de: [18].	19
2.14.	Proyecto de sistema de configuración del esquema. Tomada de: [19].	20
2.15.	Esquema de la red inteligente implementada. Tomada de: [20].	21
2.16.	Interconexión entre varios componentes. Tomada de: [21].	22
2.17.	Diagrama de la red inteligente empleada en la aplicación. Tomada de: [22].	23
3.1.	Diagrama de flujo del algoritmo para la detección de las fallas relacionadas a la tensión monofásica.	25
3.2.	Diagrama de flujo del algoritmo para la detección del estado de los valores de THD en tensión y corriente.	26
3.3.	Diagrama de flujo del algoritmo para la detección de las variaciones de frecuencia.	27
3.4.	Diagrama de flujo del algoritmo para la detección de las fluctuaciones de corriente.	28
3.5.	Diagrama de flujo del algoritmo para la detección y clasificación de fallas de baja impedancia o cortocircuito.	29
4.1.	Diagrama de conexión del medidor inteligente a la red eléctrica. Adaptada de: [6].	32
4.2.	Acceso a la red WiFi del medidor B6FF.	32
4.3.	Interfaz de inicio de acceso al medidor inteligente B6FF.	33

4.4. Selección de la red WiFi que conectará a internet al medidor inteligente B6FF	34
4.5. Actualización del firmware para el medidor inteligente B6FF	34
4.6. Acceso al gestor de aplicaciones para el medidor inteligente B6FF	35
4.7. Captura del terminal accediendo al medidor utilizando comunicación SSH	35
4.8. Configuración de los parámetros de conexión necesarios para la comunicación SSH sobre PuTTY.	36
4.9. Conexión establecida con el medidor desde la consola de PuTTY.	36
4.10. Opción gráfica de restauración al estado de fábrica predeterminada del medidor.	38
4.11. Confirmación de codificación de contraseña SHA1 desde el medidor al servidor web.	40
 5.1. Información del ‘status’ del firewall configurado para el servidor web.	56
5.2. Respuesta de confirmación del estado del servidor web Apache.	56
5.3. Respuesta de confirmación del estado de la base de datos MySQL.	57
5.4. Respuesta de confirmación del estado de la base de datos MySQL.	58
5.5. Comprobación de versión de PHP.	58
5.6. Selección de Apache2 como servidor web.	59
5.7. Comprobación de versión de PHP.	59
5.8. Solicitud de contraseña de aplicación entre MySQL y PhpMyAdmin.	60
5.9. Interfaz de PhpMyAdmin.	60
5.10. Diagrama de la distribución de las bases de datos implementadas en el servidor web.	61
 6.1. Captura de la vista de escritorio de la página de inicio de la interfaz.	67
6.2. Mensaje de alerta por parte de la página de inicio cuando se ingresan las credenciales de acceso incorrectas.	67
6.3. Captura de la vista de escritorio de la página de registro de la interfaz.	68
6.4. Mensaje de alerta por parte de la página de registro cuando se presiona el botón de registrarse sin haber llenado todos los campos de texto.	69
6.5. Mensaje de alerta por parte de la página de registro cuando se ingresa un correo con un dominio no válido.	69
6.6. Mensaje de alerta por parte de la página de registro cuando se detecta que la contraseña y la confirmación no coinciden.	69
6.7. Mensaje de confirmación por parte de la página de registro mostrando los datos ingresados.	70
6.8. Mensaje de alerta por parte de la página de registro cuando se registra el usuario exitosamente.	70
6.9. Mensaje de alerta por parte de la página de registro cuando el usuario ingresado ya existe en la base de datos.	70
6.10. Mensaje de alerta por parte de la página de registro cuando se presentan errores de almacenamiento en el servidor o en la base de datos.	71
6.11. Captura de la vista de escritorio de la página de modificación de contraseña de la interfaz.	71
6.12. Mensaje de confirmación por parte de la página de registro mostrando los datos ingresados.	72

6.13. Mensaje de alerta por parte de la página de registro cuando se modifica la contraseña del usuario exitosamente.	73
6.14. Mensaje de alerta por parte de la página de registro cuando el usuario ingresado no existe en la base de datos.	73
6.15. Mensaje de alerta por parte de la página de registro cuando los datos ingresados no coinciden con los almacenados en la base de datos.	73
6.16. Captura de la vista de escritorio de la página principal para estudiantes del aplicativo web.	74
6.17. Captura de la vista de escritorio del menú desplegable de la página principal del aplicativo web.	74
6.18. Contenedor del gráfico de tensión para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para estudiantes.	75
6.19. Contenedor del gráfico de tensión acercado para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para estudiantes.	76
6.20. Contenedor del gráfico de frecuencia para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para docentes.	77
6.21. Gráfico maximizado de potencia para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para docentes.	78
6.22. Gráfico maximizado de potencia para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para supervisores.	79
6.23. Pie de página de la interfaz gráfica y botón de mostrar tabla de estados en la página principal para supervisores.	80
6.24. Tabla de valores y botón de mostrar/ocultar la tabla de estados en la página principal para supervisores para la temperatura del chip 1 del medidor de sistemas.	81
6.25. Página de visualización del listado de fallas críticas detectadas por el sistema.	82
 7.1. Topología de red de la prueba trifásica implementada	83
7.2. Implementación real del medidor para la red trifásica a analizar.	84
7.3. Datos reales obtenidos sobre la consola del medidor correspondientes a la red trifásica	85
7.4. Gráfica de potencia registrada sobre las pruebas.	85
7.5. Gráfica de tensión registrada sobre las pruebas.	86
7.6. Gráfica de corriente registrada sobre las pruebas.	86
7.7. Gráfica de frecuencia registrada sobre las pruebas.	87
7.8. Gráfica de temperatura registrada sobre las pruebas.	87
7.9. Gráfica de distorsión armónica de tensión registrada sobre las pruebas.	88
7.10. Gráfica de distorsión armónica de corriente registrada sobre las pruebas.	88
7.11. Gráfica de fases para la red trifásica registrada sobre las pruebas.	89

Índice de tablas

5.1. Distribución de los puertos TCP para la comunicación con las bases de datos.	62
5.2. Estructura de la tabla creada para el almacenamiento de los datos de los usuarios.	62
5.3. Estructura de la tabla creada para el almacenamiento de los datos capturados para una variable dependiente de las tres líneas.	63
5.4. Estructura de la tabla creada para el almacenamiento de los datos capturados para una variable independiente de las líneas.	63

Capítulo 1

Generalidades

1. Introducción

En el presente proyecto se describe el diseño e implementación de un sistema de monitoreo y detección de fallas que se presentan en una microrred eléctrica de la Universidad Distrital Francisco José de Caldas, que comprende tres puntos de medición, dos puntos de consumo y un punto de generación de energía, utilizando como herramienta principal el módulo de medición inteligente eGeo Analyzer, el cuál realiza la medición de tensión, corriente, frecuencia, temperatura, THD, potencia, ángulos de fase, entre otros; sin la necesidad de tener que realizar conexiones invasivas en la red. Este módulo será la base fundamental del proyecto, dado que obtendrá los parámetros necesarios de la red, y de esta forma, permitirá realizar el procesamiento de los datos necesarios que permita la detección de fallos en la microrred.

El sistema de detección de fallos tendrá una base de datos que almacenará en tiempo quasi-real la información obtenida del medidor inteligente, además será controlada y ajustada por diferentes tipos de usuarios, para lo cual se hace necesario el desarrollo de una interfaz gráfica que sea amigable con estos usuarios y que proporcione de manera sencilla los datos de interés para cada usuario. Esto implica de manera directa la elaboración de jerarquías, que permitan restringir la información, a partir del tipo de usuario que quiera acceder a la base de datos, como por ejemplo un supervisor, un docente o un estudiante.

Adicional a esto, se tendrá la capacidad de desactivar la red en caso de presentarse una falla crítica, por lo tanto, se deberá implementar el módulo de control que permita activar o desactivar la red de manera automática y manual empleando tecnologías en comunicaciones y sistemas de potencia.

2. Justificación

2.1. Académica

Con este proyecto, se busca realizar un desarrollo académico, en el sentido de que no se dispone conocimiento acerca de medidores y redes inteligentes, métodos de enlaces de sistemas de comunicaciones, manejo de datos en la nube, etc. Para lo cual resulta importante para los autores tener la oportunidad de aprender, profundizar y actualizar estos conocimientos, con base en los fundamentos de la teoría de la electrónica, por supuesto. De esta manera, el proyecto es un impulso a la inmersión de nuevos conocimientos, lo cual, es uno de los principios fundamentales de la academia, el impulsar al estudiante a asumir nuevos retos y resolverlos.

2.2. Investigativa

Debido al desarrollo de generación de energías limpias que se viene desarrollando actualmente en nuestra sociedad, resulta importante la investigación en el campo de redes y medidores inteligentes para brindar soporte y herramientas que puedan ayudar a la fomentación de estas tecnologías.

2.3. Social

El proyecto como tal, es un aporte directo que se desarrolla para la Universidad Distrital Francisco José de Caldas, la componente social del proyecto, es la oportunidad de brindarle a la Universidad un sistema tecnológico que le permita monitorear y detectar fallos dentro de su red eléctrica (para este caso en específico sobre tres puntos estratégicamente seleccionados), además, a partir de estas ubicaciones, detectar posibles riesgos que puedan afectar directa o indirectamente el bienestar y la integridad de la comunidad universitaria, los equipos tecnológicos y sus instalaciones físicas.

2.4. Económica

Al implementar un sistema de monitoreo y detección de fallos en la red eléctrica de la Universidad, de manera directa pero a largo plazo, se está favoreciendo a sí misma, de manera económica. Se favorece en el sentido de que se tiene un registro de la energía que se consume y se produce, de manera que se puede encontrar que componentes se deben reconfigurar para buscar una mayor eficiencia energética; además de que se evitan en lo posible las fallas que puedan causar daños a los equipos, los cuales en su mayoría resultan ser bastante costosos. Se habla de a largo plazo, porque se debe realizar una inversión en la adquisición y mantenimiento de los módulos (medidores inteligentes) utilizados en el proyecto, lo cual debe ser contabilizado y estudiado para contrastar la relación costo/beneficio, comparándolo con el ahorro monetario que se puede obtener al ahorrar energía eléctrica y al prevenir daños físicos en equipos e instalaciones de la Universidad.

2.5. Ambiental

Actualmente, uno de los principales problemas que enfrenta el planeta tierra es la contaminación ambiental, siendo el sector industrial uno de los actores principales en la producción de partículas y materia contaminante. El problema que se busca atacar con el proyecto, es el evitar las fugas y pérdidas de energía dentro de la red eléctrica, mediante diferentes parámetros que se obtendrán con el sistema de monitoreo; de esta manera se busca aumentar la eficiencia energética, evitando el desperdicio de energía y la sobre-distribución en las plantas de energía.

2.6. Personal

Para nosotros, resulta importante el poder desarrollar este proyecto incluyendo a la universidad de manera directa, aportando así un pequeño desarrollo dentro del campo de las redes inteligentes y las microrredes. De esta manera, poder vigilar nuestros equipos y la calidad de la energía de la red que nosotros mismos estamos utilizando, obteniendo información necesaria para tomar medidas correctivas que mejoren el consumo energético en la universidad, y por supuesto, dejar abierto un espacio de desarrollo e investigación por medio del proyecto y el grupo de investigación GITUD, para que en el futuro docentes y estudiantes puedan seguir desarrollando temas asociados.

3. Planteamiento de problema

En la actualidad, una de las mayores fuentes de consumo de recursos por los seres humanos es la energía eléctrica, generando así la necesidad de abastecer a una gran parte de todas las personas que habitan el planeta tierra. El sector industrial por su parte, es uno de los principales productores y consumidores de energía eléctrica, dependiendo directamente de una fuente de energía funcionando durante jornadas continuas. En este orden de ideas, se hace bastante tedioso el encargar a distintas personas el constante monitoreo y vigilancia, con el fin de prevenir posibles errores y detectar fallas dentro de las redes eléctricas.

Recientemente, las redes de distribución han venido adaptando medidores inteligentes, con el fin de garantizar la entrega de energía bajo diferentes estándares de calidad. El problema radica en que esta energía llega hasta el usuario, bajo altos estándares de calidad, sin embargo, al ingresar a la planta o a la residencia del usuario, se pierde la certeza de como es el comportamiento de la energía durante el suministro de la corriente hacia los diferentes dispositivos dentro de la red. Además de que presenta bastante dificultad, el proceso de medir en tiempo real o quasi-real los parámetros de la red eléctrica, para de esta manera obtener información de su comportamiento.

Lo anterior se traduce en que no se tiene completa certeza, acerca de que es lo que está pasando con la red de los distintos usuarios, adicionalmente a ello, si se llegase a presentar algún tipo de falla, se hace bastante complicado e ineficiente la acción de estar continuamente revisando ¿Dónde se encuentra la falla?, ¿Qué tipo de falla se presentó?,

¿En qué momento y por qué causas sucedió la falla?, etc. Hasta el punto en que en algunos casos, resulta más laborioso el encontrar la falla, que su solución misma. Por ende, se requiere de una solución eficiente que ayude a monitorear y vigilar, qué es lo que está sucediendo con los parámetros de las redes eléctricas dentro de las plantas y residencias.

Siendo este sistema de monitoreo en tiempo quasi-real el reto a cumplirse, ¿Cuál sería el sistema de medición y detección de fallas apto para el monitoreo en tiempo quasi-real de la microrred de la Universidad Distrital Francisco José de Caldas?

4. Objetivos

General

Diseñar e implementar un sistema de monitoreo y detección de fallos para la microrred eléctrica de la Universidad Distrital Francisco José de Caldas, en la facultad de Ingeniería, a partir de medidores inteligentes.

Específicos

- Identificar las funciones, alcances y limitaciones del módulo eGeo Analyzer, como medidor inteligente.
- Diseñar e implementar un sistema embebido que permita la comunicación entre el medidor inteligente y un sistema de conmutación de la red eléctrica.
- Estructurar la base de datos y la interfaz gráfica del sistema de monitoreo de fallas que esté soportada por almacenamiento en la nube.
- Realizar pruebas de precisión, validación de los datos adquiridos y análisis de los resultados.

Capítulo 2

Marco de referencia

1. Marco teórico

1.1. Micro Red Inteligente

Una Micro Red Inteligente es un sistema de generación eléctrica bidireccional que permite la distribución de electricidad desde los proveedores hasta los consumidores, utilizando tecnología digital y favoreciendo la integración de las fuentes de generación de origen renovable, con el objetivo de ahorrar energía, reducir costes e incrementar la fiabilidad. Los elementos de los que consta son: Sistemas de generación distribuida; sistemas de almacenamiento de energía; técnicas para la gestión de cargas; sistemas de monitorización y control del flujo de potencia; y técnicas y procedimientos de mantenimiento preventivo [1].

La Micro-Red es un sistema de interconexión con capacidad de auto-abastecerse y operar de forma aislada si es necesario. Incluye tanto la generación, como el almacenamiento y el transporte eléctrico, además de los equipos para optimizar la gestión inteligente de la energía, por parte del usuario final (alumbrado público, automatización de edificaciones, entre otros) [2].

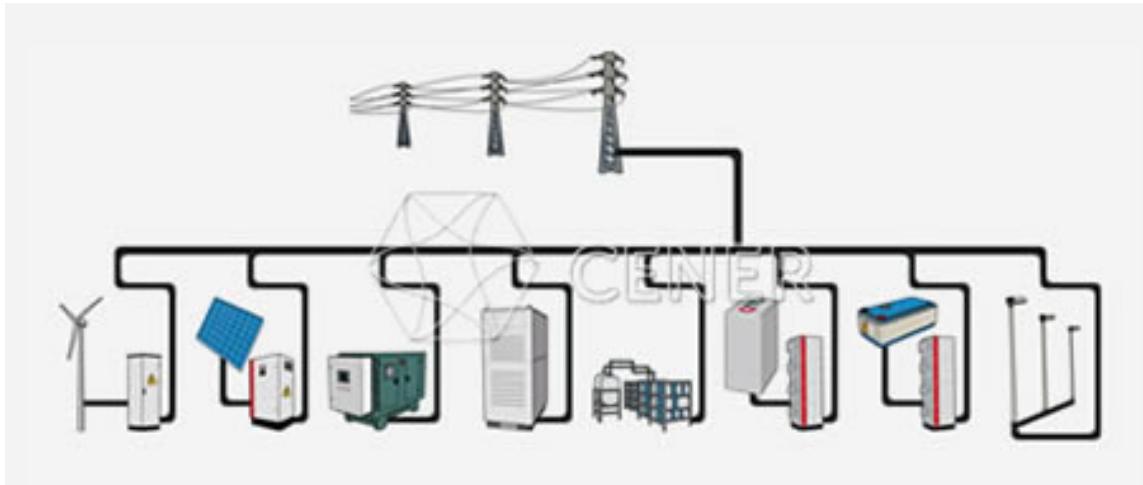


Figura 2.1: Diagrama simple de una microrred eléctrica. Tomada de: [3]

La microrred es gestionada por un controlador central que está a la cabeza del sistema jerárquico de control. Este controlador central (MGCC) proporcionará las consignas a los controladores del resto de los equipos, tales como fuentes de generación, sistemas de almacenamiento de energía y cargas inteligentes. La microrred podrá funcionar de dos modos distintos: conectado a la red principal y aislada de la misma en caso de existir algún problema en esta última [3].

Cuando la microrred funciona en modo conectado con la red principal, ésta proporcionará las referencias de tensión y frecuencia necesarias para que el resto de elementos de generación de la microrred funcionen sin ningún problema. Así pues, no existirán problemas de estabilidad en la microrred cuando ésta funcione conectada a la red principal. Cuando la microrred funciona en modo aislado de la red principal, los generadores tienen que ser capaces de responder con rapidez a los cambios en el consumo para que así tanto la tensión como la frecuencia se mantengan estables. Al tratarse de equipos con interfaces de electrónica de potencia los generadores de la microrred no tienen inercia para asumir los desequilibrios puntuales entre generación y consumo del modo que ocurre en los sistemas eléctricos convencionales con los grandes generadores síncronos [3].

1.2. Medidor inteligente

Es un dispositivo electrónico inteligente que constituye la evolución de los medidores de energía eléctrica tradicionales. En los medidores inteligentes se integran funciones metrológicas avanzadas, almacenamiento y procesamiento de datos a nivel local, comunicaciones bidireccionales con otros dispositivos y una interfaz de usuario. Para aprovechar todas las utilidades de este dispositivo es necesario que opere dentro de una Infraestructura de medición avanzada [4].

Un medidor inteligente o Smart Meter por su nombre en inglés, es un contador electrónico inteligente capaz de registrar el consumo de energía eléctrica en pequeños intervalos de tiempo (comúnmente inferiores a una hora) y comunicar la información procedente de

ese registro a la unidad de control. Se trata, por tanto, de un instrumento que cumple la función de un sensor y de un punto de interacción simultáneamente, de un nodo inteligente [5].

Los beneficios que aporta el uso de Smart Meter a una red de suministro eléctrico son muy diversos, entre ellos destacan la detección de averías por parte de las compañías, así como los beneficios económicos que reportan del control y gestión de la información proporcionada. Pero el uso de estos aparatos supone también un gran avance para los clientes de la red, puesto que les permite monitorizar su consumo casi en tiempo real. Como sabemos, el conocimiento del consumo propio es uno de los pilares más importantes de cara a favorecer la eficiencia energética [5].

1.2.1. Medidor Inteligente eGEO Analyzer

eGEO Analyzer es un medidor inteligente desarrollado por la compañía eGeo, el cual tiene la capacidad de poder medir los parámetros de la red eléctrica como por ejemplo: frecuencia, voltaje, corriente, factor de potencia y energía de armónicos, entre otros. Es también, un contador de energía eléctrica con precisión 0.2S de energía activa y reactiva, inversa y directa de hasta 6 circuitos monofásicos o 2 circuitos trifásicos [6].

Cuenta con una plataforma de desarrollo muy poderosa con conexión Wi-Fi y Ethernet para enviar datos a través de la red de una forma rápida y segura. También permite hacer desarrollos propios de metrología y/o como concentrador de datos en C++, Python, Node, entre otros [6]. El hardware que utiliza el medidor inteligente se muestra a continuación:

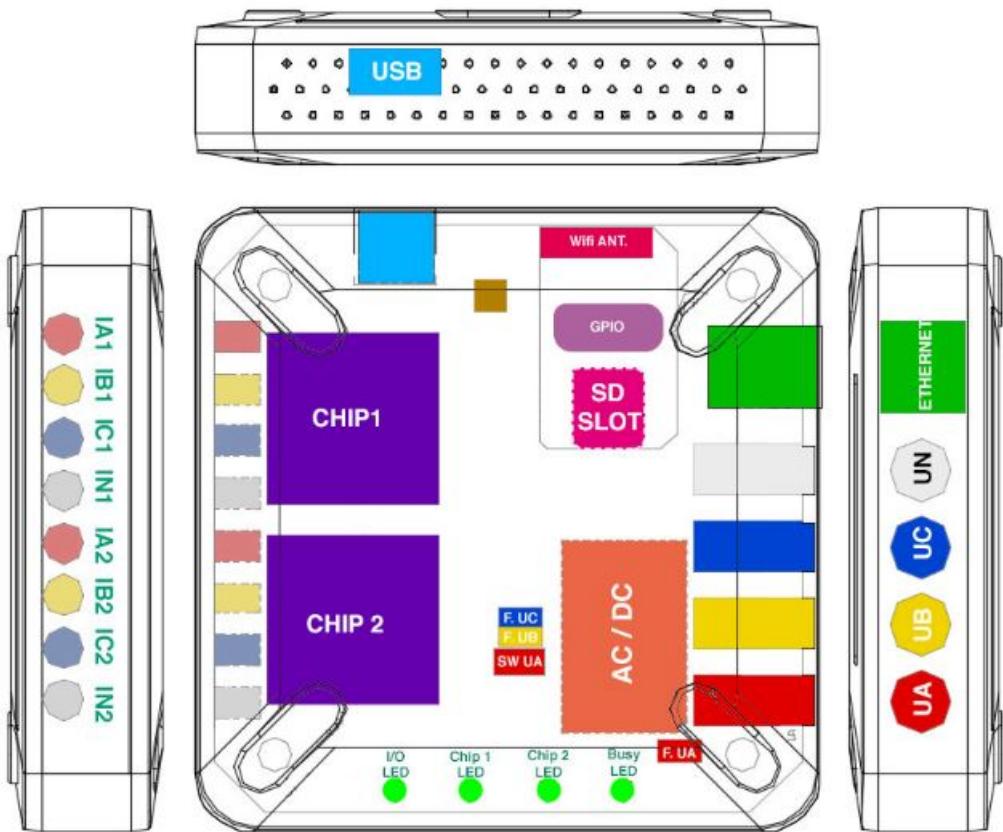


Figura 2.2: Hardware del medidor eGeo Analyzer. Tomada de: [6].

En la figura 2.2 se muestra el hardware interno del medidor, junto con las vistas laterales del mismo. Las especificaciones de este hardware son:

- Voltaje: 85 V – 305 V (AC) (Fase a tierra).
- Corriente: 0.05 A - 120 A (AC) por entrada.
- Pulsos de energía: 1600 imp/kWh.
- Ethernet: 10/100 Mbps.
- Wi-Fi: Banda 2.4 GHz.
- SD tipo de slot: Micro SD.

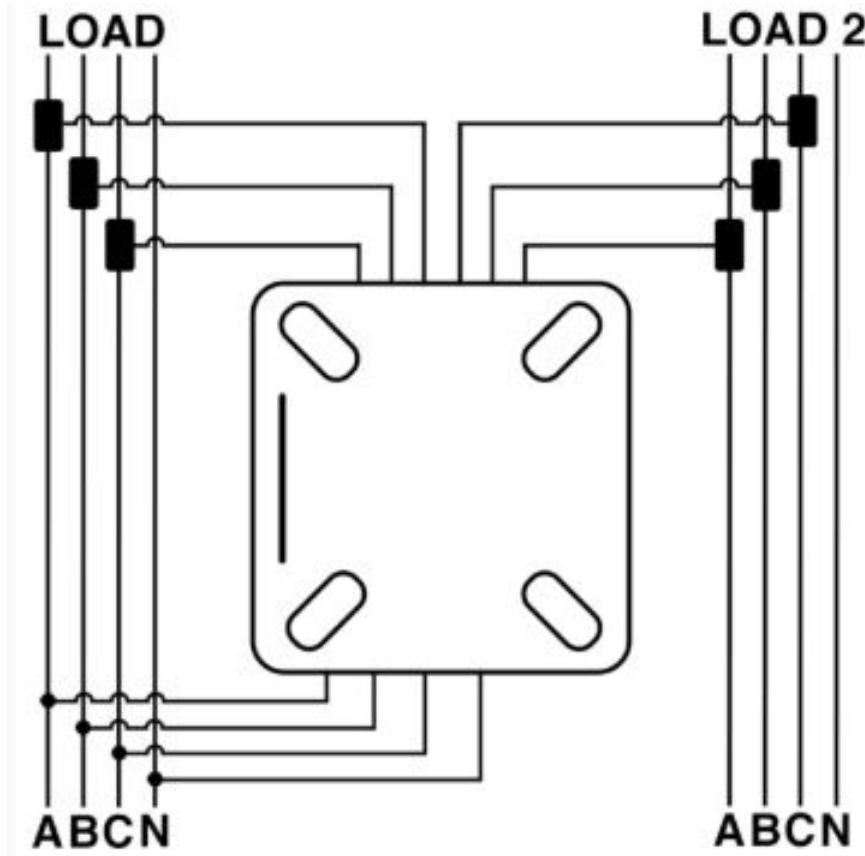


Figura 2.3: Diagrama de conexión unifilar. Tomada de: [7].

En la figura 2.3 se muestra el diagrama de conexiones que se puede realizar con el medidor, para la adquisición de los datos, como se observa, el medidor es capaz de medir en paralelo 3 fases de tensión con su respectivo neutro, junto con 6 medidas distintas de corriente, a partir de las bobinas por efecto Hall [7].

1.2.2. Sensores por efecto Hall

En términos generales, un sensor de efecto Hall es un transductor que, ante la presencia de un campo magnético, proporciona diferentes mediciones de voltaje. Así, los sensores de efecto Hall permiten medir campos magnéticos, corrientes continuas y una gran diversidad de fenómenos de distintos usos y a diferentes escalas. Existen dos tipos de sensores de efectos Hall: los sensores análogos y los sensores digitales. Los sensores de efecto Hall análogos son apropiados particularmente para ser usados como sensores de proximidad y sensores de corriente, mientras que los sensores de efecto Hall digitales son usados como interruptores de límite o sensores de contacto. Otra subdivisión entre estos productos está en los sensores de lazo cerrado y los sensores de lazo abierto. Los sensores de efecto Hall de lazo cerrado tienen por lo general un precio más alto, pero son más precisos en la medición; también responden de manera más rápida frente a los sensores de lazo abierto. Estos últimos, en cambio, tienen la capacidad de recibir altas cargas de corriente para su medición y responden a frecuencia de una manera bastante efectiva [8].



CE

RoHS

**Hole size:
16mm**



KCT-16

Figura 2.4: Transformador de corriente de núcleo partido Sensor de corriente alterna KCT-16. Tomada de: [9].

En la figura 2.4 se observa los sensores por efecto Hall que se utilizarán para la adquisición de los datos de corriente, esta figura junto con sus características, se encuentran directamente en el sitio del fabricante [9].

1.3. ONION OS

eGeo analyzer esta soportado por Onion Omega2+ y Onion OS, por consiguiente, para interactuar con el medidor inteligente se debe comprender el uso de la consola de Onion OS y sus instrucciones.

La interfaz de línea de comando (CLI) es el punto de acceso del usuario al sistema operativo mediante una terminal. El sistema operativo interpreta y ejecuta toda la interacción del usuario a través de líneas de comandos en la terminal. De esta manera, un usuario ingresa un comando en una terminal para hacer que algo suceda [10]. Los comandos de esta terminal se envían mediante el lenguaje Linux, como se muestra en la figura 2.5.

```

COM4 - PuTTY
root@Omega-1CAE:/# login
Omega-1CAE login: root
Password:

BusyBox v1.23.2 (2016-01-15 18:09:58 UTC) built-in shell (ash)


-----
Q-ware: 0.0.6 b266
-----
root@Omega-1CAE:~#

```

Figura 2.5: Terminal de Onion OS. Tomada de: [10].

1.3.1. ONION omega2+

El Omega2+ es lo último en placas de desarrollo de Onion. Viene equipado con WiFi incorporado, un sistema operativo Linux, GPIO flexibles, una ranura MicroSD y un potente procesador (Mediatek MT7688) [11].

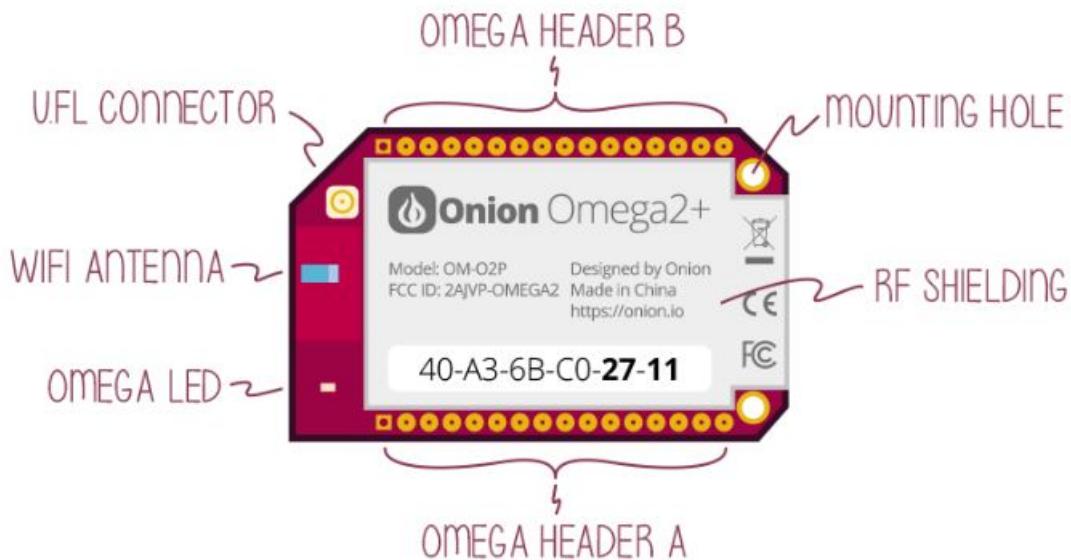


Figura 2.6: Hardware de la placa de desarrollo Omega2+. Tomada de: [11].

En la figura 2.6 se muestra el hardware de la placa de desarrollo, junto con la especificación de los puertos externos. Las especificaciones técnicas se muestran en la figura 2.7.

Omega2+ Specs	
Processor	580MHz MIPS CPU
Memory	128MB Memory
Storage	32MB Storage
USB	USB 2.0
MicroSD Slot	Yes
WiFi adapter	b/g/n Wi-Fi
GPIOs	18
PWM	2
UART	2
I2C	1
SPI	1
I2S	1

Figura 2.7: Especificaciones técnicas de la placa de desarrollo Omega2+. Tomada de: [11].

1.4. Fallas eléctricas

Una falla eléctrica se define un evento anormal que provoca el mal funcionamiento de un circuito eléctrico [12]. Dentro de una red eléctrica, la falla puede propagarse a lo largo de las conexiones, provocando irregularidades y comportamientos ajenos al sistema. Un sistema eléctrico se considera que está operando en forma normal cuando sus parámetros eléctricos característicos se encuentran dentro de los límites establecidos por la reglamentación correspondiente, y estos satisfacen las exigencias de los fabricantes de los equipos conectados a la red [13].

Las fallas eléctricas más relevantes se presentan a continuación:

1.4.1. Variaciones en la frecuencia

Se dice que existen variaciones de frecuencia en un sistema eléctrico de corriente alterna cuando se produce una alteración del equilibrio entre carga y generación. La frecuencia, en un sistema eléctrico de corriente alterna, está directamente relacionada con la velocidad de giro, es decir, con el número de revoluciones por minuto de los alternadores. Dado que la frecuencia es común a toda la red, todos los generadores conectados a ella girarán de manera síncrona, a la misma velocidad angular eléctrica. [13]

En los márgenes normales de tolerancia, el principal efecto de las variaciones de frecuencia es el cambio en la velocidad de las máquinas rotativas. En tales condiciones, pueden producirse los siguientes fenómenos:

- Los motores transmiten más o menos potencia.
- Los relojes eléctricos sincronizados con la red, atrasan o adelantan.

- Los filtros de armónicos sufren un efecto distorsionador.
- Los equipos electrónicos que utilizan la frecuencia como referencia de tiempo se ven alterados.
- Posibles problemas en el funcionamiento de instalaciones de auto-generación.

1.4.2. Variaciones lentas de tensión

Se produce una variación de tensión cuando hay una alteración en la amplitud de la señal de la red eléctrica y, por lo tanto, el valor eficaz de la onda de tensión. [13]

- **Efectos de baja tensión:** La mayor parte de los receptores pasan de un estado de funcionamiento normal a uno “anómalo” o a uno de “no funcionamiento” cuando se ven sometidos a una tensión baja, recuperando el estado “normal” cuando el valor de la tensión vuelve a situarse dentro de los márgenes de tolerancia. Así, en la mayoría de los casos, los efectos no suelen ser especialmente perjudiciales. Por ejemplo, en el momento del arranque, un motor no podrá iniciar el giro si la tensión no es suficiente para proporcionar el par mecánico que requiere el eje. Sufrirá un calentamiento que podría provocar su avería. Así mismo, en contactores o relés se pueden producir actuaciones incorrectas, afectando al proceso que estén controlando.
- **Efectos de tensión alta:** La tensión alta produce fundamentalmente un efecto de calentamiento de los receptores. En determinadas circunstancias, este calentamiento puede ocasionar la avería de los equipos si se supera el límite térmico que toleran. Es más difícil detectarla, ya que los receptores no dejan de funcionar instantáneamente y no es fácil apreciar de inmediato su sobrecalentamiento.

1.4.3. Fluctuaciones de tensión

Se dice que hay fluctuaciones de tensión cuando se producen variaciones periódicas o series de cambios aleatorios en la tensión de la red eléctrica. A su vez, las variaciones de tensión se definen como las variaciones del valor eficaz o valor de pico de tensión entre dos niveles consecutivos que se mantienen durante un tiempo finito no especificado. Su duración va desde varios milisegundos hasta unos 10 segundos y con una amplitud que no supera el 10 % del valor nominal. [13]

1.4.4. Impulsos o sobre-picos de tensión

Un impulso de tensión es una variación brusca del valor instantáneo de la amplitud de la tensión. Puede llegar a ser varias veces superior al valor nominal de ésta y su duración oscila entre algunos microsegundos y diez milisegundos, lo que equivale a medio ciclo de la onda sinusoidal. Por su amplitud y duración, los impulsos de tensión tienen que ser analizados a partir de valores instantáneos de la amplitud de la onda de tensión y no mediante valores promediados, que son los que habitualmente se utilizan para medir otro tipo de perturbaciones que afectan a la amplitud de la onda. Los impulsos de tensión suelen aparecer de forma esporádica, pero es posible también que se repitan a lo largo

del tiempo. Pueden manifestarse en cualquier punto de la red. A partir de éste, tienden a desplazarse a lo largo de la misma con la velocidad de propagación de una onda en un medio conductor. Por ello, en la práctica, suele considerarse que aparecen en todos los puntos de dicha red en el mismo instante en el que es generado, aunque con parámetros diferentes, especialmente en lo que se refiere al valor de pico –y, por tanto, a su energía asociada–, que disminuye cuanto más se aleja del punto de generación. En consecuencia, es posible que ciertos impulsos generados en líneas de alta tensión se propaguen por ellas, se transmitan a través del acoplamiento inductivo de los transformadores y aparezcan, atenuados, en las líneas de tensiones más bajas. [13]

1.4.5. Distorsión armónica

Se dice que existe distorsión armónica cuando la onda sinusoidal, prácticamente pura, que generan las centrales eléctricas sufre deformaciones en las redes de alimentación a los usuarios. Para cuantificar el grado de deformación de una onda de tensión o de intensidad que no es sinusoidal pura –aunque sí periódica, con 60 Hz de frecuencia–, se recurre a su análisis frecuencial. Este se lleva a cabo normalmente mediante la transformada rápida de Fourier, un algoritmo de cálculo que nos proporciona los contenidos de las diferentes ondas sinusoidales puras que componen la onda deformada. Estos contenidos hacen referencia a:

- La componente fundamental de la onda (60 Hz de frecuencia).
- Las componentes de frecuencias armónicas (múltiplos de 60 Hz), que reciben la denominación de armónicos de tensión o de intensidad. Su presencia debe limitarse.

Se puede considerar que, en su mayor parte, los equipos y elementos que componen los sistemas de distribución de energía eléctrica son lineales. En otras palabras, que su característica de intensidad/tensión se mantiene constante. No obstante, hay algunos equipos que tienen características no lineales, es decir, cuya intensidad demandada no es sinusoidal pura y sí, por tanto, una onda deformada. Estos equipos emiten armónicos a la red general de alimentación eléctrica en el punto de conexión común (PCC). [13]

1.4.6. Fallas de baja impedancia

Las fallas de baja impedancia, también conocidas como fallas de derivación o de corto circuito, ocurren cuando hay una conexión no deseada entre un conductor de fase y tierra o entre un conductor de fase y otro. El estándar americano IEEE y el estándar europeo IEC analizan y clasifican las fallas de derivación en cuatro categorías.

Como se evidencia en la figura 2.8, existen cuatro tipos de fallas de derivación: Fallas Trifásicas, monofásicas, bifásicas y bifásicas a tierra, de acuerdo al IEEE Std 3002.3TM-2018 y al IEC 60909 [14]. Las fallas bifásicas son más comunes que las fallas trifásicas y poseen corrientes de falla de aproximadamente un 87 % de la corriente de falla trifásica. Las fallas bifásicas a tierra son fallas monofásicas que se han intensificado para incluir un conductor de segunda fase.

En comparación, las magnitudes de las fallas bifásicas a tierra son generalmente mayores que las fallas bifásicas, sin embargo, son menores que las magnitudes de las fallas trifásicas [15].

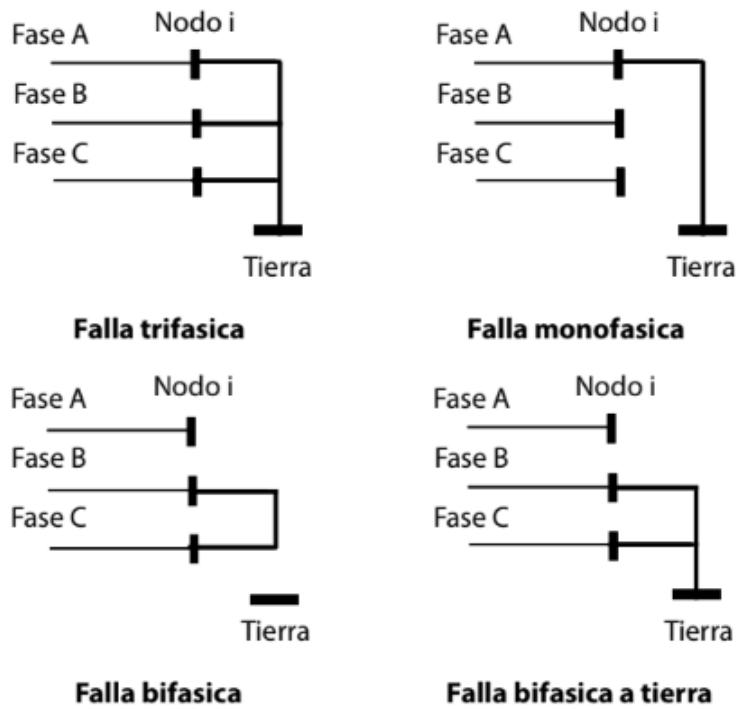


Figura 2.8: Tipos de fallas en derivación o de baja impedancia. Adaptada de: [14].

En la figura 2.9, se evidencian algunas de las fallas eléctricas que son detectables utilizando el medidor mostrado en la figura 2.2.

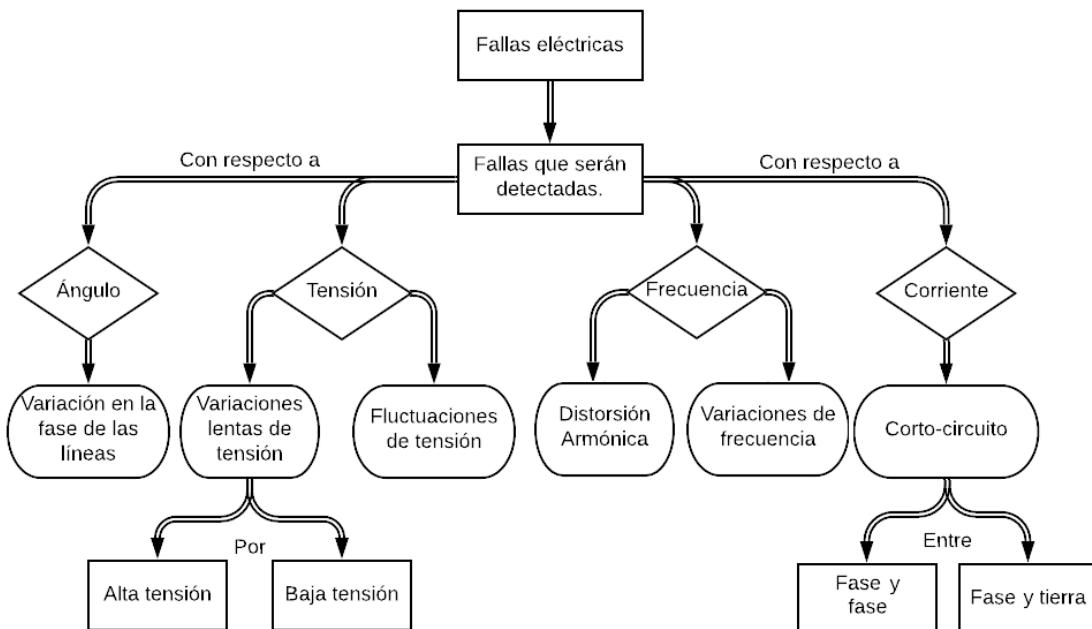


Figura 2.9: Diagrama de las fallas eléctricas detectables.

En la figura 2.10, se muestran algunas de las fallas eléctricas que no son detectables con el medidor, debido a carencia de fiabilidad en las medidas, ya que, en caso de ocurrir alguna de estas fallas, no es diferenciable para el sistema.

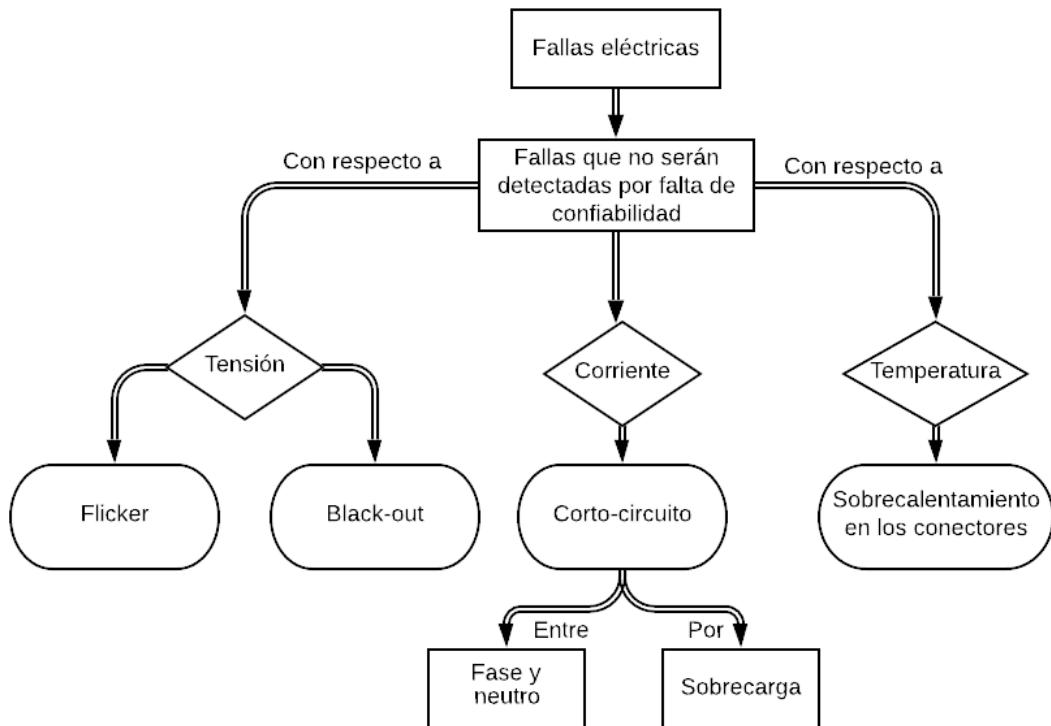


Figura 2.10: Diagrama de las fallas eléctricas no detectables.

2. Estado del arte

Los sistemas de medición y monitoreo, han sido empleados en múltiples ámbitos académicos, industriales e investigativos, principalmente en redes eléctricas; pues son estos sistemas integrados los que permiten brindar seguridad y protección tanto para los usuarios, como para los equipos relacionados. En este documento se presenta literatura con propuestas de aplicaciones con medidores inteligentes en sistemas de monitoreo, con enfoques industriales y sociales, algunas de ellas ya están siendo implementadas en diferentes países.

2.1. Medición inteligente y funcionalidades de medidores inteligentes en redes inteligentes: Una revisión

En el artículo [16] se realiza una revisión a cerca de los diferentes entornos en los que se puede involucrar la medición inteligente, se discute a cerca de varios elementos de la medición inteligente, el estado actual de las tecnologías relacionadas con la red inteligente, la infraestructura de medición avanzada (AMI) y el flujo de datos del medidor en la red.

Para realizar una medición inteligente, es necesario tener una infraestructura de medición avanzada (AMI), la cual es un sistema que permite tener enlaces de comunicación entre los usuarios y la red inteligente, a partir de los medidores inteligentes y sistemas de gestión de datos de los medidores (MDMS) respectivamente. El AMI proporciona una gestión inteligente, un mejor mantenimiento, adiciones y reemplazos más fáciles y adecuados de los activos de servicios públicos, lo que resulta en una mejor calidad de energía [16].

El sistema de gestión de datos del medidor (MDMS), es el corazón del AMI, este sistema se encarga de recopilar los datos que ingresan al centro de operaciones desde los concentradores, luego procesarlos y almacenarlos adecuadamente.

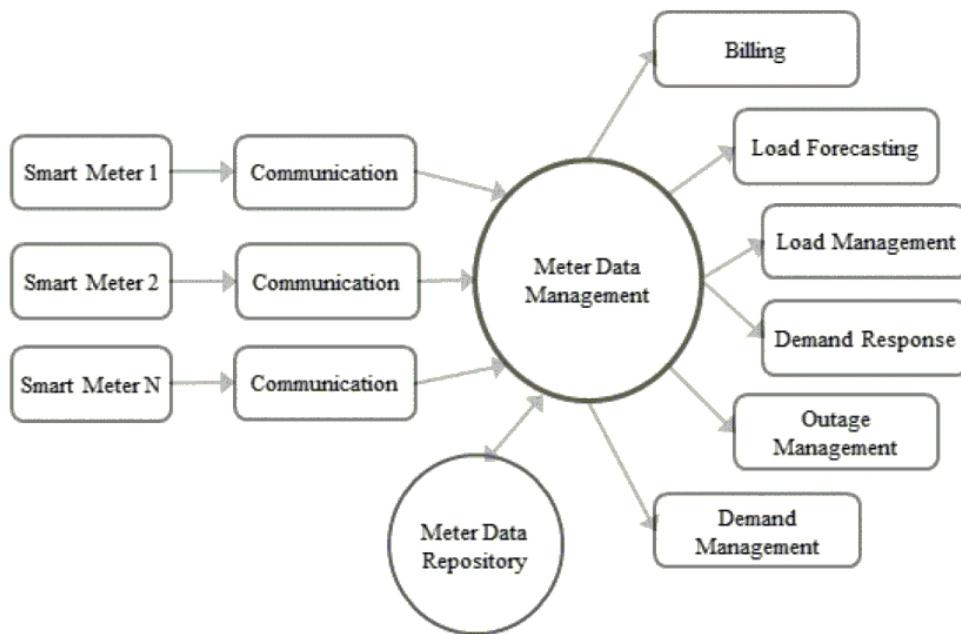


Figura 2.11: Sistema de gestión de datos del medidor (MDMS). Tomada de: [16].

En la figura 2.11 se muestra la arquitectura de conexión del MDMS en la que se presenta como las diferentes secciones del sistema de operación y administración interactúan con él y permiten que de esta manera se recopilen los datos de interés. Sin embargo, en este sistema centralizado, se pueden presentar problemas de no escalabilidad, como retrasos en la comunicación de los datos, pérdida de datos, escasez de procesamiento y desperdicio de una gran cantidad de recursos de comunicación [16].

2.2. Medición inteligente de energía y control de robo de energía con Arduino y GSM

En el artículo [17] presentan una manera de monitorear la potencia en el servicio eléctrico, con el fin de evitar cualquier tipo de robo energético, principalmente en países tan sobre-poblados como India, de manera que para automatizar y controlar el medidor, plantean conectar y desconectar el medidor cuando se detecten actividades no autorizadas, de

manera que se le informa al servidor central del usuario a través de un mensaje SMS, empleando tecnología GSM que se comunica directamente con un microcontrolador Atmega328P, el cual se encarga de desconectar el suministro no autorizado, por medio de relés de estado sólido para tratar las pérdidas no técnicas y la complicación de la fluctuación de voltaje.

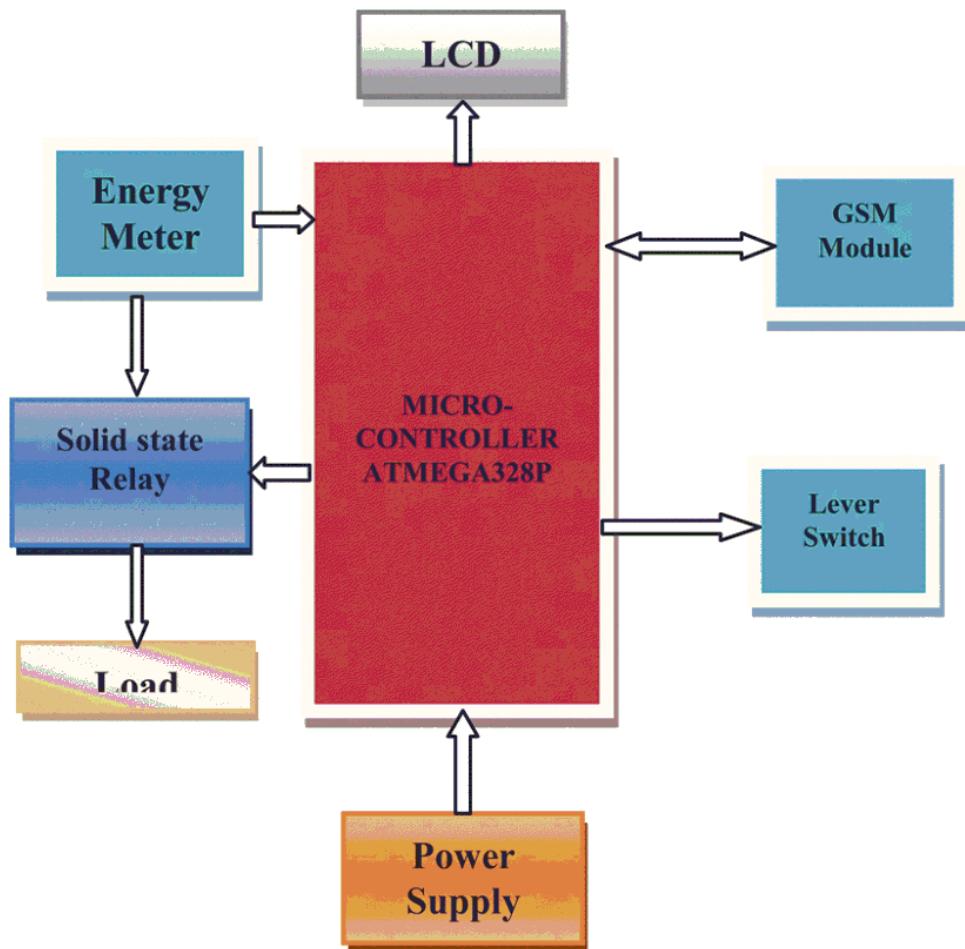


Figura 2.12: Diagrama de la arquitectura del sistema del medidor de energía inteligente basado en Arduino y GSM. Tomada de: [17].

En la figura 2.12 se muestra la arquitectura empleada para la medición, en donde se presentan los bloques que componen el sistema, algunos de ellos más complejos que otros, como por ejemplo el bloque del medidor de energía, el cual se compone por un sensor de corriente, un sensor de voltaje, y el Arduino, el cual tiene internamente guardado el programa que calcula la cantidad de kilovatios hora consumidos, a partir de los datos obtenidos por los sensores. Otro bloque importante, es el encargado de la polarización del sistema, la cual tiene su propia etapa de rectificación y regulación, de manera que el microcontrolador esté siempre energizado, en caso de que se detecte alguna anomalía en el sistema eléctrico del usuario, a cualquier hora del día, los bloques de comunicación, GSM, o los de los relés de estado sólido, son de enlace directo con el controlador [17].

2.3. Medidor de energía inteligente basado en IoT para una utilización eficiente de la energía en la red inteligente

En el artículo [18] se presenta un sistema de medición inteligente empleando Internet de las cosas (IoT), este medidor propuesto controla y calcula el consumo de energía utilizando ESP8266-12E, un módulo Wi-Fi y lo carga en la nube desde donde el consumidor o productor puede ver la lectura. Por lo tanto, el análisis de energía por parte del consumidor se vuelve mucho más fácil y controlable, sin mencionar, que este sistema también ayuda a detectar el robo de energía.

En este sistema propuesto, se conecta un medidor de energía al ESP8266 12E mediante un optoacoplador. Una pantalla OLED también está conectada al sistema. El circuito del controlador ULN2003 se usa para conducir el relé para cambiar las cargas. Un sensor de corriente también está equipado para determinar el robo de energía [18].

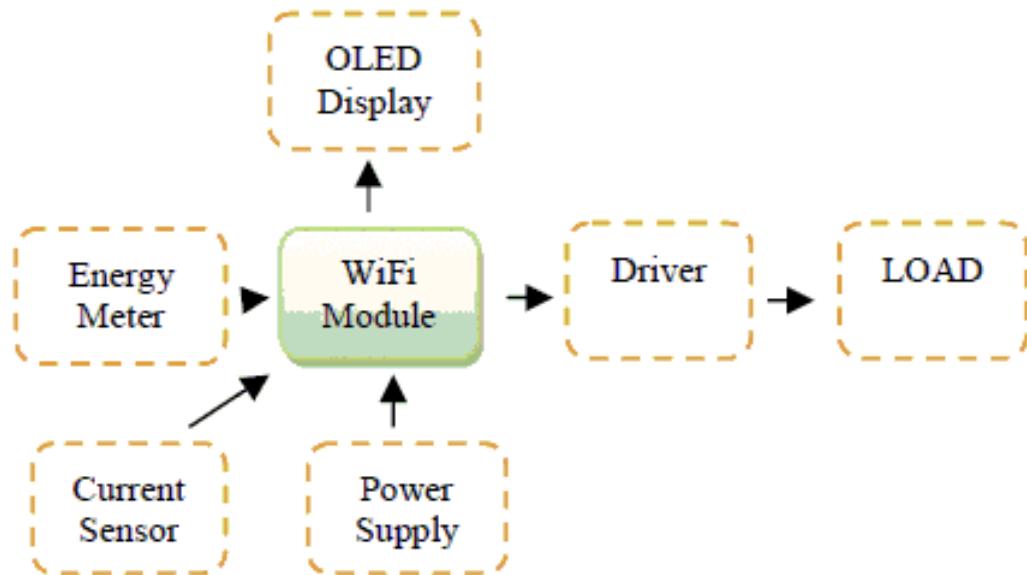


Figura 2.13: Diagrama de bloques funcionales del medidor inteligente. Tomada de: [18].

En la figura 2.13 se presenta el diagrama de bloques del medidor propuesto, en el cual se muestran los bloques independientes que se emplean en el sistema, el medidor de energía se comunica con el módulo Wi-Fi a través de ESP8266-12E, la fuente de alimentación energiza el medidor con 230 V de corriente alterna y el módulo Wi-Fi con 5 V en corriente directa, el driver es a través de un controlador de relé como es mencionado anteriormente y el sensor de corriente es un sensor ACS712 con capacidad de hasta 5 A [18].

2.4. Un sistema de configuración para monitoreo y control en tiempo real del comportamiento del consumo de electricidad

En el artículo [19] el gobierno Indonesia presenta una propuesta para reducir el consumo de energía en un 1 % hasta 2025, mediante realimentación en tiempo real a través de

los teléfonos inteligentes de los usuarios. Los usuarios recibirán una notificación e información sobre el estado de los equipos eléctricos / electrónicos que se han desperdiciado y los alentará a tomar las medidas necesarias para administrarlos.

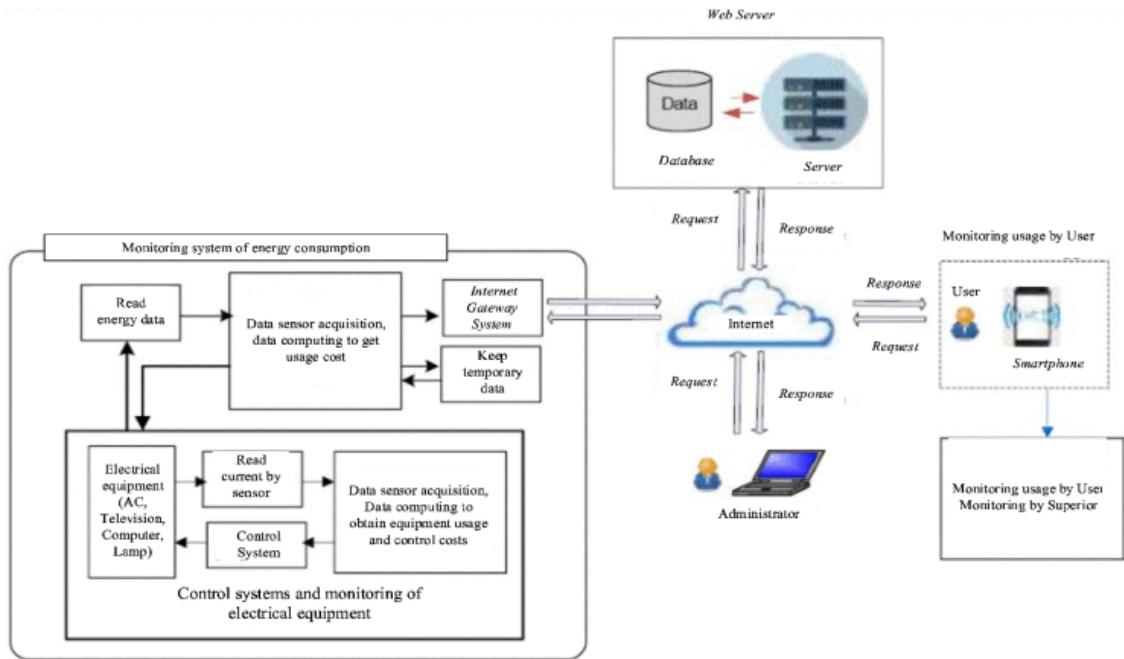


Figura 2.14: Proyecto de sistema de configuración del esquema. Tomada de: [19].

En la figura 2.14 se muestra el esquema propuesto para el proyecto, a partir de esta arquitectura el sistema puede controlar el uso de energía eléctrica en un edificio en función de la cantidad de electricidad utilizada por los equipos electrónicos que utilizan un sensor de corriente montado en los dispositivos. Los datos leídos por el sensor se enviarán al servidor web a través de Internet y se guardarán en la base de datos. Wi-Fi en la habitación junto con el teléfono inteligente del usuario formará un sistema que puede monitorear el movimiento del usuario entrando o saliendo de la habitación y el estado de encendido / apagado del equipo. Si el sistema identifica que el usuario abandona la sala sin apagar los dispositivos de la sala, se enviará una notificación al teléfono inteligente del usuario y se le indicará que regrese a la sala y tome medidas para ahorrar energía [19].

2.5. Estudio sobre una red inteligente y el sistema de tecnología clave para impulsar el desarrollo de la generación distribuida de energía.

En el artículo [20] se presenta un sistema de generación distribuida de energía, el cual comprende una componente de generación de energía renovable y una componente de generación de energía tradicional, para el cual se pretende realizar una integración de tecnología de detección avanzada, sistema de comunicaciones y tecnología de control automático, en donde el sistema supervisa la red en tiempo real y coordina distintos procesos.

Este sistema está presentado para una red eléctrica a gran escala, se diseña principalmente para impulsar el monitoreo constante de la calidad y el control de los parámetros que permitirán el buen funcionamiento de una fuente de energías renovables, de esta manera un sistema de paneles fotovoltaicos serán activados por el sistema de monitoreo solo al detectar que existe presencia de luz solar, de igual manera, un sistema de energía eólica será activado solamente cuando se presente cierto nivel en los vientos que cruzan por la planta.

El autor considera una tecnología importante la implementación de módulos inteligentes para mejorar la eficiencia energética del sistema, distribuir correctamente la energía a áreas remotas lejos de los centros de distribución y por supuesto, lograr ahorro de energía y reducir las emisiones contaminantes al medio ambiente.

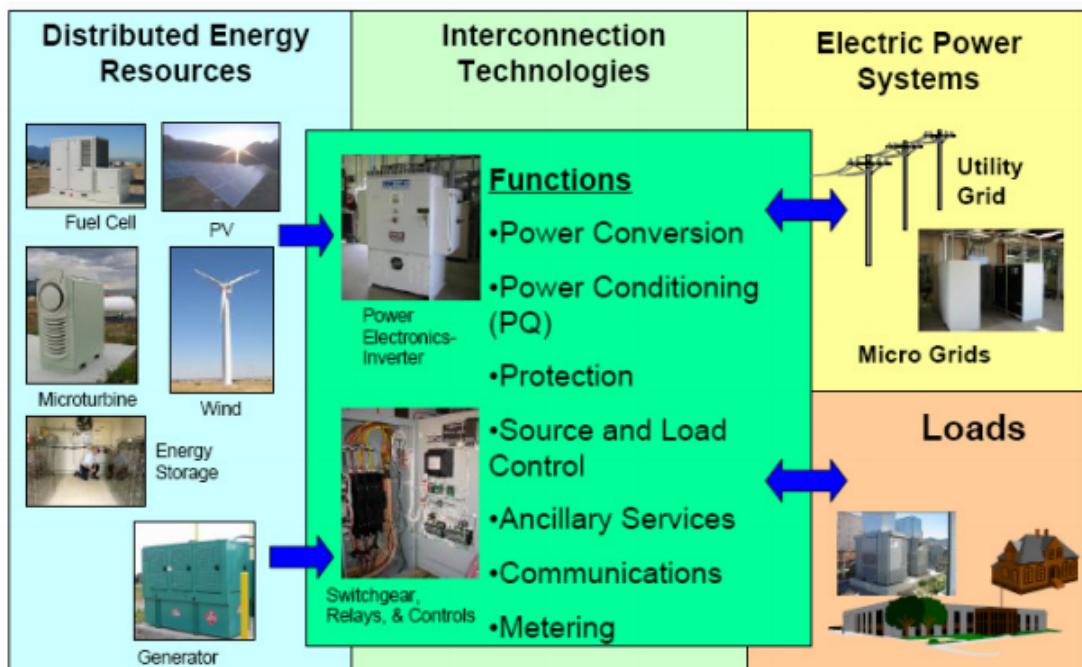


Figura 2.15: Esquema de la red inteligente implementada. Tomada de: [20].

En la figura 2.15 se muestra un esquema gráfico que describe las componentes que estructuran dicha red inteligente explicada anteriormente.

2.6. Modelos de software para una red inteligente

En el artículo [21], se propone como etapa principal antes de pensar en cualquier integración de una red inteligente, la arquitectura de un software necesario para cuantificar y evaluar un proyecto, además de la parte que nos interesa para nuestro proyecto, que es el de monitorear el progreso y planificar la realización de una red inteligente.

Dado que la red eléctrica al querer ser soportada mediante módulos inteligentes, esta deberá utilizar sistemas digitales de control y comunicaciones para monitorear y controlar los flujos de energía, con el objetivo de que la red sea más resistente, eficiente y rentable.

A medida que la tecnología de las redes inteligentes avanza con el tiempo, la complejidad en la gestión de la red eléctrica también va aumentando. Por esto, se hace necesario medir la eficiencia, fiabilidad y calidad de los servicios que proporciona o consume la red inteligente para calcular su progreso. Así, un modelo de software es requerido para mostrar el estado actual de la red inteligente y de esta manera establecer planes a futuro.

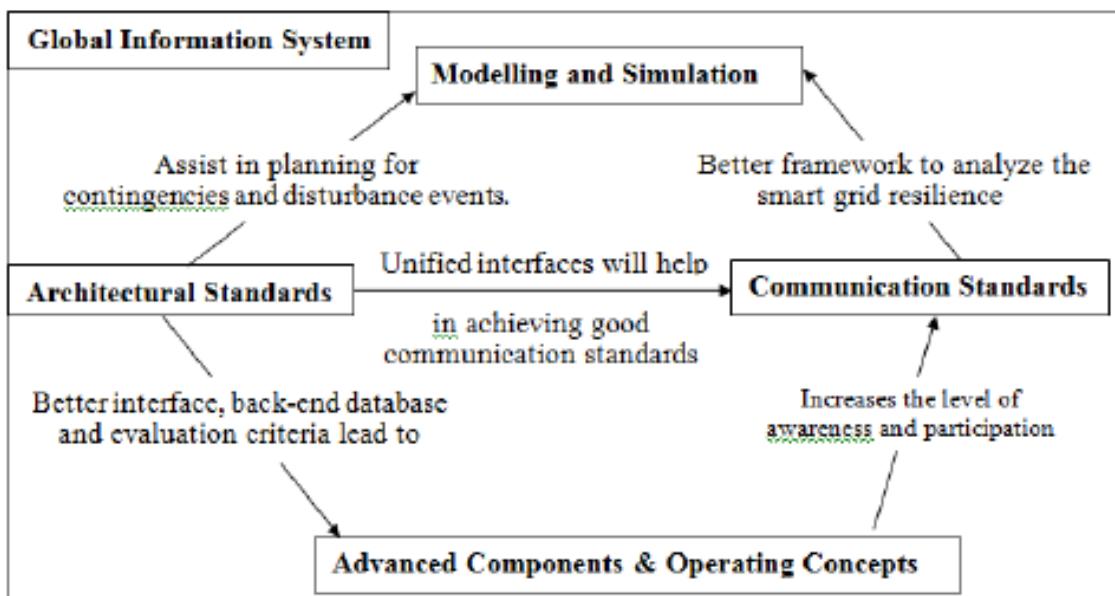


Figura 2.16: Interconexión entre varios componentes. Tomada de: [21].

En la figura 2.16 se muestra la propuesta del enlace entre hardware y software desarrollada por los autores.

2.7. Aplicación de la comunicación por línea eléctrica en consumo de energía inteligente

En el documento [22], se expone a cerca de la importancia de una red inteligente como medida importante para responder al problema del calentamiento global y la escasez de energía. El autor enfatiza en que las tecnologías de comunicaciones son la herramienta clave para lograr su obtener información de la red e interconectarla a los actores dependientes de esta, su automatización e interactividad. En este documento, se realiza un experimento de prueba de rendimiento de comunicación de una línea de banda ancha y la aplicación de un PLC para un proyecto de tipo prueba, con el fin de estudiar el consumo de energía por medio de módulos inteligentes, con el fin de demostrar una perspectiva favorable de la aplicación de PLC en el campo de consumo de energía de una red inteligente. Y finalmente, también se presenta una predicción de la tendencia de desarrollo a futuro.

En la figura 2.17 se muestra la propuesta de modelamiento de la red inteligente, en la que se muestra la interconexión entre el sistema de comunicaciones, la red eléctrica y los sistemas de control y medición.

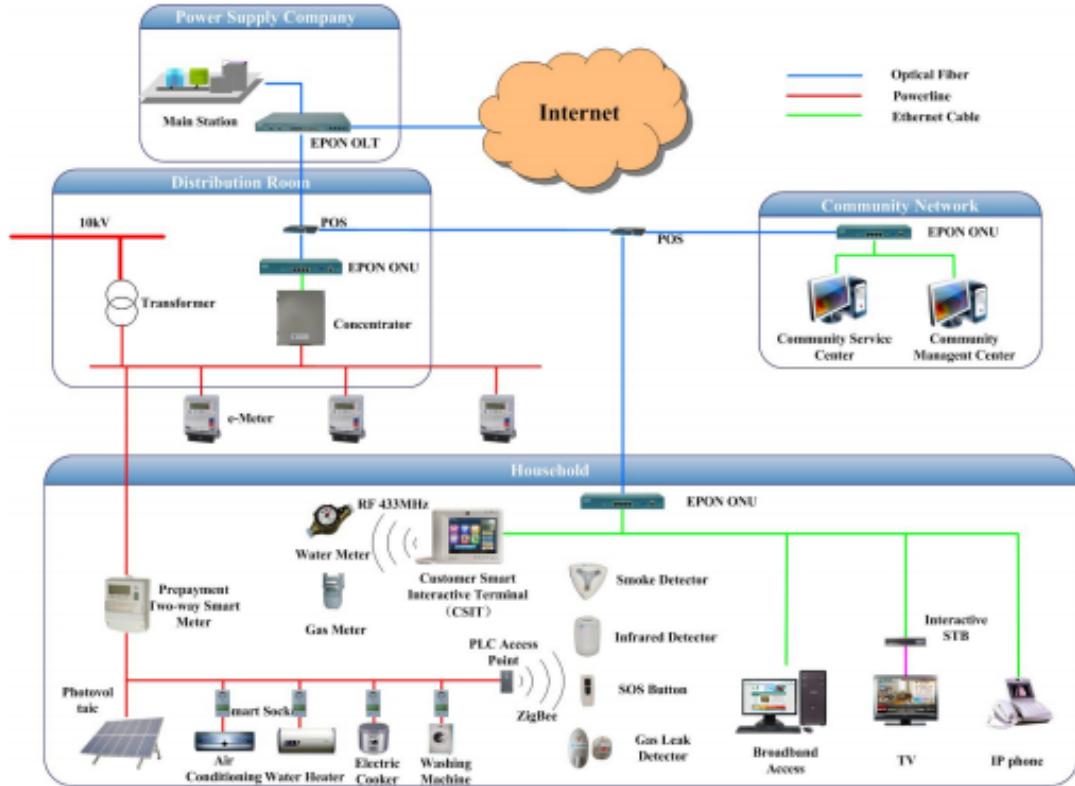


Figura 2.17: Diagrama de la red inteligente empleada en la aplicación. Tomada de: [22].

2.8. Smart Grid: una evaluación de oportunidades y desafíos en el despliegue del sistema de energía de Ghana

El artículo [23] se expone un estudio de los beneficios y acciones que conlleva la implementación de una red inteligente en el país de Ghana para el manejo de generación y distribución de energía. Nos muestra de igual manera un estudio general de las oportunidades y beneficios que traería al país si se llega a implementar una red inteligente.

Básicamente, el documento parte del potencial que tiene Ghana para generar riqueza por medio de una industrialización masiva y la creación de empleos, para lo cual el fuerte fundamental de este sector abarcará el mercado energético, es decir, la generación y distribución de energía en un sector local y a futuro pensar en la exportación de energía eléctrica. Para 2015, se planeó un estudio que permitiría aumentar la capacidad de generación de energía de 2 GW a 5 GW, con el fin de pensar aumentar el acceso a la electricidad que era del 66 % a un acceso universal.

Lo anterior, tuvo como causa directa la generación de políticas que tuvieron como objetivo la búsqueda de un sistema que garantizara que esta generación de energía se realizara de manera eficiente, tuviese una mayor confiabilidad de la que se tenía antes y que los suministros de energía se revisaran de forma constante. Por lo cual, el país optó por una actualización de su red eléctrica obsoleta, ordenando así el estudio para la implementación de una red inteligente (Smart grid) [23].

Capítulo 3

Detección de fallas

La detección de fallas eléctricas es un punto clave en el proyecto, pues permite conocer el lugar y el momento exacto en el cual se presentó dicha falla; conociendo esta información, se pueden tomar acciones de desconexión o corrección del error con mucha más rapidez. Tal y como se evidencia en las figuras 2.8, 2.9 y 2.10, con los datos de los parámetros capturados por el medidor, se puede detectar cierto tipo de falla, que se clasifica según la variable eléctrica relacionada. Dependiendo de su respectiva variable y el tipo de falla medible que pueda ocurrir, se detecta de una manera diferente. En este capítulo se muestra la manera en la que se clasifican y se detectan las fallas medibles, a partir del parámetro eléctrico capturado.

1. Variaciones lentas y fluctuaciones de tensión

Debido a que las fluctuaciones de tensión equivalen a un 10 % por encima o por debajo del valor nominal, no tienen grandes repercusiones cuando se está trabajando con dispositivos con esta tolerancia. Sin embargo, es una falla que genera errores mayores en el sistema, cuando se mantiene por más de un minuto, generando así variaciones lentas de tensión.

Por otra parte, cuando la tensión decrementa por debajo del 90 % o aumenta por encima del 110 %, se generan reducciones (Sag) o elevaciones (Swell) de tensión respectivamente. Este tipo de fenómenos también dependen del tiempo de duración, ya que se pueden clasificar entre instantáneos, momentáneos o temporales; de la misma forma que con las fluctuaciones de tensión, si las reducciones o elevaciones de tensión se mantienen por más de un minuto, se conocen como subtensiones y sobretensiones respectivamente.

Cabe resaltar que cuando la tensión cae por debajo del 10 % del valor nominal, se conoce este fenómeno como caída de tensión, manteniendo valores bastante pequeños pero superiores a cero. En un caso mucho más grave, cuando la tensión cae completamente a un nivel de cero, esta perturbación se conoce como interrupción y puede generar errores en el sistema con respecto a la corriente que fluye a través de los equipos. Si se mantiene la interrupción de tensión por más de un minuto, la perturbación se conoce como colapso y puede afectar seriamente todos los equipos del sistema, por lo que se recomienda desconectar completamente de la red de energía. En la figura 3.1 se muestra el diagrama de flujo utilizado para detectar y clasificar todas estas perturbaciones de tensión.

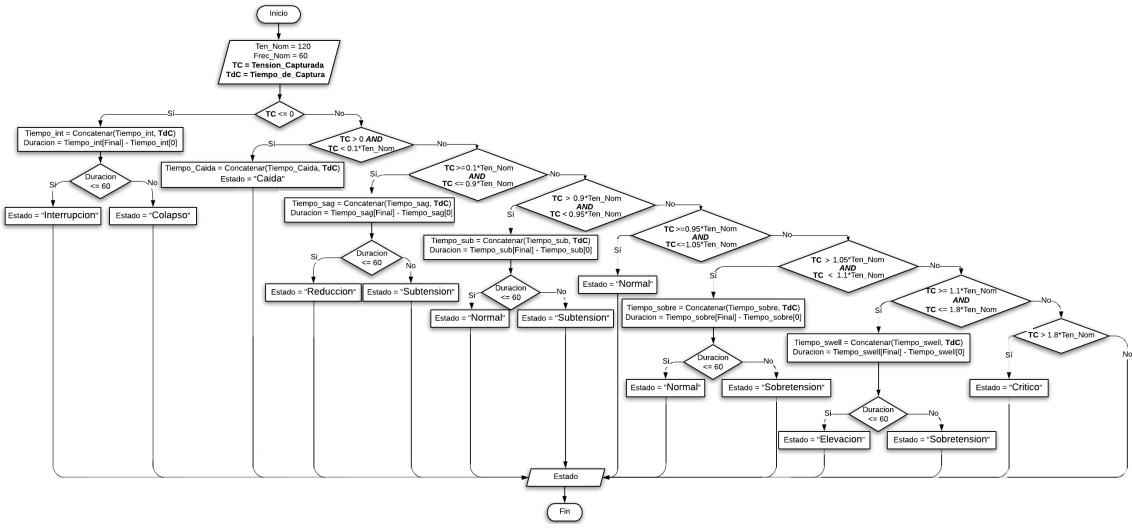


Figura 3.1: Diagrama de flujo del algoritmo para la detección de las fallas relacionadas a la tensión monofásica.

En la figura 3.1 se presenta el diagrama de flujo utilizado para detectar y clasificar las variaciones en tensión. Este proceso se realiza dentro de una función, por lo que se repite para cada una de las líneas medidas. El algoritmo parte del valor de tensión capturado y el tiempo de captura de dicho valor, además de las constantes nominales de tensión y frecuencia como valores de referencia. Básicamente, se emplean varios condicionales que dependen del valor capturado; según el rango en el que se encuentre el valor capturado se ejecutan unas acciones específicas. Estas acciones detectan el tiempo de duración de la perturbación y le asignan un estado a dicha perturbación, en este caso, estos estados pueden ser: “Interrupción”, “Colapso”, “Caída”, “Reducción”, “Subtensión”, “Normal”, “Sobretensión”, “Elevación” o “Crítico”. También se muestra en pantalla la duración de la perturbación y su ubicación, una vez se estabiliza la tensión y se detecta un estado “Normal”.

2. Distorsión armónica y variaciones de frecuencia

La distorsión armónica tanto en tensión como en corriente, es un factor que tiene bastante importancia cuando se quiere que los equipos y dispositivos trabajen correctamente. Esto se debe a que los equipos que suelen trabajar con niveles de distorsión elevados, son más propensos a sufrir daños y fallos no deseados, como por ejemplo, sobrecarga de los conectores, sobrecarga de baterías de condensadores, disparo de las protecciones, aumento de pérdidas sobre motores de inducción, entre otros. Sin mencionar que es importante minimizar estos niveles de distorsión, porque sus efectos pueden repercutir en instalaciones próximas de otros usuarios. Por tal motivo, es de suma importancia conocer el nivel de distorsión armónica que poseen las señales de tensión y de corriente con las que se está trabajando; de esta manera, se pueden tomar acciones para prevenir cualquier tipo de daño o accidente.

En Colombia se definen los niveles de distorsión operables, para la calidad de la energía y la operación de los equipos, con la norma IEC/EN 61000-2-2. En la cual se establece que los valores de THD en tensión por encima del 5 % y en corriente por encima del 30 % poseen riesgo de calentamiento y de sufrir fallos de funcionamiento. Por lo tanto, en la figura 3.2 se muestra el diagrama de flujo utilizado para detectar el estado de los valores capturados para THD en tensión y corriente, según los límites establecidos por la norma.

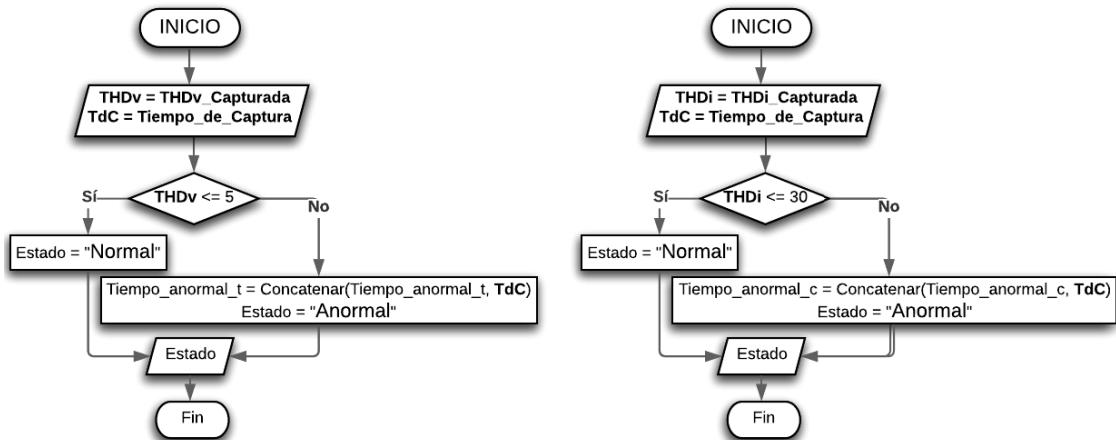


Figura 3.2: Diagrama de flujo del algoritmo para la detección del estado de los valores de THD en tensión y corriente.

Como se puede observar en la figura 3.2, los algoritmos para la detección de estos estados son bastante simples, pues si el valor capturado se encuentra por debajo del umbral, se le asigna un estado “Normal” a dicho valor. Por otra parte, si el valor es mayor al umbral en cada caso, se le asigna un estado “Anormal”, que implica directamente que se deben tomar acciones para corregir estos niveles de distorsión, de esta manera prevenir algún tipo daño en los equipos. En la figura 3.2, el diagrama de la izquierda corresponde al realizado para clasificar los valores THD en tensión, mientras que el diagrama de la parte derecha, corresponde al realizado para clasificar los valores de THD en corriente. También se muestra en pantalla la duración del parámetro fuera de rango y su respectiva ubicación, una vez se estabilizan los niveles de THD y se detecta de nuevo un estado “Normal”.

Además de la distorsión armónica, uno de los parámetros más delicados en términos de operación, funcionamiento e interconexión de dispositivos, es la frecuencia, dado que si ocurren alteraciones en este parámetro, se corren riesgos al momento de operar elementos y dispositivos eléctricos y electrónicos. La norma establece que la variación límite para este parámetro es de 4 % por encima y por debajo del valor normal de frecuencia. En Colombia, se opera con una frecuencia de red eléctrica igual a 60 Hz, para lo cual, se crea un límite de frecuencia que define un estado normal en la red, comprendido entre 57,6 Hz y 62,4 Hz para el valor de frecuencia medido. Los valores dentro de este intervalo se pueden operar y tratar como valores en estado normal. Mientras que, los valores fuera del intervalo definido anteriormente, se clasifican como valores con variación de frecuencia. Estos valores implican una falla en la red en términos de frecuencia. En la figura 3.3 se muestra el diagrama de flujo utilizado para detectar y clasificar estos valores de frecuencia.

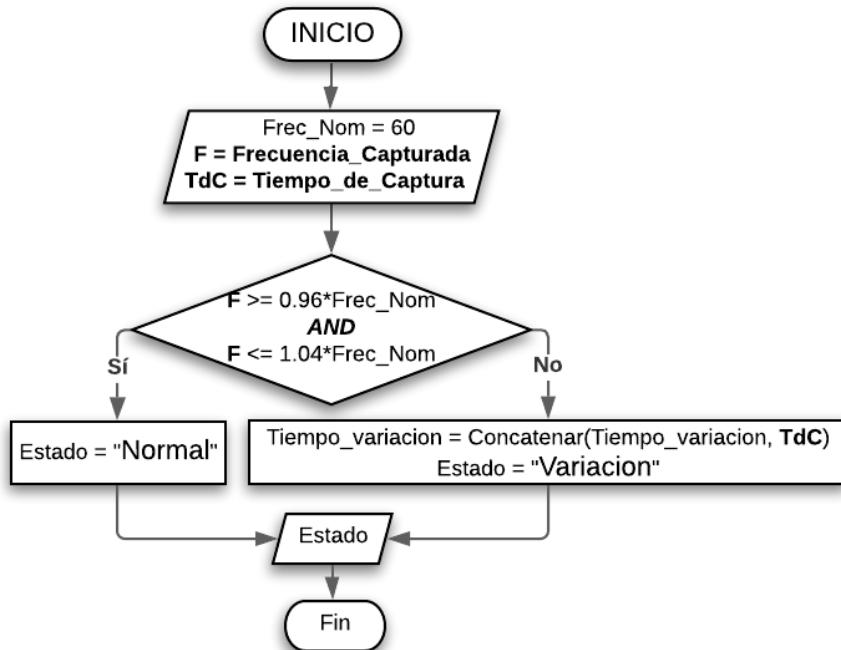


Figura 3.3: Diagrama de flujo del algoritmo para la detección de las variaciones de frecuencia.

Como se puede observar en la figura 3.3, se define una valor de frecuencia nominal y se genera un condicional para el rango de tolerancia. Si los valores capturados se encuentran dentro de este rango, se le asigna un estado “Normal” a dicho valor. Si por el contrario, el valor capturado se encuentra fuera del rango, se detecta la duración de la variación y se le asigna el estado “Variacion” al valor medido. También se muestra en pantalla la duración de la variación y su ubicación, una vez se estabiliza la frecuencia y se detecta de nuevo un estado “Normal”.

3. Fluctuaciones de corriente

Las fluctuaciones de corriente son fenómenos que pueden ocurrir por fallos físicos internos de la red eléctrica o por alto consumo de corriente. Estas fluctuaciones presentan valores de corriente por encima de los estándares de valores nominales para la red eléctrica, por lo cual, resulta indispensable, la detección de los valores que superen los límites establecidos. Cabe mencionar que este tipo de falla es completamente distinto a las fallas de baja impedancia o cortocircuito que serán tratadas en la siguiente sección. En la figura 3.4 muestra el diagrama de flujo empleado para la detección y clasificación de estas fluctuaciones.

Como se puede evidenciar en la figura 3.4, el diagrama de flujo para la detección de las fluctuaciones de corriente, es un algoritmo bastante sencillo, similar a los mostrados anteriormente. En este caso, se establecen dos valores umbrales iguales a 10 A y 20 A respectivamente.

Si un valor es de corriente capturado es inferior a 10 A, se le asigna un estado “Normal” a dicho valor; si en su lugar el valor capturado es mayor a 10 A pero inferior a 20 A, se considera un valor de alerta, por lo que el estado de dicho valor se establece como “Alerta”. Si en caso contrario, este valor es superior al segundo umbral (20 A) inmediatamente se le asigna un estado “Crítico”, indicando que deben realizarse acciones para mejorar el funcionamiento. Para todos los casos donde el estado es diferente al estado normal, se toma el tiempo de duración hasta que haya un cambio de estado; este valor se muestra en pantalla junto con la ubicación de esta anomalía. Cabe resaltar que dependiendo la aplicación y el rango de operación de los equipos y dispositivos estos valores umbrales pueden modificarse para ampliar o disminuir los diferentes rangos.

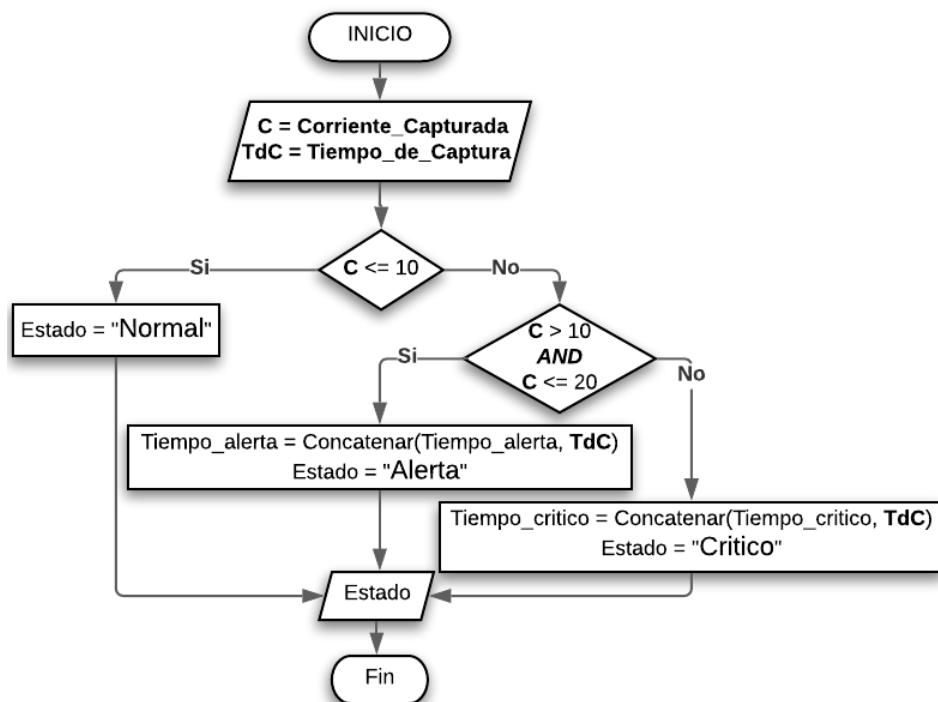


Figura 3.4: Diagrama de flujo del algoritmo para la detección de las fluctuaciones de corriente.

4. Fallas de baja impedancia o cortocircuito

Las fallas de baja impedancia o fallas en derivación, son las fallas más críticas y peligrosas cuando se trabaja con equipos cuya fuente de alimentación es trifásica. Ya que estas fallas presentan diferentes consecuencias operativas, de infraestructura, financieras y hasta humanas. Como se mencionó en el capítulo anterior, estas fallas ocurren cuando hay alguna conexión no deseada entre un conductor de fase y tierra o entre conductores de fase. Además, en la figura 2.8 se muestra que estas fallas tienen una clasificación en cuatro categorías: Falla trifásica, falla monofásica, falla bifásica y falla bifásica a tierra. En la figura 3.5 se muestra el diagrama de flujo empleado para la detección y clasificación de estas fallas.

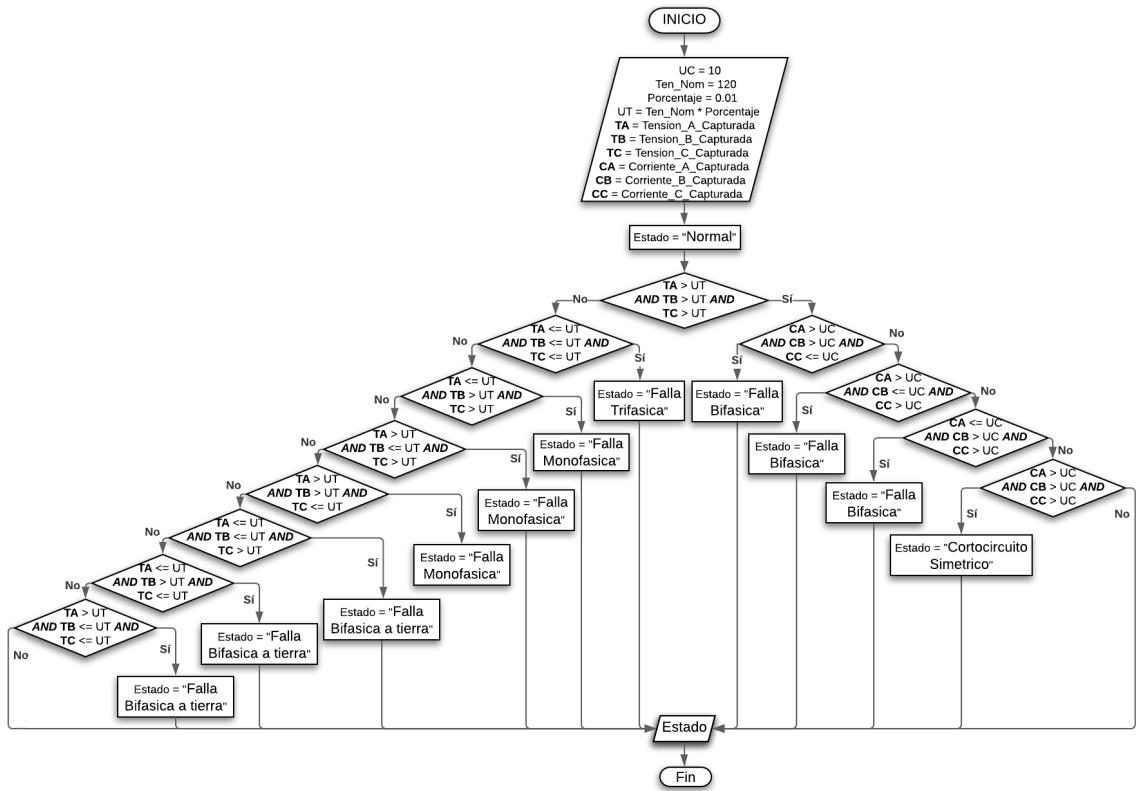


Figura 3.5: Diagrama de flujo del algoritmo para la detección y clasificación de fallas de baja impedancia o cortocircuito.

En la figura 3.5 se puede evidenciar que el diagrama de flujo empleado para detectar y clasifica las fallas de baja impedancia, posee varias entradas, esto se debe a que para realizar esta clasificación es necesario conocer los valores de tensión y corriente en cada una de las líneas. Al inicio del diagrama también se puede observar que hay unas entradas constantes, esto se realiza para ajustar los umbrales de los condicionales; en este caso, se establece un umbral de tensión del 1 % del valor nominal (120 V) y un umbral de corriente de 10 A. Para cada línea y variable se ingresan los tiempos de captura respectivos, para un análisis adicional de tiempos. Después de todas las entradas, se establece el estado trifásico por defecto, el cual para este caso, se establece en “Normal”.

El algoritmo se basa en bastantes condicionales anidados, en los cuales se decide que estado asignar, dependiendo los valores de tensiones y corrientes que se hayan capturado. Inicialmente se discrimina si todas las líneas en tensión son superiores al valor umbral, indicando que los valores en tensión son normales, por decirlo de cierta manera. En caso de ser así, se realizan condicionales con los valores de corriente en cada línea, si dos valores son superiores al umbral en corriente, se clasifica esta falla como una falla bifásica y se establece el estado trifásico como “Falla Bifásica”; en su lugar, si los tres valores superan el umbral en corriente, el estado se establece como “Cortocircuito Simétrico” (clase adicional). Si por el contrario, los valores no concuerdan con ninguna de estas condiciones, el estado inicial permanece igual.

Por otra parte, si desde el inicio, en el primer condicional, donde se discrimina si todos los valores en tensión superaron su respectivo umbral, resulta tener respuesta negativa, esto indica que por lo menos una de las líneas presenta falla. En ese caso se realizan más condicionales para determinar cual o cuales líneas son las que poseen la falla en derivación. Si únicamente una de las líneas es inferior al umbral de tensión, se clasifica esta falla como una falla monofásica y se establece el estado como “Falla monofásica”; si en su lugar son dos líneas las que presentan valores inferiores al umbral, se detecta una falla bifásica a tierra y se clasifica de esta manera, estableciendo el estado trifásico como “Falla bifásica a tierra”. En caso contrario, donde las tres líneas sean inferiores al umbral de tensión, se clasifica esta falla como una falla trifásica y se establece el estado trifásico como “Falla Trifásica”. En todos los casos cuando ocurre una falla se empieza a contabilizar su duración. Cuando se cambia de estado, se imprime en pantalla la duración, el tipo de falla y su respectiva ubicación.

Capítulo 4

Medidor inteligente

Como es mencionado en capítulos anteriores, el medidor inteligente es el pilar fundamental del proyecto, ya que este dispositivo, es el encargado de extraer y enviar al servidor los parámetros de la red eléctrica para ser manipulados y visualizados; por lo tanto, resulta de gran importancia describir su proceso de adaptación, configuración e instalación, desarrollado a lo largo del proyecto. En este capítulo, se pretender realizar una explicación de manera concisa a cerca del procedimiento implementado con respecto al medidor inteligente (acceso, configuraciones y código desarrollado).

1. Acceso al medidor inteligente

Como primera medida, se realiza la descripción del paso a paso del proceso de conexión, configuración y adaptación del medidor inteligente, al momento de ser conectado por primera vez en una red eléctrica; además, se describe el procedimiento utilizado para que el medidor inteligente pase de un punto inicial de fábrica, con especificaciones técnicas predeterminadas, a valores de acceso propios relevantes en el proyecto. Con la ventaja de que los pasos aquí descritos, pueden ser utilizados para configurar el medidor desde su estado inicial, en caso de ser necesarios.

1.1. Conexiones físicas

Es fundamental que el medidor se encuentre conectado a la red eléctrica, dado que, este dispositivo extrae la energía para su funcionamiento de la red que se está midiendo. Tal y como se mostró en el marco teórico, el módulo cuenta con 2 CHIPS para la medición de corriente, los cuales serán energizados a partir de una entrada de tensión de tres líneas con neutro.

Una vez conectado el medidor inteligente, este encenderá un led verde, el cual corroborará que el medidor ha sido conectado de manera correcta. La figura 4.1 muestra el diagrama de conexión del medidor a una red eléctrica describiendo sus componentes físicas y electrónicas.

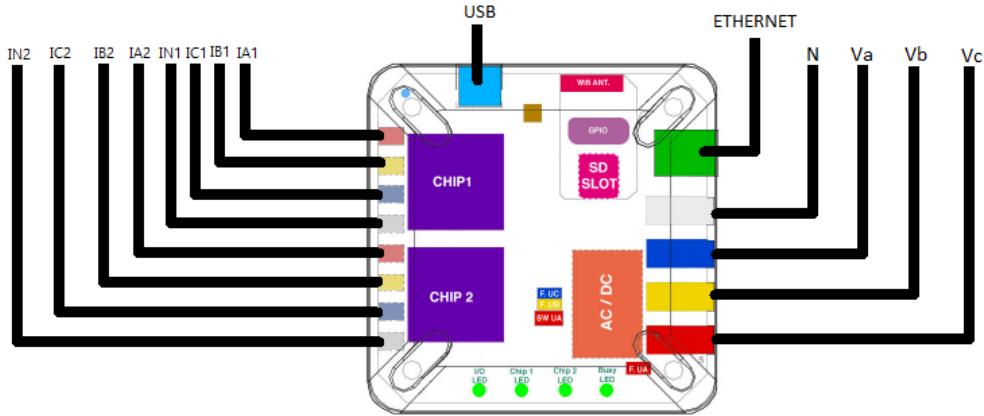


Figura 4.1: Diagrama de conexión del medidor inteligente a la red eléctrica. Adaptada de: [6].

1.2. Acceso al sistema operativo del medidor - Acceso a la red WiFi

El medidor cuenta con tres vías de acceso para ser controlado desde la consola de Linux: SSH, HTTP y WiFi/Ethernet. El primer paso que se requiere para acceder al medidor, es conectarlo a la red WiFi local, este proceso puede ser realizado fácilmente; con el fin de que el medidor pueda ser accedido mediante otros ordenadores. El proceso es sencillo, iniciando con la conexión a la red WiFi del medidor. Cada medidor funciona como un repetidor, siendo capaz de generar una red inalámbrica propia, con el fin de que el usuario administrador acceda a ella, para que a través de ésta, cree una sesión por consola para gestionar el medidor. Para lo cual, el primer paso es acceder a esta red, la figura 4.2, muestra la red del medidor B6FF. Es importante aclarar que las contraseña configurada para la conexión WiFi de los 3 medidores es 'eGEOtech900'.

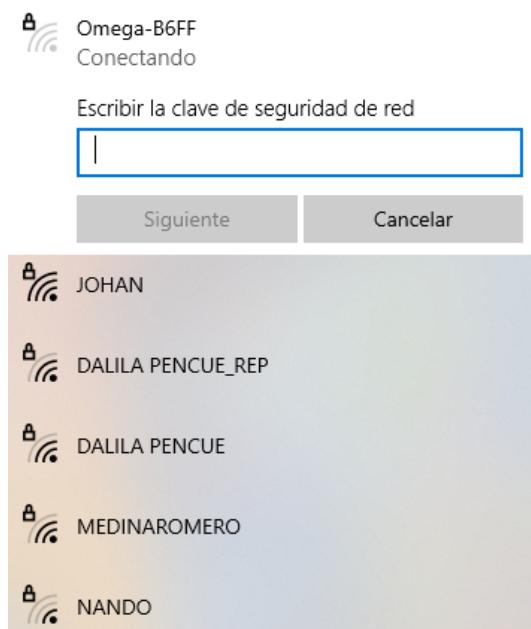


Figura 4.2: Acceso a la red WiFi del medidor B6FF.

Una vez accedido a la red WiFi del medidor, podremos ingresar mediante SSH o HTTP. La metodología de conexión mediante estos protocolos, se muestran a continuación.

1.2.1. Conexión HTTP

HTTP es un protocolo de red, que presta el servicio de brindar una conexión punto a punto (cliente - servidor), en donde para este caso en particular los administradores serán los clientes y el servidor será el medidor inteligente. La URL de acceso puede ser utilizada mediante un navegador web, con la siguiente estructura: <http://omega-XXXX.local/>, en donde el parámetro ‘XXXX’ representa el ID del medidor, el cual en este caso será el medidor B6FF. Consumiendo la URL <http://omega-B6FF.local/> en el navegador, se establece la conexión y se ingresa a la interfaz de logeo. La figura 4.3 muestra dicha interfaz.

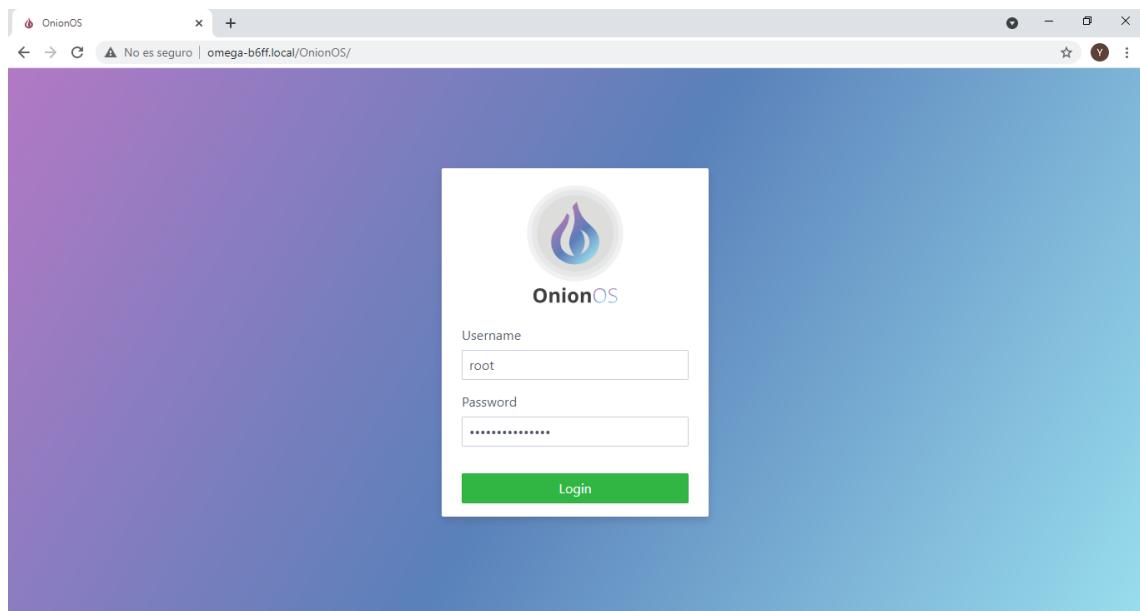


Figura 4.3: Interfaz de inicio de acceso al medidor inteligente B6FF.

Esta interfaz, permitirá el ingreso a la consola del medidor inteligente, así como a documentación y una interfaz de configuración propia del sistema operativo OnionOS. Para lo cual se debe ser muy cuidadoso a la hora de manipular o realizar acciones en dicha interfaz. Las credenciales establecidas para el acceso a dicha interfaz son:

- Usuario: root
- Password: onioneer (Por defecto)

Una vez ingresadas dichas credenciales, se debe establecer la red WiFi a utilizar para acceder de manera remota al medidor sin tener que utilizar su red WiFi, si no que desde cualquier dispositivo conectado a la misma red WiFi que el medidor, se pueda establecer una sesión HTTP o SSH mediante internet. La interfaz de configuración de la red WiFi que

conectará el medidor a la red LAN que se prefiera y así conectar el dispositivo a internet se presenta en la figura 4.4. Anterior a esta configuración es posible que el medidor le vuelva a pedir las credenciales de acceso para el usuario root y presente un mensaje de bienvenida.

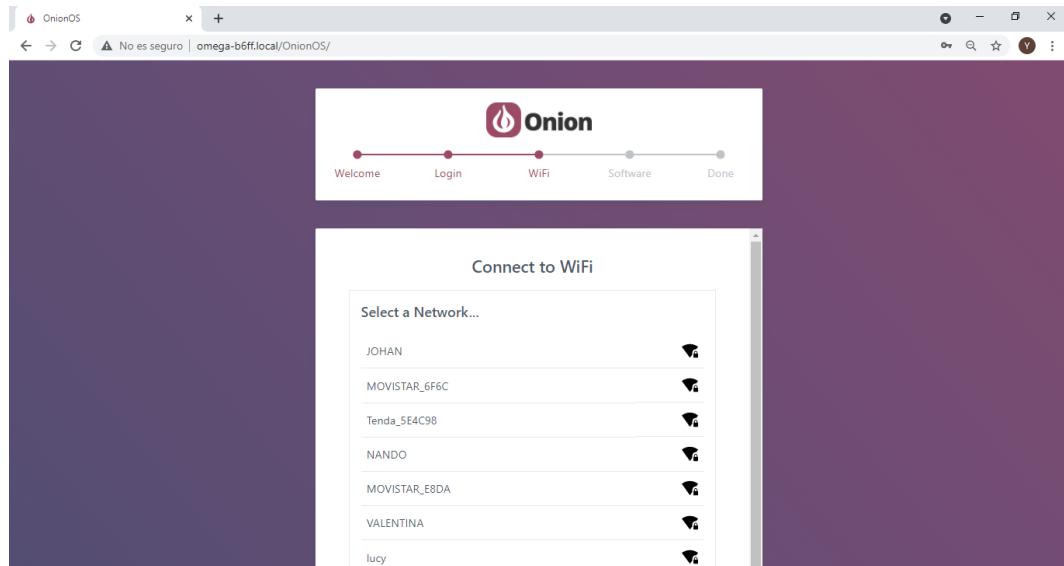


Figura 4.4: Selección de la red WiFi que conectará a internet al medidor inteligente B6FF.

Finalmente, una vez conectado el medidor inteligente a internet mediante WiFi, este buscará en el repositorio de OnionOS si existen actualizaciones para el firmware del dispositivo. Es recomendable que este proceso sea llevado a cabo, y una vez iniciado NO desconecte el medidor de la sesión HTTP ni lo apague o reinicie. El asistente de actualización de firmware se muestra en la figura 4.5.

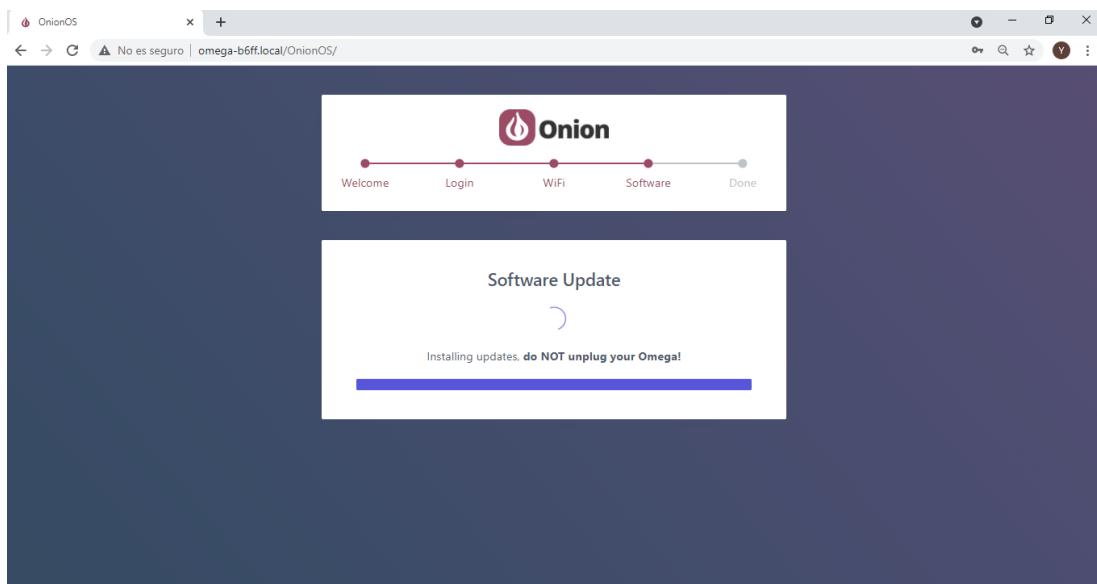


Figura 4.5: Actualización del firmware para el medidor inteligente B6FF.

Una vez finalizado, el medidor estará listo para usarse y este permitirá el acceso a la consola de comandos y la interfaz de configuración del dispositivo. A este punto el medidor queda a disposición del administrador para que desarrolle a nivel de hardware la aplicación que desee. La figura 4.6 muestra el gestor de aplicaciones del medidor.

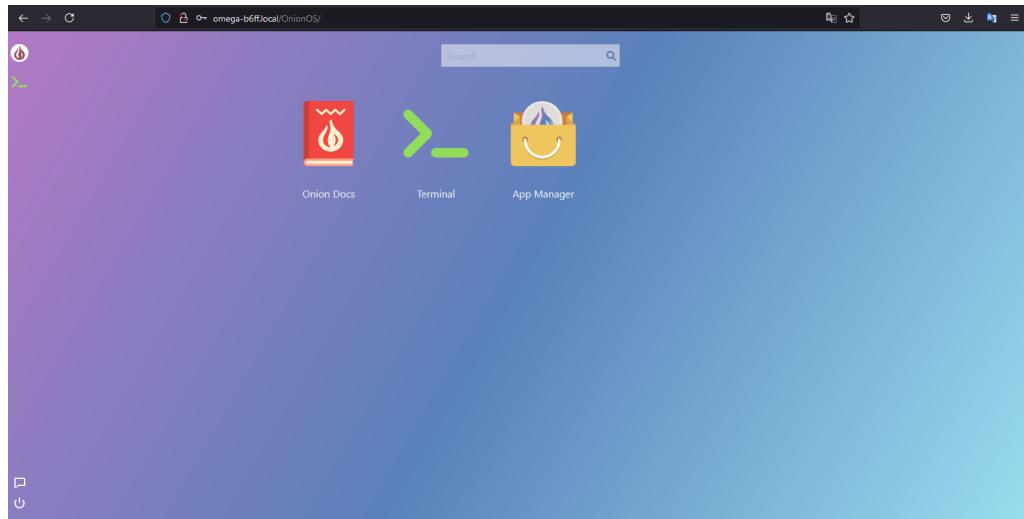


Figura 4.6: Acceso al gestor de aplicaciones para el medidor inteligente B6FF.

1.2.2. Conexión vía SSH

El protocolo utilizado por defecto para el proyecto para términos de facilidad, rapidez y seguridad, fue la conexión vía SSH, dado que basta con ingresar en el terminal o en el símbolo del sistema el siguiente comando:

```
ssh root@omega-XXXX.local
```

Donde los caracteres ‘XXXX’ representan el número de serie de cada módulo. Inmediatamente ingresado el comando, se debe ingresar la contraseña del usuario root del medidor en cuestión. La figura 4.7 muestra un ejemplo de acceso al medidor mediante comunicación SSH.

 A screenshot of a terminal window titled "OpenSSH SSH client". The window shows the following text:


```
Microsoft Windows [Versión 10.0.19042.985]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\johan>ssh root@omega-AE25.local
The authenticity of host 'omega-ae25.local (fe80::42a3:6bff:fea1:ae26%4)' can't be established.
RSA key fingerprint is SHA256:4MSXkh3gx2Y/yRuiQ6dy969z7tTucmAYNruzKV8W25Q.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'omega-ae25.local,fe80::42a3:6bff:fea1:ae26%4' (RSA) to the list of known hosts.
root@omega-ae25.local's password:

BusyBox v1.28.3 () built-in shell (ash)

/ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ 
W H A T \ W I L L \ Y O U \ I N V E N T ? \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ 
-----
```

The terminal then shows the Omega logo and the command "root@Omega-AE25:~# ls" followed by a list of files: "get-pip.py", "new", "prueba.txt", and "pw.txt".

Figura 4.7: Captura del terminal accediendo al medidor utilizando comunicación SSH.

Una medida alternativa a la consola de comandos (CMD) puede ser el programa disponible para Windows PuTTY, el cual ofrece una fácil configuración de la comunicación SSH en una interfaz amigable. La figura 4.8, muestra un ejemplo de la configuración de los parámetros de acceso correspondientes al medidor, en el cual, el parámetro ‘Host Name’ hace referencia a la dirección del dispositivo, en este caso, se debe ingresar el comando mencionado anteriormente ‘omega-XXXX.local’, dirigido hacia el puerto 22, el cual es el puerto destinado para la comunicación SSH.

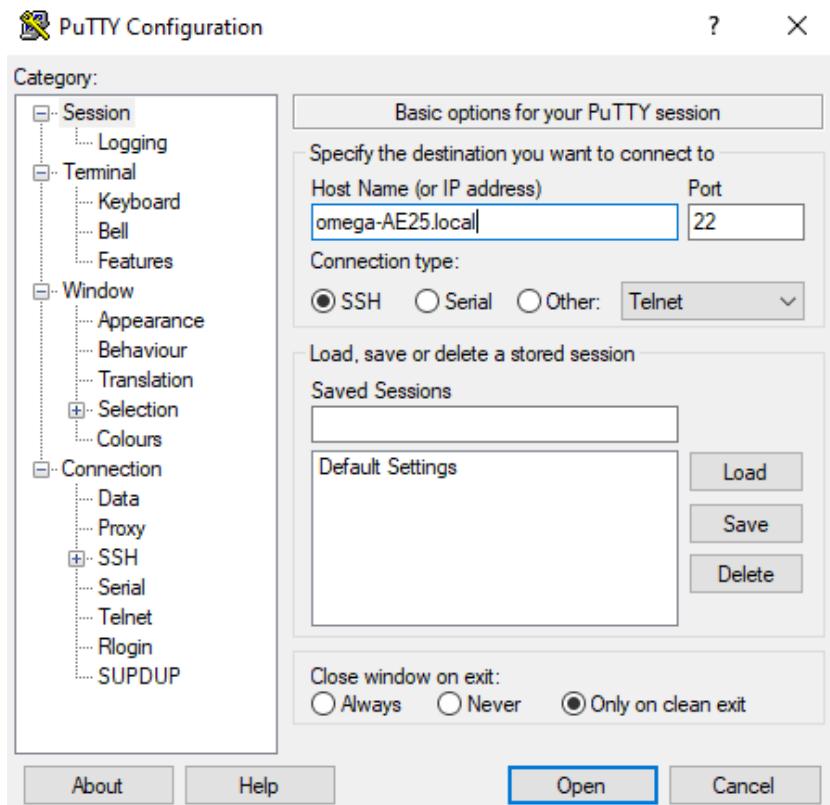


Figura 4.8: Configuración de los parámetros de conexión necesarios para la comunicación SSH sobre PuTTY.

```

root@Omega-AE25: ~
login as: root
root@omega-AE25.local's password:

BusyBox v1.28.3 () built-in shell (ash)

/ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / 
W H A T W I L L Y O U I N V E N T ? / _ _ _ / _ _ _ / _ _ _ / _ _ _ / _ _ _ / 
----- Q-ware: 0.3.2 b233 -----
root@Omega-AE25: ~#

```

Figura 4.9: Conexión establecida con el medidor desde la consola de PuTTY.

Una vez configurados estos parámetros y abierta la conexión, se desplegará una consola propia de PuTTY, en la cual, bastará con acceder mediante el usuario root y la contraseña del medidor, tal y como se muestra en la figura 4.9.

1.2.3. Conexión Ethernet

Para este método, basta simplemente conectar mediante un cable Ethernet la interfaz física de red de un computador (LAN) con la interfaz física de red del módulo (LAN), esto con el fin de interconectar los dispositivos y acceder a través de la consola mediante SSH; procedimiento descrito en la sección anterior. Este método es el menos recomendado, dado que es mucho más sencillo conectarse mediante la red WiFi del medidor, y realizar su configuración interna a través de ésta.

2. Configuración del medidor

En esta sección se pretender realizar una descripción con respecto a la configuración de un medidor eGeo Analyzer, desde el estado inicial de fábrica para ser utilizado en el proyecto.

2.1. Configuración técnica inicial

En caso de tener un medidor sin ningún tipo de configuración, ningún complemento o librerías instaladas, aquí se describe detalladamente los procesos que se deben realizar para llevar a cabo estas configuraciones, específicamente para uso propio del proyecto.

NOTA: Para el proceso de restauración al estado de fábrica del medidor, se debe acceder a la opción Factory Reset en el menú de configuración del medidor, tal y como se muestra en la figura 4.10. Si no se desea realizar de esa manera, se pueden ejecutar los siguientes comandos:

```
firstboot -y  
sync  
reboot
```

Una vez restaurado a su estado de fábrica, se debe acceder al medidor mediante alguno de los métodos explicados en la sección anterior. Para poder tener acceso a la red WiFi, la contraseña por defecto de la red WiFi del medidor es pw = ‘eGEOTech2018’ ó pw = ‘12345678’ y la dirección IPv4 por defecto del medidor se cambiará a 192.168.3.1.

Una vez conectados a la red WiFi del medidor por primera vez, es posible que el medidor realice una búsqueda e instalación de actualizaciones. Seguidamente, se debe acceder a la consola mediante el siguiente usuario y su respectiva contraseña:

```
user: root  
password: onioneer
```

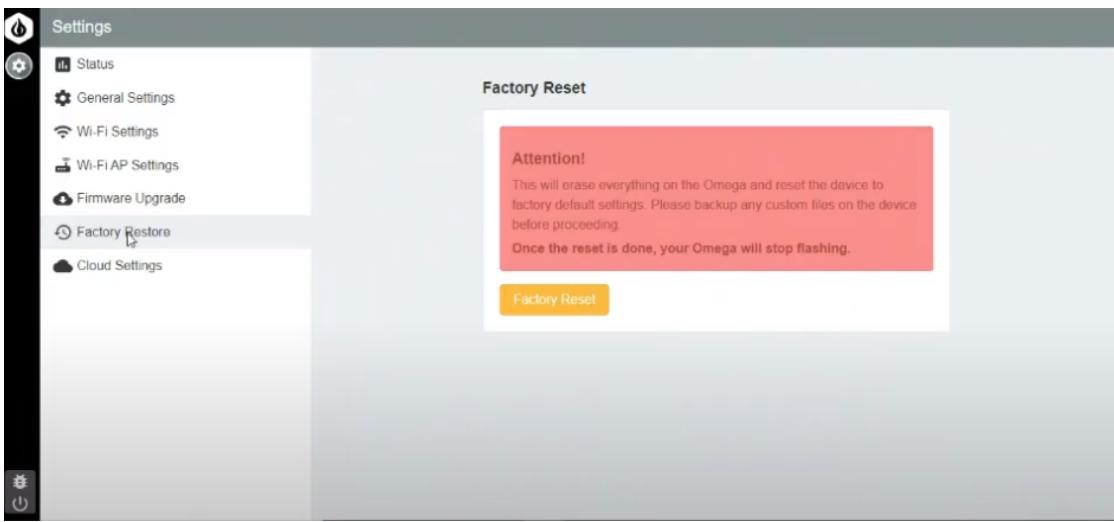


Figura 4.10: Opción gráfica de restauración al estado de fábrica predeterminada del medidor.

Dado que la dirección IPv4 por defecto del medidor es la misma para todos, se debe asignar la dirección IPv4 acordada por el fabricante, esto con el fin de evitar confusiones entre los medidores. Esta configuración se realiza mediante los siguientes comandos:

```
uci set network.wlan.ipaddr=192.168.3.x
uci commit network
/etc/init.d/network restart
```

Una vez cambiada la dirección IPv4, se debe asignar el número de serie del medidor en la red WiFi (por ejemplo para el medidor 1 se tiene que este sería ‘eGEO-B7F7’). Esta configuración se realiza mediante el siguiente comando:

```
uci set wireless.@wifi-iface[0].ssid=eGEO-xxxx
```

Luego de esto, se debe establecer la contraseña de la red WiFi del medidor, de la siguiente manera:

```
uci set wireless.@wifi-iface[0].key='eGEOtech900'
uci commit
```

Una vez configurados los datos técnicos del medidor, se debe establecer una contraseña de superusuario (root), diferente a la establecida de fábrica, por términos de seguridad. El siguiente comando realiza dicha acción:

```
passwd
```

Una vez se ingresa en consola el comando `passwd`, se deberá escribir la nueva contraseña a establecer y la consola pedirá confirmación. Finalizada esta parte, las variables técnicas del medidor habrán sido configuradas correctamente.

2.2. Instalación de complementos y librerías

En este apartado se muestra el proceso de instalación de los complementos y las librerías que necesita el medidor inteligente para poder ser utilizado correctamente, de manera que permita extraer y capturar las variables eléctricas de forma adecuada. Como se describe en el marco teórico, el medidor esta desarrollado en la distribución OnionOS de Linux y su lenguaje de desarrollo es Python, por lo tanto, se deben instalar las dependencias que permiten enlazar el hardware con el software.

Como en la mayoría de distribuciones de Linux, existe un administrador de paquetes que puede ser utilizado mediante un comando, para instalar actualizaciones, descargar y/o instalar paquetes prefabricados, desde repositorios de paquetes en internet. En este caso, OnionOS utiliza OpenWrt (OPKG), el cual permitirá instalar los complementos y librerías necesarios para el correcto funcionamiento del medidor. El primer paso es realizar su actualización mediante el siguiente comando:

```
opkg update
```

Una vez actualizado el administrador de paquetes, se realiza la instalación de ‘Python light’ y ‘pip3’. Se trabaja en Python, debido a que este es el lenguaje de desarrollo implementado para el medidor inteligente y se instala su versión light, por términos de espacio de almacenamiento. Se debe instalar ‘pip3’, pues este es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python; los cuales son necesarios para la implementación de los scripts desarrollados en el proyecto. Esta instalación se realiza de la siguiente manera:

```
opkg install python3-light  
opkg install python3-pip
```

Una vez finalizadas ambas instalaciones, se utiliza el sistema de gestión de paquetes ‘pip3’ para realizar la instalación de la biblioteca ‘setuptools’, la cual, es una biblioteca de procesos de desarrollo de paquetes diseñada para facilitar el encapsulamiento de proyectos en Python. Después de esto, se debe realizar la actualización de ‘pip’. Seguidamente, se realiza la instalación de ‘paho-mqtt’ como dependencia del protocolo mqtt, ‘pyserial’ como dependencia de la comunicación serial implementada desde Python, ‘DateTime’ para manejar y administrar formatos de tiempo, y por último, ‘pexpect’ para realizar y ejecutar comandos en consola dentro de un script de Python.

```
opkg update  
pip3 install --upgrade setuptools  
pip3 install --upgrade pip  
pip3 install paho-mqtt  
pip3 install pyserial  
pip3 install DateTime  
pip3 install pexpect
```

Finalmente, como procedimiento de utilidad, se realiza la configuración de la zona horaria de trabajo a -5 UTC, la cual es la zona horaria de Bogotá. Esta acción se realiza mediante el comando:

```
opkg update
onion time set America/Bogota COT5
```

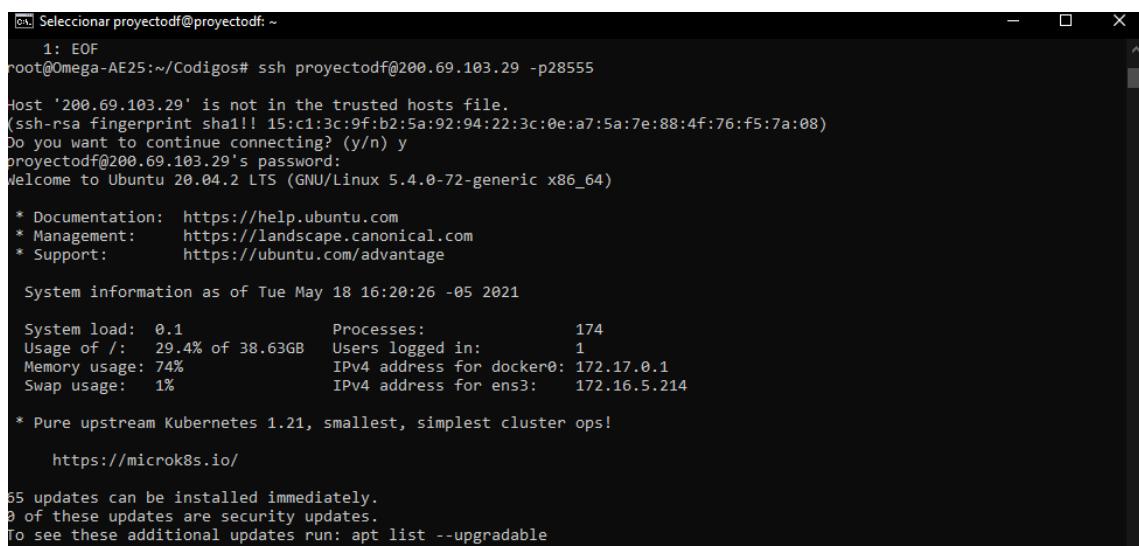
Una vez finalizado este procedimiento, se puede utilizar el medidor inteligente para capturar los parámetros de la red. Se recomienda probar el correcto funcionamiento del dispositivo mediante la captura de un dato. Un ejemplo de esta prueba se presenta a continuación:

```
opkg update
opkg install screen
screen /dev/ttyS1 230400
{"chip": "1", "operation": "getVoltageA"}
```

Una vez ejecutadas las líneas de comando, se debe mostrar en pantalla la tensión obtenida en la línea A del medidor. Se debe tener en cuenta que al momento de instalar el paquete ‘screen’, este paquete no permite visualizar los datos ingresados por teclado, por lo tanto, se recomienda pegar el comando de captura directamente en la terminal.

3. Códigos implementados

Antes de realizar cualquier acción desde un medidor recientemente restaurado a su estado de fábrica, o recientemente obtenido por parte del proveedor, se debe realizar una conexión directa desde el módulo hacia el servidor web, utilizando las credenciales del servidor mediante comunicación SSH. Ya que al conectarse mediante SSH al servidor web, este solicitará el uso de una clave SHA1 para permitir la conexión, por lo que el usuario deberá aceptar dicha petición, para poder conectarse satisfactoriamente al servidor.



The screenshot shows a terminal window titled 'Seleccionar proyecto df' with the command 'ssh proyecto df@200.69.103.29 -p28555'. It displays the SSH fingerprint and asks if the user wants to continue connecting (y/n). The password is entered, and the user is welcomed to Ubuntu 20.04.2 LTS. The system information is then displayed, including load average, memory usage, swap usage, and network interfaces. A note about Kubernetes is shown, followed by a link to microk8s.io. Finally, it lists available updates.

```
root@Omega-AE25:~/Códigos# ssh proyecto df@200.69.103.29 -p28555
Host '200.69.103.29' is not in the trusted hosts file.
(sshd-rsa fingerprint shall! 15:c1:3c:9f:b2:5a:92:94:22:3c:0e:a7:5a:7e:88:4f:76:f5:7a:08)
Do you want to continue connecting? (y/n) y
proyecto df@200.69.103.29's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-72-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue May 18 16:20:26 -05 2021

System load: 0.1      Processes:          174
Usage of /: 29.4% of 38.63GB  Users logged in:   1
Memory usage: 74%
Swap usage:  1%          IPv4 address for docker0: 172.17.0.1
                         IPv4 address for ens3:   172.16.5.214

* Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!

https://microk8s.io/

65 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable
```

Figura 4.11: Confirmación de codificación de contraseña SHA1 desde el medidor al servidor web.

Un ejemplo de esta situación, se puede evidenciar en la figura 4.11, en la que se observa al medidor omegaAE25 estableciendo una conexión por primera vez con el servidor (de no realizarse esta confirmación, no se podría acceder al servidor y las conexiones arrojarían errores). Una vez que son realizadas todas las configuraciones con las librerías, los complementos y los permisos, se pueden realizar los scripts necesarios. El código que captura, almacena y envía los parámetros de la red hacia las bases de datos se encuentra codificado en lenguaje Python 3, ya que este lenguaje permite utilizar de una forma más eficiente las librerías compatibles con el medidor.

3.1. Captura de los parámetros de manera serial

Antes de capturar cualquier parámetro de la red con el medidor, se debe realizar un script de Python con extensión .py que ejecute los comandos específicos para dicha captura. Antes de eso, en el código se deben importar las librerías que permiten realizar estas acciones, como lo son la librería “datetime” que permite el manejo y la creación de fechas y horas, la librería “serial” que permite utilizar la comunicación serial para la captura de los datos, la librería “json” que permite crear y manipular archivos con extensión .json y la librería “pexpect” que permite realizar y ejecutar comandos en consola mientras se ejecuta el script.

Luego de importar las librerías, lo siguiente es inicializar el objeto encargado de la comunicación serial, tal y como se muestra a continuación:

```
Ser1 = serial.Serial('/dev/ttyS1', 230400, timeout = 0.3)
```

En el comando anterior, se establece que la variable “Ser1” es un objeto de tipo serial, que se comunica con el puerto serial /dev/ttyS1, con una velocidad en baudios de 230400 y un timeout de 0.3, este último valor hace referencia al tiempo predeterminado para realizar la ejecución del comando serial sin presentar ningún tipo de error.

Con el fin de mantener un código funcional, las capturas de los parámetros de la red, se realizan dentro de funciones que tienen como argumentos de entrada: La línea que se desea capturar, el chip que se está utilizando, el objeto serial y el valor anterior de la medida; de la misma forma, retornan el valor de la variable capturada, el tiempo de captura y un valor de la medida para utilizar como argumento de entrada en la siguiente llamada a la función. Para el caso de una variable como la tensión, que depende de la línea que se desea capturar, la función es la siguiente:

```
def Captura_Tension(Linea, Chip, Ser1, Constante_Tension):
    Tiempo_Tension = datetime.datetime.today()
    if (Chip == 1):
        if (Linea == "A"):
            Ser1.write("""{"chip": "1", "operation": "getVoltageA"}""".encode())
        elif (Linea == "B"):
            Ser1.write("""{"chip": "1", "operation": "getVoltageB"}""".encode())
        else:
            Ser1.write("""{"chip": "1", "operation": "getVoltageC"}""".encode())
    else:
        if (Linea == "A"):
            Ser1.write("""{"chip": "2", "operation": "getVoltageA"}""".encode())
        elif (Linea == "B"):
            Ser1.write("""{"chip": "2", "operation": "getVoltageB"}""".encode())
```

```

        else:
            Ser1.write("""{"chip": "2","operation": "getVoltageC"}""".encode())
Lectura_Serial_Tension = Ser1.readline()
if (len(Lectura_Serial_Tension)!=0):
    jsonTension = json.loads(Lectura_Serial_Tension)
    Tension = jsonTension['value']
    Constante_Tension = Tension
else:
    Tension = Constante_Tension
return Tension, Tiempo_Tension, Constante_Tension

```

Cuando se realiza la llamada a la función para la captura de la variable tensión, denominada “Captura_Tension”, la primera acción que realiza es obtener la fecha y el tiempo exacto de la captura; lo siguiente es capturar el dato de tensión, para ello, se realizan unas sentencias condicionales, dependiendo del chip y de la línea ingresados en la función, con estos valores se genera el comando de captura y se sobre-escribe en el objeto serial. Cuando ya se tiene el determinado comando de captura, se realiza una acción de lectura en el objeto serial, para obtener los datos devueltos.

Lo siguiente es evaluar si el resultado devuelto es diferente de vacío, en caso afirmativo, se carga el resultado de la lectura como una variable json y se extrae el valor de la tensión, para ser almacenada en la variable “Tension”, además, se actualiza la constante con este nuevo valor; en caso contrario, se le asigna a la variable “Tension” el valor almacenado en la constante. Finalmente la función retorna el valor capturado, el tiempo en que se capturó y el valor de la constante.

Para la captura de algunas variables como la frecuencia o la temperatura, no se necesita información a cerca de las líneas, solo del chip y del objeto serial. La función de captura de estas variables es bastante similar a las otras, la función de captura de la temperatura del medidor, se muestra a continuación:

```

def Captura_Temperatura(Chip, Ser2, Constante_Temperatura):
    Tiempo_Temperatura = datetime.datetime.today()
    if (Chip == 1):
        Ser2.write("""{"chip": "1","operation": "getTemp"}""".encode())
    else:
        Ser2.write("""{"chip": "2","operation": "getTemp"}""".encode())
    Lectura_Serial_Temperatura = Ser2.readline()
    if (len(Lectura_Serial_Temperatura)!=0):
        jsonTemperatura = json.loads(Lectura_Serial_Temperatura)
        Temperatura = jsonTemperatura['value']
        Constante_Temperatura = Temperatura
    else:
        Temperatura = Constante_Temperatura
    return Temperatura, Tiempo_Temperatura, Constante_Temperatura

```

Como se menciona anteriormente, la función de captura para estas variables se comporta de forma similar al resto. Inicialmente se obtiene el valor de la fecha y hora de captura y luego se sobre-escribe el comando de captura en el objeto serial, esta vez, los condicionales únicamente dependen del chip ingresado para generar este comando. Al igual que con las otras variables, se realiza la lectura del objeto serial y se evalúa si este resultado es diferente de vacío, de ser así, se extrae el valor del parámetro de interés y se almacena en la variable correspondiente, sin olvidar actualizar el valor de la constante; en

caso de que el resultado se encuentre vacío, la variable se actualiza con el valor almacenado en la constante. Finalmente, la función retorna el valor capturado, el tiempo en que se capturó y el valor de la constante.

3.2. Detección del estado para los parámetros de la red

Tal y como fue mencionado en capítulos anteriores, existen diferentes fallas que pueden ser detectadas a partir de los datos capturados con el medidor. Cabe resaltar que estas fallas dependen de ciertos parámetros en particular, no de todas las variables capturadas por el medidor; como por ejemplo, el caso de las variaciones de frecuencia, las cuales dependen únicamente de la frecuencia capturada de la red y se presentan si se encuentra fuera del rango de operación normal.

Al igual que el valor de la variable capturada y el tiempo de captura, otro parámetro que tiene bastante relevancia es el estado de la variable. Para detectar este estado en cada variable, se genera una función de detección de estado, que permite asignarle un estado a la variable capturada, dependiendo el valor que tenga dicha variable. Este estado se asigna como un parámetro extra, cuando el valor de la variable se encuentra en un rango específico; si este rango equivale al rango de operación esperado, el estado de la variable tendrá un valor “Normal”, en caso contrario se deben realizar varios criterios y condiciones para determinar este estado, según la variable en cuestión.

Tal es el caso de la tensión, que puede tener diferentes estados según el valor capturado, los cuales pueden ser: Colapso, interrupción, caída, reducción, subtensión, normal, elevación, sobretensión y crítico. Como también entra a colación el caso de la frecuencia que solo puede tener dos estados: Normal y variación. La función de detección de estado de la frecuencia, se presenta a continuación, con su respectiva explicación:

```
def Deteccion_Estado_Frecuencia(Chip, Frecuencia, Tiempo, Tiempo_Variacion):
    Frecuencia_Normal = 60
    if (Frecuencia >= 0.96*Frecuencia_Normal and Frecuencia <= 1.04*Frecuencia_Normal):
        Estado = "Normal"
    if (Frecuencia < 0.96*Frecuencia_Normal or Frecuencia > 1.04*Frecuencia_Normal):
        Tiempo_Variacion.append(Tiempo)
        Estado = "Variacion"
    else:
        if (Tiempo_Variacion != []):
            Duracion_Variacion = Tiempo_Variacion[len(Tiempo_Variacion)-1] - Tiempo_Variacion[0]
            if (len(Tiempo_Variacion) == 1):
                Duracion_Variacion= Tiempo - Tiempo_Variacion[0]
            Duracion_Variacion_Segundos = Duracion_Variacion.total_seconds()
            if (Duracion_Variacion_Segundos > 0):
                print("""Hubieron variaciones de frecuencia
                en el Chip""", Chip, "durante:", Duracion_Variacion)
            Tiempo_Variacion = []
    return Estado, Tiempo_Variacion
```

En la función de detección del estado de la frecuencia ingresan como argumentos de entrada el chip que se está utilizando, el valor de la frecuencia capturado, el tiempo de captura y el tiempo de duración de la variación que se utiliza como un acumulador. Lo primero que se realiza es ajustar el valor de la frecuencia en estado normal, que para Colombia equivale a 60 Hz, lo siguiente es evaluar si el valor capturado se encuentra dentro del rango de operación esperado, el cual es del 4 % por encima y por debajo del valor de

frecuencia normal, si este es el caso, se le asigna un estado “Normal”. En caso contrario, a la variable de tipo lista denominada “Tiempo_Variacion” se le agrega el tiempo de captura y se cambia el estado a un valor “Variacion”.

Si ocurre el caso donde el estado cambia de “Variacion” a “Normal”, se evalúa si el arreglo de la duración de la variación es diferente de vacío, en caso afirmativo, se calcula el tiempo de duración de la falla restando el ultimo valor de tiempo menos el primer valor de tiempo en el arreglo; este tiempo se almacena en la variable “Duracion_Variacion”, la cual es una variable de tipo datetime, por lo que se debe pasar a segundos para poder mostrarla en consola. Si este valor de tiempo es mayor a 0 segundos, se imprime en pantalla que se detectaron variaciones de frecuencia en el respectivo chip durante la cantidad de tiempo calculada, por último se borran los datos en el arreglo de tiempos de variación.

La impresión en pantalla solo ocurre cuando se cambia de un estado de variación a un estado normal, en caso de mantenerse el estado normal o cambiar a variación no se imprime nada en consola; esto se realiza con el fin de solo mostrar en pantalla las duraciones de las fallas más no los cambios de estado en sí. Finalmente la función retorna el valor de estado y el arreglo de tiempos de variación para ser utilizado en la siguiente llamada de la función.

3.3. Imprimir en pantalla cada parámetro de la red

Cada parámetro capturado por el medidor posee más de diez dígitos decimales, es por esta razón que debe acortarse a un número de dígitos decimales que representen una mayor información en la aplicación, por lo tanto, este valor se debe acotar al numero de decimales deseados; además el usuario debe poder visualizar estos datos al momento de la captura, de una forma ordenada independientemente de las unidades de la variable o la notación empleada. Para llevar a cabo esta tarea, se emplea una función que permite realizar estas acciones de una manera sencilla y funcional, para cada una de las variables. La función que muestra en pantalla los valores correspondientes a la potencia se muestra a continuación:

```
def Mostrar_Potencia(Linea, Chip, Potencia, Tiempo_Potencia, Estado_Potencia):
    Potencia_Abs = abs(Potencia)
    Potencia_str = str(Potencia_Abs)
    pos = Potencia_str.find(".")
    Potencia_decimal = Potencia_str[pos:]
    while (len(Potencia_decimal)<=6):
        Potencia_decimal = Potencia_decimal + '0'
    Potencia_completa = Potencia_str[:pos] + Potencia_decimal[:7]
    if (Potencia_Abs >= 1000):
        if (len("{0:.3f}".format(Potencia_Abs/1000)) >= 6):
            print("Potencia Línea", Linea, "chip", Chip, ":", """
            """, "{0:.3f}".format(Potencia_Abs/1000), """ [kW],
            Tiempo:""", Tiempo_Potencia, ", Estado:", Estado_Potencia)
        else:
            print("Potencia Línea", Linea, "chip", Chip, ":", """
            """, "{0:.3f}".format(Potencia_Abs/1000), """ [kW],
            Tiempo:""", Tiempo_Potencia, ", Estado:", Estado_Potencia)
    elif (Potencia_Abs < 1):
        if (len("{0:.3f}".format(Potencia_Abs*1000)) >= 7):
            print("Potencia Línea", Linea, "chip", Chip, ":", """
            """, "{0:.3f}".format(Potencia_Abs*1000), """ [mW],
            Tiempo:""", Tiempo_Potencia, ", Estado:", Estado_Potencia)
```

```

        elif (len("{0:.3f}".format(Potencia_Abs*1000)) == 6):
            print("Potencia Línea", Linea, "chip", Chip, " :", """
            "" {0:.3f}".format(Potencia_Abs*1000), "" [mW],
            Tiempo:"", Tiempo_Potencia, ", Estado:", Estado_Potencia)
        else:
            print("Potencia Línea", Linea, "chip", Chip, " :", """
            "" {0:.3f}".format(Potencia_Abs*1000), "" [mW],
            Tiempo:"", Tiempo_Potencia, ", Estado:", Estado_Potencia)
    else:
        if (len("{0:.3f}".format(Potencia_Abs)) >= 7):
            print("Potencia Línea", Linea, "chip", Chip, " :", """
            "" {0:.3f}".format(Potencia_Abs), "" [W],
            Tiempo:"", Tiempo_Potencia, ", Estado:", Estado_Potencia)
        elif (len("{0:.3f}".format(Potencia_Abs)) == 6):
            print("Potencia Línea", Linea, "chip", Chip, " :", """
            "" {0:.3f}".format(Potencia_Abs), "" [W],
            Tiempo:"", Tiempo_Potencia, ", Estado:", Estado_Potencia)
        else:
            print("Potencia Línea", Linea, "chip", Chip, " :", """
            "" {0:.3f}".format(Potencia_Abs), "" [W],
            Tiempo:"", Tiempo_Potencia, ", Estado:", Estado_Potencia)
    return Potencia_completa

```

La función que permite mostrar en pantalla los valores de las variables, tiene como argumentos de entrada: La línea (si la requiere), el chip, el valor capturado, el tiempo de captura y el estado de la variable. La primera acción que realiza la función es obtener el valor absoluto del valor capturado, en caso de que posea un valor negativo por el sentido en que se midió la corriente, lo siguiente es convertir a String el valor para poder trabajar mejor, ya que Python no permite redondear los números con punto flotante correctamente, por lo que trabajarlos en formato String resulta más conveniente. Luego de la conversión, se separa la parte entera y la parte decimal del valor, dejando la parte decimal con 6 dígitos, eliminando el resto si posee más de 6 dígitos o llenando con '0' si posee menos dígitos.

La siguiente tarea es imprimir los respectivos valores de la variable en pantalla, para ello, se evalúa el valor original para determinar el rango en el que se encuentra, de esta forma adecuar las unidades para que el valor aparezca más limpio en la pantalla, además de eso, en pantalla se muestran únicamente 3 dígitos decimales para cada variable, el valor se ajusta dependiendo sus unidades; seguido a los valores, se muestra el tiempo de captura y el estado de la variable. Finalmente, la función retorna el valor de la variable capturada solo con 6 dígitos decimales en formato String para procesos siguientes.

En el caso de la potencia, como se muestra en el código de la función, si el valor supera los 1000 W, el valor se ajusta a 3 dígitos y las unidades a kW. Hacia el otro extremo, si el valor es menor que 1 W, el valor se ajusta a 3 dígitos decimales y las unidades a mW; de esta manera se asegura que la longitud del texto mostrado en pantalla sea fija, independientemente de la variable, linea, chip o valor de la variable, asemejándose a una tabla.

La longitud del texto solo puede cambiar al final en el estado de la variable, pues todas las variables poseen estados distintos. La impresión en pantalla es inmediata una vez se captura el valor del parámetro, el tiempo y el estado. Su flujo es hacia abajo de forma que solo se interrumpe cuando se detecta un cambio de estado en algún parámetro y se imprime el texto que muestra la duración de dicho cambio.

3.4. Estructuración y transferencia del archivo en formato JSON

Las funciones de captura de los datos, las detecciones de estado y las impresiones en pantalla de los valores, se realizan de forma secuencial dentro de un ciclo infinito generado con una sentencia while verdadera. Dentro de este ciclo se ejecutan todas estas funciones en orden, separados por chip, por variables y por líneas, si es el caso. Como se obtienen y visualizan estos valores de forma ordenada, también se almacenan de forma ordenada, dentro de cada variable, dentro del chip, dentro del ciclo while, se estructura una variable de tipo diccionario que permite almacenar todos y cada uno de estos valores. La estructura de la variable de tipo diccionario y el segmento de código que inserta los datos en esta variable se muestra a continuación para el parámetro de corriente:

```
Variables = ["TENSION", "CORRIENTE", "FRECUENCIA",
"TEMPERATURA", "THD_V", "THD_I", "POTENCIA", "FASÉ"]

def Borrar_JSON(Variables):
    Datos_1 = {} ; Datos_2 = {}
    for Variable in range(len(Variables)):
        Datos_1[Variables[Variable]] = []
        Datos_2[Variables[Variable]] = []
    return Datos_1, Datos_2
```

Antes de empezar el ciclo while, se debe crear la variable que almacenará los datos, para ello se utiliza la función “Borrar_JSON”, la cual permite crear la variable, como limpiarla y dejarla vacía. La estructura de la variable de datos es simple, es un diccionario que posee 8 diferentes elementos, cada uno de ellos es un arreglo, que hace referencia a cada una de las variables medibles. Cada uno de los elementos dentro de los arreglos internos, es un conjunto de datos capturados por el medidor en un ciclo de ejecución, porque lo que dependiendo del parámetro, esto conjunto de datos puede ser de 3 o de 9 valores (estos valores son los datos del parámetro, el tiempo y el estado, en caso de tener distintas líneas, pues serán estos datos por cada línea). La longitud del arreglo, es equivalente a los ciclos que se almacenaron los datos.

```
# CORRIENTE
# Línea A
Corriente_A1, Tiempo_Corriente_A1, Constante_Corriente_A1 = Captura_Corriente(
Linea_A, Chip_1, Ser2, Constante_Corriente_A1)
Estado_Corriente_A1, Tiempo_Alerta_A1, Tiempo_Critico_A1 = Deteccion_Estado_Corriente(
Linea_A, Chip_1, Corriente_A1, Tiempo_Corriente_A1, Tiempo_Alerta_A1, Tiempo_Critico_A1)
Corriente_A1 = Mostrar_Corriente(
Linea_A, Chip_1, Corriente_A1, Tiempo_Corriente_A1, Estado_Corriente_A1)
# Línea B
Corriente_B1, Tiempo_Corriente_B1, Constante_Corriente_B1 = Captura_Corriente(
Linea_B, Chip_1, Ser2, Constante_Corriente_B1)
Estado_Corriente_B1, Tiempo_Alerta_B1, Tiempo_Critico_B1 = Deteccion_Estado_Corriente(
Linea_B, Chip_1, Corriente_B1, Tiempo_Corriente_B1, Tiempo_Alerta_B1, Tiempo_Critico_B1)
Corriente_B1 = Mostrar_Corriente(
Linea_B, Chip_1, Corriente_B1, Tiempo_Corriente_B1, Estado_Corriente_B1)
# Línea C
Corriente_C1, Tiempo_Corriente_C1, Constante_Corriente_C1 = Captura_Corriente(
Linea_C, Chip_1, Ser2, Constante_Corriente_C1)
Estado_Corriente_C1, Tiempo_Alerta_C1, Tiempo_Critico_C1 = Deteccion_Estado_Corriente(
Linea_C, Chip_1, Corriente_C1, Tiempo_Corriente_C1, Tiempo_Alerta_C1, Tiempo_Critico_C1)
Corriente_C1 = Mostrar_Corriente(
Linea_C, Chip_1, Corriente_C1, Tiempo_Corriente_C1, Estado_Corriente_C1)
# Subir al JSON
Datos_1["CORRIENTE"].append({"Corriente_A": Corriente_A1,
```

```

"Tiempo_A": str(Tiempo_Corriente_A1), "Estado_A": Estado_Corriente_A1,
"Corriente_B": Corriente_B1, "Tiempo_B": str(Tiempo_Corriente_B1),
"Estado_B": Estado_Corriente_B1, "Corriente_C": Corriente_C1,
"Tiempo_C": str(Tiempo_Corriente_C1), "Estado_C": Estado_Corriente_C1})

```

Como se evidencia en el código, las funciones se ejecutan por parámetros y separadas por lineas (si se presenta el caso), como todos estos valores son diferentes, se tienen que manejar cada una en una variable, hasta que al final se cargan dentro de la variable de tipo diccionario denominada “Datos_1”, en el espacio asignado para corriente; allí se almacenan los nueve valores (tres por cada linea: Valor, tiempo y estado) de una forma ordenada y secuencial por cada ciclo donde se cargue la variable “Datos_1”. Cabe resaltar que este segmento del código es para el parámetro de corriente en el chip 1, por ese motivo las variables suelen tener un nombre con un sufijo “_1”.

De esta manera se generan dos variables de tipo diccionario, una para cada chip, que almacenan los datos en cada ciclo, sin embargo, para que estos datos se puedan enviar al servidor web se deben pasar en un archivo en formato json. La cantidad de datos que se desea transferir depende de la aplicación o el uso que se quiera dar a los datos, en este caso se desea visualizar una cierta cantidad de tiempo en la interfaz gráfica, por lo tanto, al final de cada ciclo se evalúa si ya se cumplió este tiempo; en caso negativo, se ejecuta otro ciclo y se almacenan los datos con normalidad, en caso afirmativo, los datos se convierten en formato json y se envían al servidor. El segmento de código que permite realizar estas acciones se muestra a continuación:

```

IP = "200.69.103.29"; Puerto = "28555"; Usuario = "proyectodf";
Contramarca = "cecad*proyectodf"; Ruta = "/home/proyectodf/"; Carpeta = "Datos";

def Transferir_JSON(Datos_1, Datos_2, IP, Puerto, Usuario, Contramarca, Carpeta):
    with open("/root/" + Carpeta + "/Datos_1.json", "w") as file:
        json.dump(Datos_1, file, indent=4, ensure_ascii=False)
    with open("/root/" + Carpeta + "/Datos_2.json", "w") as file:
        json.dump(Datos_2, file, indent=4, ensure_ascii=False)
    Comando_Transferencia = pexpect.spawn(
        "scp -P " + Puerto + " /root/" + Carpeta + "/Datos_1.json /root/" + Carpeta +
        "/Datos_2.json " + Usuario + "@" + IP + ":" + Ruta + Carpeta + "/")
    Reaccion_Transferencia = Comando_Transferencia.expect(
        [Usuario + "@" + IP + "'s password: ", pexpect.EOF])
    if (Reaccion_Transferencia == 0):
        Comando_Transferencia.sendline(Contramarca)
        Comando_Transferencia.expect(pexpect.EOF)
    else:
        print("Error de autenticación de credenciales")

```

En el segmento de código anterior, se encuentran las credenciales que permiten el acceso al servidor desde cualquier dispositivo, estas credenciales son de suma importancia, pues se emplean repetidamente en el código para la comunicación con las bases de datos. También se muestra la función de nominada “Transferir_JSON”, la cual tiene como argumentos de entrada los dos diccionarios con los datos y todas las credenciales mencionadas.

Inicialmente, la función genera dos diferentes archivos con extensión .json (uno para cada variable de datos) con los datos ingresados y se guardan de forma local en una carpeta dentro del medidor. Lo siguiente es ejecutar un comando en consola para establecer

la comunicación con el servidor web y además trasferir estos archivos json, este comando utiliza el protocolo SCP para esta transferencia y las credenciales para comunicarse correctamente. Debido a que el servidor web posee contraseña, esta debe ser ingresada cuando el servidor realice la solicitud, es en esta situación donde la librería pexpect se emplea, ya que esta librería permite ejecutar los comandos en consola y esperar una respuesta, dependiendo si esta respuesta es la esperada se pueden ejecutar otras acciones. Para este caso, después de ejecutar el comando scp se queda esperando la solicitud de contraseña del servidor, una vez que la recibe, ingresa la contraseña establecida inicialmente en las credenciales y se transfieren los archivos; en caso de presentarse algún problema en la comunicación se muestra un aviso en consola indicando que se presentó un error de autenticación y el código sigue con su funcionamiento sin interrupciones.

Como es mencionado anteriormente, esta función solo se ejecuta cuando el tiempo de subida o el tiempo establecido para subir los datos al servidor ya se ha cumplido, de lo contrario, las variables simplemente siguen almacenando datos. Después de haber ejecutado la función correctamente, los archivos se encuentran ubicados en la ruta especificada en las credenciales y se puede acceder a ellos desde el servidor sin ningún inconveniente.

3.5. Comunicación con las bases de datos desde el medidor

Ahora que ya se sabe como capturar todos los valores relacionados a los parámetros de la red, como almacenarlos en un archivo de manera local y como transferir este archivo al servidor, lo siguiente, es cargar los datos desde el archivo ubicado en el servidor en las bases de datos. La idea es que la comunicación entre las bases de datos y el medidor sea automática, para ello, se deben ejecutar comandos desde el script de Python del medidor que ejecuten acciones directamente en el servidor y a su vez sobre las bases de datos. Las principales acciones que se requieren son: Vaciar las tablas que contengan las bases de datos, insertar los datos en las tablas de las bases de datos y limpiar los datos en las tablas de las bases de datos.

3.5.1. Vaciar las tablas de las bases de datos

El proceso de vaciado de las bases de datos, como tal no es un requerimiento, es más un proceso opcional que permite que las tablas de las bases de datos se encuentren completamente vacías antes de que se inserten datos en ellas. Este proceso se realiza opcionalmente antes de ejecutar el ciclo while en el script, a través de una función denominada “Vaciar_DB”, cuyo código y explicación se muestran a continuación:

```
def Vaciar_DB(IP, Puerto, Usuario, Contramarca):
    Comando_Vaciar = pexpect.spawn(
        "ssh " + Usuario + "@" + IP + " -p " + Puerto +
        " cd Funciones; python3 Vaciar_sistemas.py")
    Reaccion_Vaciar = Comando_Vaciar.expect([
        [Usuario + "@" + IP + "'s password: ", pexpect.EOF]])
    if (Reaccion_Vaciar == 0):
        Comando_Vaciar.sendline(Contramarca)
        Comando_Vaciar.expect(pexpect.EOF)
    else:
        print("Error de autenticación de credenciales")
```

Similar a la función “Transferencia_JSON”, esta función ejecuta un comando a través de la librería pexpect, sin embargo, este comando no es un comando de transferencia, sino es un comando ssh para establecer la conexión con el servidor web. Este comando de conexión utiliza las credenciales ingresadas a la función como argumentos de entrada para comunicarse con el servidor y ejecutar unos comandos internamente. Las acciones que realizan estos comandos son: Cambiar el directorio a la carpeta “Funciones” y allí ejecutar el script de Python 3 llamado “Vaciar_sistemas.py”. Lo siguiente que realiza la función es esperar la solicitud de contraseña por parte del servidor y si es la esperada, inyectar la contraseña adecuada. Si la comunicación con el servidor no presenta ningún inconveniente, se ejecuta internamente el script de Python establecido en el comando.

Una vez que en el código del medidor se llevó a cabo la función “Vaciar_DB”, en el servidor se ejecuta el script de Python 3 denominado como “Vaciar_sistemas.py”. Este script ejecuta las acciones para comunicarse con las bases de datos y vaciar todas las tablas en ellas. El código de este script se muestra a continuación:

```
import pymysql

def Vaciar_DB(Conexion, Tablas):
    for Tabla in range(len(Tablas)):
        with Conexion.cursor() as Cursor:
            Consulta = "SELECT * FROM " + Tablas[Tabla] + ";"
            Cursor.execute(Consulta)
            Respuestas = Cursor.fetchall()
            if (Respuestas != ()):
                Truncar = "TRUNCATE " + Tablas[Tabla] + ";"
                Cursor.execute(Truncar)

try:
    Conexion_1 = pymysql.connect(host='0.0.0.0', port=6601,
                                user='proyectodf', passwd='cecad*proyectodf', db='CHIP_1',
                                charset='utf8', cursorclass=pymysql.cursors.DictCursor)
    Conexion_2 = pymysql.connect(host='0.0.0.0', port=6602,
                                user='proyectodf', passwd='cecad*proyectodf', db='CHIP_2',
                                charset='utf8', cursorclass=pymysql.cursors.DictCursor)
    print("Conexión correcta")
except (pymysql.err.OperationalError, pymysql.err.InternalError) as Error:
    print("Ocurrió un error al conectar: ", Error)

Tablas = ["TENSION", "CORRIENTE", "FRECUENCIA", "TEMPERATURA",
          "THD_V", "THD_I", "POTENCIA", "FASE"]
Vaciar_DB(Conexion_1, Tablas)
Vaciar_DB(Conexion_2, Tablas)
```

Como se observa en el código, este script necesita importar una librería llamada “pymysql” que permite realizar la comunicación de Python con las bases de datos en MySQL. Este script también declara una función llamada “Vaciar_DB” que tiene como argumentos de entrada una variable de tipo conexión y un arreglo con el nombre de las tablas existentes en la base de datos. Dentro de esta función se genera un ciclo for que recorre todos los elementos del arreglo con los nombres de las tablas y crea un cursor a partir de la variable de tipo conexión; dentro de esta conexión se realiza una consulta en cada tabla con el comando de mysql “SELECT”, extrayendo todos los datos que contenga y se evalúa si el resultado se encuentra vacío. En caso de existir algún dato dentro de la tabla, se realiza el comando de mysql “TRUNCATE” que vacía la tabla correspondiente.

De esta manera se eliminan todos los datos en las tablas que se encuentren en el arreglo denominado “Tablas”.

Fuera de la función se establece la conexión para cada chip del medidor, utilizando la librería pymysql permite establecer la conexión para cada base de datos, ingresando las credenciales adecuadas. De esta manera se genera una variable de tipo conexión para cada base de datos que se puede utilizar de forma independiente, en caso de presentarse algún problema de autenticación en las credenciales, se genera un aviso en consola indicando el error encontrado.

Finalmente, se escribe la variable global que contiene el nombre de todas las tablas en cada base de datos y se ejecuta la función de vaciado con cada uno de las variables de tipo conexión, es decir, se realiza el vaciado en ambas bases de datos. Cabe resaltar que el script tiene el sufijo “_sistemas” ya que se realiza solo para el medidor ubicado en las salas de sistemas; los medidores ubicados en las salas de laboratorio y en el grupo LIFAE utilizan scripts con sufijos distintos, de la misma forma, para establecer las conexiones con las bases de datos, emplean puertos distintos, pues hacen referencia a distintos contenedores.

3.5.2. Insertar los datos en las tablas de las bases de datos

La inserción de los datos en las tablas de las bases de datos se realizan únicamente después de transferir el archivo json, de esta manera se asegura que se inserten solo datos nuevos en las tablas. El proceso es similar al vaciado de las bases de datos y se realiza a través de una función, cuyo código y explicación se muestran a continuación:

```
def Insertar_Datos(IP, Puerto, Usuario, Contramarca):
    Comando_Insertar = pexpect.spawn(
        "ssh " + Usuario + "@" + IP + " -p" + Puerto +
        " cd Funciones; python3 Insertar_sistemas.py")
    Reaccion_Insertar = Comando_Insertar.expect([
        [Usuario + "@" + IP + "'s password: ", pexpect.EOF]])
    if (Reaccion_Insertar == 0):
        Comando_Insertar.sendline(Contramarca)
        Comando_Insertar.expect(pexpect.EOF)
    else:
        print("Error de autenticación de credenciales")
```

De igual forma que con la función de vaciado, se ejecuta el comando para ejecutar el script en el servidor, se utilizan las credenciales para la autenticación y se espera la solicitud de la contraseña, si es la correcta, se inyecta la contraseña y se ejecuta el script de Python denominado “Insertar_sistemas.py”. Este script ejecuta las acciones para comunicarse con las bases de datos e insertar los datos del servidor en las tablas de cada base de datos. El código de este script se muestra a continuación:

```
import pymysql
import json

def Insertar_Datos(Conexion, Datos, Tablas):
    for Tabla in range(len(Tablas)):
        with Conexion.cursor() as Cursor:
            if (Tablas[Tabla] == "FRECUENCIA" or Tablas[Tabla] == "TEMPERATURA"):
                Insertar = "INSERT INTO " + Tablas[Tabla] + "(VALOR, TIEMPO,
```

```

        ESTADO) VALUES (%s, %s, %s);"
else:
    Insertar = "INSERT INTO " + Tablas[Tabla] + "(LINEA_A, TIEMPO_A,
ESTADO_A, LINEA_B, TIEMPO_B, ESTADO_B, LINEA_C, TIEMPO_C,
ESTADO_C) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s);"
for Ciclo in range(len(Datos[Tablas[Tabla]])):
    Registro = list()
    for Cifra in Datos[Tablas[Tabla]][Ciclo].values():
        Registro.append(Cifra)
    Cursor.execute(Insertar, Registro)
Conexion.commit()
print("Inserción correcta")

try:
    Conexion_1 = pymysql.connect(
        host='0.0.0.0', port=6601, user='proyectodf', passwd='cecad*proyectodf',
        db='CHIP_1', charset='utf8', cursorclass=pymysql.cursors.DictCursor)
    Conexion_2 = pymysql.connect(
        host='0.0.0.0', port=6602, user='proyectodf', passwd='cecad*proyectodf',
        db='CHIP_2', charset='utf8', cursorclass=pymysql.cursors.DictCursor)
    print("Conexión correcta")
except (pymysql.err.OperationalError, pymysql.err.InternalError) as Error:
    print("Ocurrió un error al conectar: ", Error)

with open('/home/proyectodf/Datos/Datos_1.json') as file:
    Datos_1 = json.load(file)

with open('/home/proyectodf/Datos/Datos_2.json') as file:
    Datos_2 = json.load(file)

Tablas = ["TENSION", "CORRIENTE", "FRECUENCIA", "TEMPERATURA",
"THD_V", "THD_I", "POTENCIA", "FASE"]
Insertar_Datos(Conexion_1, Datos_1, Tablas)
Insertar_Datos(Conexion_2, Datos_2, Tablas)

```

En este script además de utilizar la librería pymysql, se utiliza la librería json que permite utilizar archivos con extensión .json y operarlos dentro del código. La función denominada “Insertar_Datos” tiene como argumentos de entrada, la variable de tipo conexión, la variable que contiene los datos y el arreglo con los nombres de las tablas. Al igual que en el caso anterior, se genera un ciclo for a través de los elementos del arreglo con los nombres de las tablas y se establece un cursor para realizar comandos mysql; como se tiene que utilizar el comando de mysql “INSERT” la estructura del comando es fija para la tabla, así que primero se realizan los condicionales para ajustar a la tabla correspondiente (de 4 o 10 elementos). Cuando ya se tiene la estructura del comando lista para una determinada tabla, se genera un ciclo for que itera a través de los elementos dentro de la variable de los datos, la cual contiene los datos capturados y en formato json convertido a un diccionario de Python. De esta manera se insertan los datos de un ciclo capturado como una fila en cada una de las tablas de la base de datos.

La conexión con las bases de datos se realiza de la misma manera que para el caso de vaciado, por lo que no se repetirá la explicación de este segmento de código. Lo siguiente que se realiza fuera de la función de inserción, es la carga del archivo json y su conversión en variables de tipo diccionario, para cada uno de los archivos json, es decir, para ambos chips del medidor. El arreglo con los nombres de las tablas es el mismo, por lo que no se modifica y finalmente se llama a la función de inserción para cada una de las variables de tipo conexión, es decir, para cada una de las bases de datos.

Caben destacar que después de la inserción de los datos en cada una de las bases de datos, el script de Python del medidor sigue ejecutándose, por lo que se debe borrar las variables que almacenan los datos, esta acción se realiza con el llamado de la función “Borrar_JSON” que restaura estas variables a su estado inicial. Los archivos con extensión .json guardados de forma local, tanto en el medidor como en el servidor, no necesitan ser borrados, pues estos archivos se sobre-escriben cada vez que se crean en la función de transferencia, sin repetir la información.

3.5.3. Limpiar datos de las tablas de las bases de datos

De la misma forma que son ejecutadas las funciones de transferencia e inserción, la función de limpieza se ejecuta cuando se cumple un tiempo determinado, siendo este tiempo mucho mayor al tiempo de subida de los datos. Cada que se cumpla este tiempo, se ejecuta la función de limpieza, sin embargo, si es la primera vez que se va a realizar la limpieza, se tiene la condición de que hayan suficientes datos antes de limpiar. Si se quiere hacer una limpieza cada cierto periodo de tiempo, en las bases de datos tienen que haber mínimo el doble de tiempo en datos. Por ejemplo, se desea hacer la limpieza cada día, en las bases de datos debe existir mínimo dos días de datos. En este caso se eliminan los datos del día más antiguo; de esta manera se asegura que haya mínimo un periodo completo de datos capturados en las tablas. La función que permite realizar estas acciones se muestra a continuación:

```
def Limpiar_DB(IP, Puerto, Usuario, Contramarca):
    Comando_Limpiar = pexpect.spawn(
        "ssh " + Usuario + "@" + IP + " -p" + Puerto +
        " cd Funciones; python3 Limpiar_sistemas.py")
    Reaccion_Limpiar = Comando_Limpiar.expect([
        [Usuario + "@" + IP + "'s password: ", pexpect.EOF]])
    if (Reaccion_Limpiar == 0):
        Comando_Limpiar.sendline(Contramarca)
        Comando_Limpiar.expect(pexpect.EOF)
    else:
        print("Error de autenticación de credenciales")
```

Al igual que la función de vaciado o la de inserción, se ejecuta el comando para ejecutar el script en el servidor, se utilizan las credenciales para la autenticación y se espera la solicitud de la contraseña, si es la correcta, se inyecta la contraseña y se ejecuta el script de Python denominado “Limpiar_sistemas.py”. Este script ejecuta las acciones para comunicarse con las bases de datos y elimina la mitad de los datos que hayan en las tablas. El código de este script se muestra a continuación:

```
import pymysql

def Limpiar_DB(Conexion, Tablas):
    for Tabla in range(len(Tablas)):
        with Conexion.cursor() as Cursor:
            Consulta_Max = "SELECT MAX(CICLO) AS CICLO FROM " + Tablas[Tabla] + ";"
            Consulta_Min = "SELECT MIN(CICLO) AS CICLO FROM " + Tablas[Tabla] + ";"
            Cursor.execute(Consulta_Max)
            Ultimo_Registro = Cursor.fetchall()
            Ultimo = list(); Primer = list();
            for Cifra in Ultimo_Registro[0].values():
                Ultimo.append(Cifra)
            Cursor.execute(Consulta_Min)
```

```

Primer_Registro = Cursor.fetchall()
for Cifra in Primer_Registro[0].values():
    Primer.append(Cifra)
Limite = int((Ultimo[0]-Primer[0])/2) + Primer[0]
Limpiear = "DELETE FROM " + Tablas[Tabla] + " WHERE CICLO <= %s;"
Cursor.execute(Limpiear, Limite)
Conexion.commit()
print("Limpieza correcta")

try:
    Conexion_1 = pymysql.connect(
        host='0.0.0.0', port=6601, user='proyectodf', passwd='cecad*proyectodf',
        db='CHIP_1', charset='utf8', cursorclass=pymysql.cursors.DictCursor)
    Conexion_2 = pymysql.connect(
        host='0.0.0.0', port=6602, user='proyectodf', passwd='cecad*proyectodf',
        db='CHIP_2', charset='utf8', cursorclass=pymysql.cursors.DictCursor)
    print("Conexión correcta")
except (pymysql.err.OperationalError, pymysql.err.InternalError) as Error:
    print("Ocurrió un error al conectar: ", Error)

Tablas = ["TENSION", "CORRIENTE", "FRECUENCIA", "TEMPERATURA",
          "THD_V", "THD_I", "POTENCIA", "FASE"]
Limpiear_DB(Conexion_1, Tablas)
Limpiear_DB(Conexion_2, Tablas)

```

Se puede evidenciar en este script que se importa únicamente la librería pymysql al igual que en el script de vaciado y de la misma forma, la función de limpieza tiene como argumentos de entrada la variable que realiza la conexión y el arreglo con los nombres de la tablas. Inicialmente se genera un ciclo for a través del arreglo de las tablas y se establece la conexión y el cursor para ejecutar los comandos. Lo siguiente es generar dos consultas y obtener sus resultados; estas consultas se realizan para obtener el primer y el último registro de la tabla, para así poder extraer la mediana de los registros. Cuando se tiene este valor se eliminan los registros menores a este valor de la mediana, dejando únicamente la mitad más reciente de los datos.

Al igual que con el resto de los scripts, se establece la conexión para un determinado contenedor y una determinada base de datos. La variable con los nombres de las tablas se mantiene intacta y luego se llaman a las funciones de limpieza para su respectiva conexión, es decir, para la limpieza de cada base de datos. Los scripts mencionados se encuentran almacenados en la carpeta “Funciones” dentro del servidor, cada medidor cuenta con sus respectivos scripts. Estos códigos son los encargados de subir y modificar correctamente la información capturada por el medidor y almacenarlos de una manera organizada dentro de las bases relationales en MySQL dentro del servidor.

3.5.4. Tratamiento de las fallas críticas capturadas

A diferencia de las fallas leves que pueden ocurrir en frecuencia o incluso en tensión, las fallas críticas en corriente y tensión afectan directamente el sistema eléctrico, por lo tanto, estas fallas deben recibir un tratamiento distinto al de las fallas leves. Las fallas críticas se deben tratar de manera análoga al resto de los parámetros capturados por el medidor. Es por esta razón que estas fallas tienen una base de datos y un espacio de almacenamiento distinto al de los demás parámetros del sistema.

Como se menciona en el capítulo anterior, cuando la información de tensión y corriente es capturada por el medidor, se procede a evaluar el sistema de forma trifásica para detectar el estado de operación y validar si existe una falla crítica. En este caso, si el sistema detecta un estado de operación distinto al estado normal, se le considera una falla crítica; la cual puede ser alguna de las fallas de baja impedancia mostradas en la figura 3.5.

Una vez detectada una falla crítica, se debe manejar dicha información con bastante agilidad, es por esta razón que se debe ejecutar una función para transferir dichos datos hacia el servidor, para luego ser almacenados en una base de datos independiente y poder ser visualizados por el usuario en el aplicativo web. En el código mostrado a continuación se evidencia como se crea el archivo JSON con las características de las fallas y a su vez como se transfiere hasta el servidor. Este proceso se realiza de forma similar a las funciones mostradas anteriormente, validando las credenciales de usuario y utilizando el protocolo SCP. Dentro del servidor se ejecuta el script “Insertar_Fallas.py” que organiza la información recibida en la respectiva tabla de la base de datos. finalmente, se ejecuta el comando para limpiar el diccionario de Python y así evitar redundancia en la información.

```
def Transferencia_Fallas(Chip, Estado, IP, Puerto, Usuario, Contramarca, Carpeta):
    medidor = "Sistemas"
    Tiempo_Falla = datetime.datetime.today()
    json_fallas = {}
    json_fallas["Medidor"] = medidor; json_fallas["Chip"] = Chip;
    json_fallas["Falla"] = Estado; json_fallas["Tiempo"] = Tiempo_Falla
    with open("/root/" + Carpeta + "/Datos_Fallas.json", "w") as file:
        json.dump(json_fallas, file, indent=4, ensure_ascii=False)

    Comando_Transferencia = pexpect.spawn("scp -P " + Puerto + " /root/" + Carpeta +
                                          "/Datos_Fallas.json " + Usuario + "@" + IP + ":" + Ruta + Carpeta + "/")
    Reaccion_Transferencia = Comando_Transferencia.expect([Usuario + "@" + IP +
    "'s password: ", pexpect.EOF])
    if (Reaccion_Transferencia == 0):
        Comando_Transferencia.sendline(Contramarca)
        Comando_Transferencia.expect(pexpect.EOF)
    else:
        print("Error de autenticación de credenciales")

    Comando_Insertar = pexpect.spawn("ssh " + Usuario + "@" + IP + " -p" + Puerto +
                                     " cd Funciones; python3 Insertar_Fallas.py")
    Reaccion_Insertar = Comando_Insertar.expect([Usuario + "@" + IP +
    "'s password: ", pexpect.EOF])
    if (Reaccion_Insertar == 0):
        Comando_Insertar.sendline(Contramarca)
        Comando_Insertar.expect(pexpect.EOF)
    else:
        print("Error de autenticación de credenciales")
    json_fallas = {}
```

El script que se ejecuta en el servidor, organiza la información en una tabla, de la misma manera que las funciones de limpieza, vaciado e inserción, únicamente cambia el nombre de la base de datos y el puerto de conexión. Una vez registrada la información de la falla en la base de datos, inmediatamente se envía una señal de activación hacia el relé del sistema, para desactivar el flujo de corriente, esto con el fin de evitar errores mayores en la red y en los equipos. El comando que realiza esta acción se muestra a continuación.

```
Comando = pexpect.run("mosquitto_sub -h 192.168.0.10 -t /path/relay/set")
```

Capítulo 5

Servidor web

En este capítulo se pretende realizar una explicación detallada acerca de cual fue el procedimiento utilizado para la implementación del servidor web, dentro del espacio designado para el proyecto en el servidor de GITUD.

1. Características técnicas del servidor web

En esta sección se presentan las características técnicas del servidor web proporcionadas para el proyecto, el servidor web cuenta con las siguientes características internas:

- CPUs: 2
- RAM: 4 GB
- Storage: 40 GB

Donde la información de uso y total de la memoria RAM y el espacio de almacenamiento, se pueden obtener mediante los siguientes comandos respectivamente:

```
free -h  
df -h
```

2. Configuración servidor web - APACHE

El primer paso como cualquier otro proceso de instalación, es realizar una actualización de la lista de paquetes en el gestor de librerías y aplicaciones. Seguidamente, se debe realizar la instalación de “Apache”. Este procedimiento se realiza con los siguientes comandos:

```
sudo apt update  
sudo apt install apache2
```

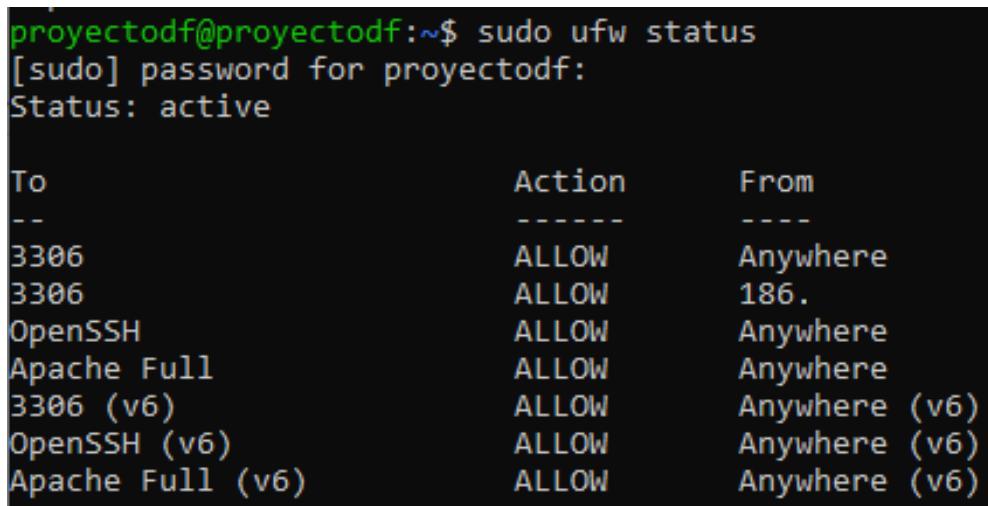
Una vez finalizada la instalación de Apache, se debe realizar una configuración al firewall, para proporcionar seguridad al momento de realizar la instalación del servidor web. Para esto, lo primero que se debe realizar es agregar una regla de excepción a la conexión SSH existente, mediante la cual, se realiza la comunicación al servidor, esta acción se realiza mediante el siguiente comando:

```
sudo ufw allow OpenSSH
```

Ahora, se deben aplicar las excepciones de la configuración del firewall para Apache y la habilitación de las mismas, de la siguiente manera:

```
sudo ufw allow in "Apache Full"  
sudo ufw enable
```

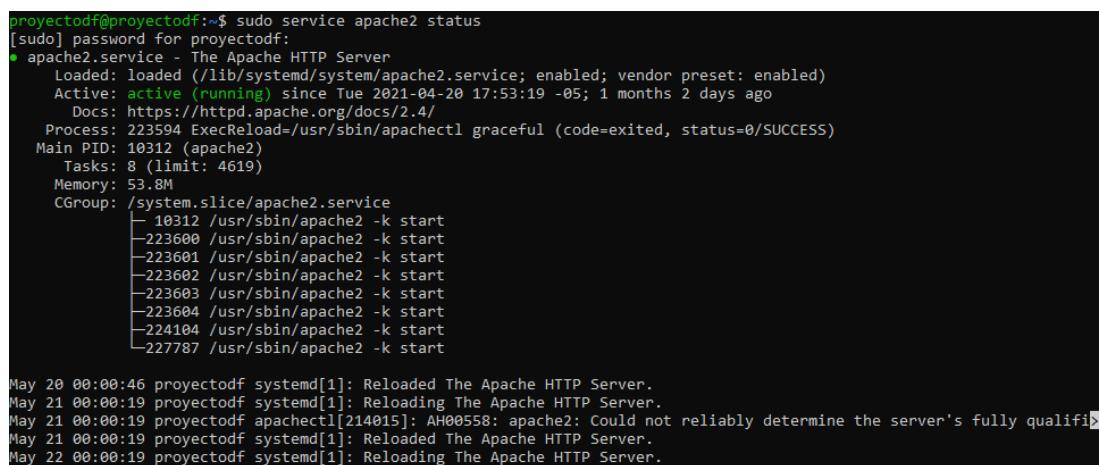
Una vez ejecutados los comandos de configuración, se debe realizar la comprobación del estado del firewall, en donde se debe mostrar como estado ‘Allow’ (‘permitido’). Mediante el comando de ‘status’ del firewall se puede realizar la comprobación. Como resultado en el servidor web se obtiene la información del funcionamiento del firewall. La figura 5.1 muestra este resultado.



```
proyectodf@proyectodf:~$ sudo ufw status  
[sudo] password for proyectodf:  
Status: active  
  
To                         Action      From  
--                         --          --  
3306                        ALLOW       Anywhere  
3306                        ALLOW       186.  
OpenSSH                      ALLOW       Anywhere  
Apache Full                  ALLOW       Anywhere  
3306 (v6)                   ALLOW       Anywhere (v6)  
OpenSSH (v6)                 ALLOW       Anywhere (v6)  
Apache Full (v6)             ALLOW       Anywhere (v6)
```

Figura 5.1: Información del ‘status’ del firewall configurado para el servidor web.

Finalmente, se comprueba el estado del servidor web, el cual debe estar activo y no debe presentar error alguno, con el fin de que permita la correcta visualización de la interfaz web. En la figura 5.2 se evidencia la respuesta generada por parte del servidor, para la solicitud de estado realizada previamente.



```
proyectodf@proyectodf:~$ sudo service apache2 status  
[sudo] password for proyectodf:  
● apache2.service - The Apache HTTP Server  
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
  Active: active (running) since Tue 2021-04-20 17:53:19 -05; 1 months 2 days ago  
    Docs: https://httpd.apache.org/docs/2.4/  
  Process: 223594 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)  
 Main PID: 10312 (apache2)  
   Tasks: 8 (limit: 4619)  
  Memory: 53.8M  
 CGroup: /system.slice/apache2.service  
         ├─ 10312 /usr/sbin/apache2 -k start  
         ├─223600 /usr/sbin/apache2 -k start  
         ├─223601 /usr/sbin/apache2 -k start  
         ├─223602 /usr/sbin/apache2 -k start  
         ├─223603 /usr/sbin/apache2 -k start  
         ├─223604 /usr/sbin/apache2 -k start  
         ├─224104 /usr/sbin/apache2 -k start  
         └─227787 /usr/sbin/apache2 -k start  
  
May 20 00:00:46 proyectodf systemd[1]: Reloaded The Apache HTTP Server.  
May 21 00:00:19 proyectodf systemd[1]: Reloading The Apache HTTP Server.  
May 21 00:00:19 proyectodf apachectl[214015]: AH00558: apache2: Could not reliably determine the server's fully qualified name, using 127.0.0.1 for Port 80.  
May 21 00:00:19 proyectodf systemd[1]: Reloaded The Apache HTTP Server.  
May 22 00:00:19 proyectodf systemd[1]: Reloading The Apache HTTP Server.
```

Figura 5.2: Respuesta de confirmación del estado del servidor web Apache.

3. MySQL

La base de datos utilizada para el proyecto es MySQL, se selecciona debido a que es una base de datos relacional, la cual, beneficia el almacenamiento de los datos debido a que cada variable de la base de datos se relaciona entre sí, además, presenta una estructura en tablas que permite operar los datos de manera organizada y selectiva mediante el parámetro ID.

Para la instalación de MySQL, se realiza el update del gestor de librerías y se ejecuta el comando de instalación, tal cual se muestra a continuación:

```
sudo apt update  
sudo apt install mysql-server
```

Donde una vez finalizada la instalación se debe comprobar el estado de la base de datos mediante el comando:

```
sudo service mysql status
```

La figura 5.3 muestra el estado de la base de datos instalada.

```
projectodf@projectodf:~$ sudo service mysql status  
[sudo] password for projectodf:  
● mysql.service - MySQL Community Server  
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)  
  Active: active (running) since Thu 2021-05-13 06:46:30 -05; 1 weeks 3 days ago  
    Main PID: 149768 (mysqld)  
      Status: "Server is operational"  
        Tasks: 39 (limit: 4619)  
       Memory: 372.8M  
         CGroupl: /system.slice/mysql.service  
                   └─149768 /usr/sbin/mysqld  
  
May 13 06:46:28 projectodf systemd[1]: Starting MySQL Community Server...  
May 13 06:46:30 projectodf systemd[1]: Started MySQL Community Server.
```

Figura 5.3: Respuesta de confirmación del estado de la base de datos MySQL.

3.1. Configuración adicional de MySQL

Una vez instalado MySQL se deben realizar configuraciones de seguridad adicionales, esto con el fin de que las bases de datos no sean accedidas por terceros. Para esto, se debe realizar la instalación del script de shell ‘mysql_secure_installation’, para lo cual, se ejecuta el siguiente comando:

```
sudo mysql_secure_installation
```

Durante el proceso de ejecución del script, se deberá ingresar la contraseña de usuario root a utilizar como superusuario de MySQL, para lo cual se asigna la contraseña ‘cecad*projectodf’ como predeterminada para el usuario root de MySQL. Seguidamente, se solicitará la eliminación de usuarios anónimos como medida de seguridad, dado que un

usuario anónimo es utilizado en el proceso de instalación por MySQL y este permite acceder a cualquier persona a una sesión de MySQL, sin tener creada una cuenta de usuario.

Normalmente, al superusuari ‘root’ solo se le debe permitir conectarse desde ‘localhost’. Deshabilitar esta opción ayuda a evitar que un tercero adivine la contraseña de superusuari ‘root’ del servidor web. Finalmente, se debe acceder a MySQL para probar la conexión y verificar que la instalación fue configurada correctamente. Dicha conexión se realiza mediante el siguiente comando:

```
mysql -h localhost -u root -p
```

La figura 5.4 muestra el resultado del acceso a MySQL mediante el usuario ‘root’.

```
proyectodf@proyectodf:~$ mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 339
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figura 5.4: Respuesta de confirmación del estado de la base de datos MySQL.

4. PHP

En esta sección se explica la configuración requerida para el lenguaje de back-end seleccionado PHP, el cual permite la comunicación entre el servidor web y la interfaz gráfica. Por otra parte, el enlace con la front-end se implementa en los lenguajes HTML, CSS, JavaScript. Para la correcta ejecución de código PHP, se debe instalar un paquete en el servidor, de la siguiente manera:

```
sudo apt update
sudo apt install php libapache2-mod-php php-mysql
```

Al momento de realizar la instalación de PHP, se debe comprobar que la versión haya sido instalada correctamente mediante el siguiente comando “php -version”. La figura 5.5 muestra la comprobación de la instalación del paquete de PHP.

```
proyectodf@proyectodf:~$ php -version
PHP 7.4.3 (cli) (built: Oct  6 2020 15:47:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
```

Figura 5.5: Comprobación de versión de PHP.

5. PhpMyAdmin

Como en las secciones anteriores, se debe realizar la actualización del gestor de paquetes y la instalación de PhpMyAdmin, incluyendo la extensión ‘php-mbstring’, ya que esta extensión mejora el rendimiento de la aplicación. Los siguientes comandos realizan el procedimiento descrito anteriormente:

```
sudo apt update  
sudo apt install phpmyadmin php-mbstring
```

5.1. Proceso de instalación PhpMyAdmin

Una vez ejecutado el proceso de preparación de paquetes de la instalación, se desplegará una interfaz emergente, en donde los pasos de configuración se mostrarán a continuación:

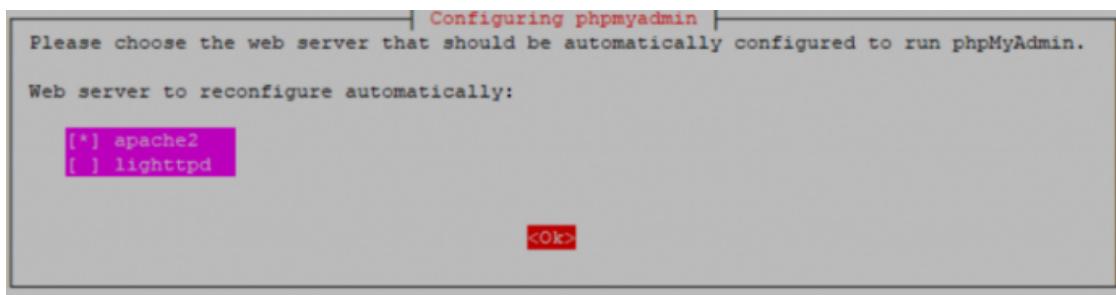


Figura 5.6: Selección de Apache2 como servidor web.

En la figura 5.6 se muestra el proceso de selección del servidor web, que en este caso será el ya configurado ‘Apache’, para lo cual simplemente se debe seleccionar y continuar.

Una vez seleccionado el servidor web, se debe establecer la base de datos perteneciente a la función de ingreso en PhpMyAdmin, para lo cual, la instalación preguntará si se desea crear dicha base de datos como ‘dbconfig-common’; en esta opción simplemente debe presionarse la opción ‘Yes’. La figura 5.7 muestra la visualización de esta opción.

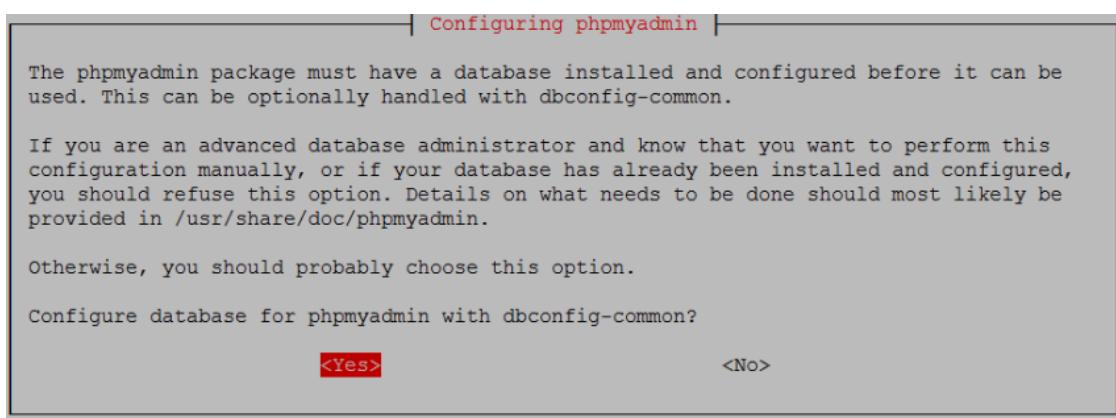


Figura 5.7: Comprobación de versión de PHP.

Como última instancia en esta configuración, se debe establecer una contraseña de aplicación entre MySQL y PhpMyAdmin, la cual será utilizada internamente para comunicación entre estos dos servicios. No se recomienda establecer una contraseña en específico, dado que al dejar en blanco dicha entrada, el proceso de instalación generará una clave aleatoria, por lo cual simplemente se debe seleccionar la opción ‘Ok’ y esperar que finalice el proceso. La figura 5.8 muestra la visualización del proceso descrito anteriormente.

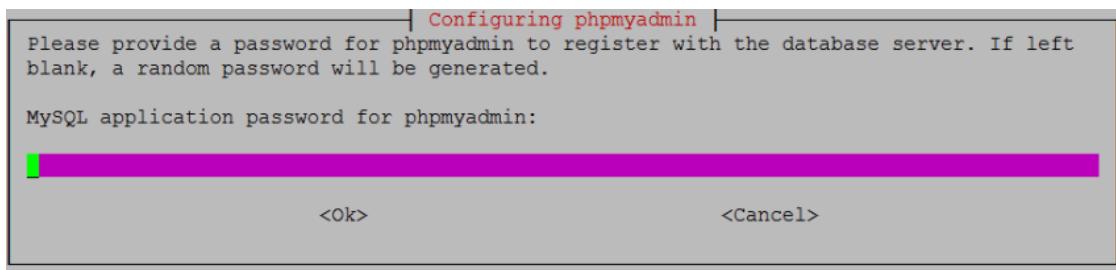


Figura 5.8: Solicitud de contraseña de aplicación entre MySQL y PhpMyAdmin.



Figura 5.9: Interfaz de PhpMyAdmin.

Finalmente, como en las secciones anteriores, se debe realizar una comprobación del servicio, esta vez se debe acceder al servidor mediante la dirección IP y el puerto HTTP del servidor web. Para este caso, se debe añadir el siguiente enlace desde un navegador web.

<http://200.69.103.29:28556/phpmyadmin/index.php>

Una vez se accede a esta URL, se debe desplegar la interfaz de PhpMyAdmin descrita en la figura 5.9 que comprueba que el proceso de instalación y configuración se realizó correctamente.

6. Bases de datos

En esta sección se pretende realizar una explicación detallada acerca de las bases de datos implementadas en el servidor. La base de datos implementada para el servidor web y para el almacenamiento de los parámetros eléctricos obtenidos por el medidor, es MySQL, dado que es una base de datos relacional, la cual como su nombre lo dice, permite relacionar en tablas discriminadas por un ‘ID’, valores que correspondan a un mismo factor en común. El ‘ID’ es un valor numérico que establece el orden de los parámetros, es decir, si un valor es tomado en primera posición, este se le asignará un valor de ‘ID’ igual a 1, seguidamente a la segunda toma de medidas se le asignará el valor de ‘ID’ igual a 2, esto con el fin de facilitar consultas y ordenar los valores obtenidos por el medidor.

Para el desarrollo de este proyecto, se analizó que podrían existir consultas simultáneas de diferentes puntos de medición en tiempo real, es decir, un usuario ‘A’ podría solicitar información de la sala de sistemas, a través de la página web y al mismo tiempo, un usuario ‘B’ podría solicitar información de la misma manera, pero en este caso de la red eléctrica de laboratorios. Para brindar una solución a este problema, se decide utilizar una tecnología que permita independizar las bases de datos y además permita realizar la comunicación a estas desde distintos puertos TCP.

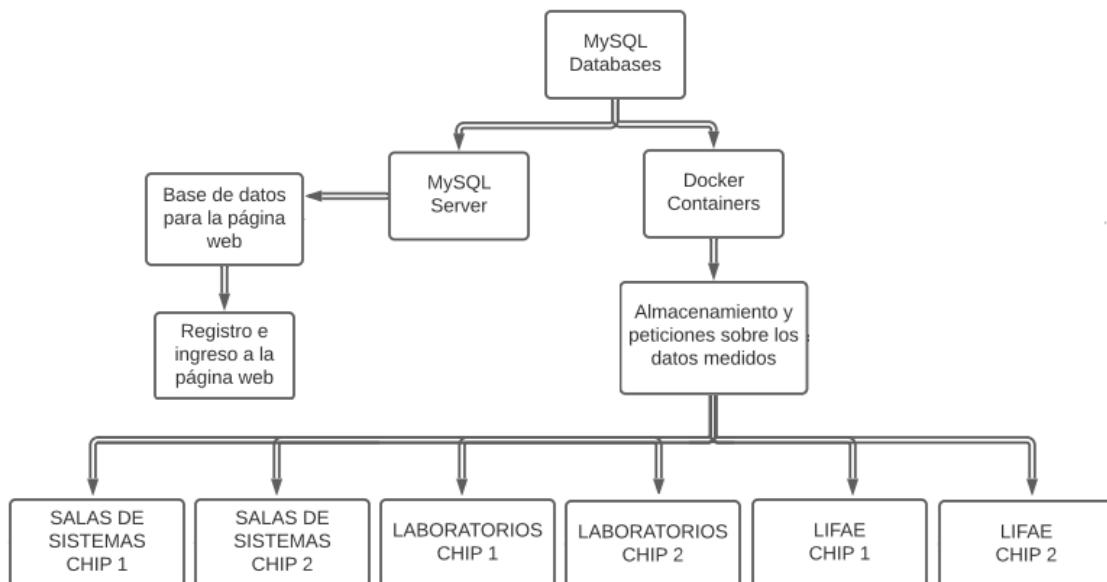


Figura 5.10: Diagrama de la distribución de las bases de datos implementadas en el servidor web.

Se selecciona *Docker*, ya que permite crear contenedores virtuales con una imagen de MySQL, que son capaces de crear bases de datos independientes para cada chip y para

cada sala de medición, incluso independientes de la base de datos de usuarios designada para la página web (Ingreso y registro); además, se utiliza la misma versión de MySQL implementada en el servidor web para tener uniformidad en dichas bases de datos. La figura 5.10 muestra la distribución de las bases de datos utilizadas en el proyecto.

Los puertos TCP utilizados para las bases de datos se presentan a continuación:

Puerto TCP para las bases de datos			
Base de datos de usuarios		Localhost (3306)	
Bases de datos	Base de datos de las salas de sistemas	Base de datos de las salas de laboratorios	Base de datos del panel del grupo LIAFE
Chip 2	6601	6603	6605
Chip 2	6602	6604	6606

Tabla 5.1: Distribución de los puertos TCP para la comunicación con las bases de datos.

En la tabla 5.1 se evidencian los puertos TCP designados para cada base de datos, los cuales permiten la comunicación de la interfaz y del medidor con la base de datos en el servidor.

6.1. Base de datos para los usuarios

En este apartado, se pretende realizar la descripción de la base de datos implementada para el almacenamiento y petición de datos de los usuarios registrados en la página web.

Field	Type	Null	Key	Default	Extra
<i>id</i>	int unsigned	NO	PRI	NULL	auto_increment
<i>apellidos</i>	varchar(50)	NO		NULL	
<i>nombres</i>	varchar(50)	NO		NULL	
<i>usuario</i>	varchar(50)	NO		NULL	
<i>email</i>	varchar(50)	NO		NULL	
<i>password</i>	varchar(50)	NO		NULL	
<i>ocupacion</i>	varchar(30)	NO		NULL	
<i>pregunta_seguridad</i>	varchar(50)	NO		NULL	
<i>respuesta</i>	varchar(30)	NO		NULL	

Tabla 5.2: Estructura de la tabla creada para el almacenamiento de los datos de los usuarios.

En la tabla 5.2 se muestra la estructura de la tabla de MySQL creada para almacenar los datos de los usuarios. Cada campo mostrado en la tabla es un dato ingresado por el usuario para su respectivo registro y consulta. Los valores que se guardan en cada campo se ingresan como una cadena de texto para la facilidad de las consultas, durante el inicio de la sesión.

6.2. Base de datos para los datos capturados

Como se menciona anteriormente, en cada contenedor se almacena la base de datos para un chip de cada medidor. Las tablas son exactamente iguales para cada contenedor, pues cada chip maneja exactamente los mismos parámetros.

Field	Type	Null	Key	Default	Extra
<i>CICLO</i>	int unsigned	NO	PRI	NULL	auto_increment
<i>LINEA_A</i>	varchar(30)	NO		NULL	
<i>TIEMPO_A</i>	varchar(50)	NO		NULL	
<i>ESTADO_A</i>	varchar(50)	NO		NULL	
<i>LINEA_B</i>	varchar(30)	NO		NULL	
<i>TIEMPO_B</i>	varchar(50)	NO		NULL	
<i>ESTADO_B</i>	varchar(50)	NO		NULL	
<i>LINEA_C</i>	varchar(30)	NO		NULL	
<i>TIEMPO_C</i>	varchar(50)	NO		NULL	
<i>ESTADO_C</i>	varchar(50)	NO		NULL	

Tabla 5.3: Estructura de la tabla creada para el almacenamiento de los datos capturados para una variable dependiente de las tres líneas.

En la tabla 5.3 se muestra la estructura de la tabla MySQL empleada para almacenar los datos capturados para una variable que posee tres líneas, tales variables son: Potencia, tensión, corriente, THDv, THDi y fase. Los campos equivalen a los valores capturados para esa variable en un ciclo de captura. Como se dijo en capítulos anteriores, estos campos almacenan los valores capturados, el tiempo de captura de cada uno de ellos y el estado instantáneo en el que se encontraban al momento de la captura, para cada una de las líneas.

Field	Type	Null	Key	Default	Extra
<i>CICLO</i>	int unsigned	NO	PRI	NULL	auto_increment
<i>VALOR</i>	varchar(30)	NO		NULL	
<i>TIEMPO</i>	varchar(50)	NO		NULL	
<i>ESTADO</i>	varchar(50)	NO		NULL	

Tabla 5.4: Estructura de la tabla creada para el almacenamiento de los datos capturados para una variable independiente de las líneas.

En la tabla 5.4 se muestra la estructura de la tabla MySQL empleada para almacenar los datos capturados para una variable que no depende de las líneas, como es el caso de la temperatura o de la frecuencia. De forma similar a la tabla 5.3 los campos equivalen a los valores capturados para esa variable en un ciclo de captura, en este caso se almacenan únicamente el valor de la variable, el tiempo de captura y el estado instantáneo de la variable en el momento de la captura.

7. Habilitación HTTPs

Para la configuración de soporte HTTPs para el sitio web, se realiza una configuración adicional de Apache, con el fin de generar compatibilidad y soporte al momento de iniciar una sesión de flujo de datos cifrados. En primera instancia, se debe configurar el servicio SSL dentro del servidor web, para lo cual se realiza la instalación de las librerías que lo soportan mediante los siguientes comandos:

```
sudo apt-get update  
sudo apt-get install openssl
```

Una vez configurada la librería de SSL a utilizar, se habilitan los módulos Mod_SSL y Mod_rewrite, con el fin de que se permita la creación de certificados digitales SSL, los cuales se utilizarán para el inicio de la sesión de cifrado:

```
a2enmod ssl  
a2enmod rewrite
```

Una vez activados estos módulos, se deberá habilitar en la ruta de la aplicación web la sobre-escritura desde el usuario root, para lo cual, se edita el archivo de configuración de apache ubicado en la siguiente ruta:

```
vi /etc/apache2/apache2.conf
```

Una vez dentro del archivo, se añade la siguiente sentencia al final del archivo:

```
<Directory /var/www/html>  
    AllowOverride All  
</Directory>
```

Una vez configurado Apache, se procede a realizar la creación y carga del certificado de raíz SSL.

7.1. Certificado SSL

Con el fin de realizar una verificación satisfactoria con respecto al Handshake del protocolo TLS, se realiza la creación y carga del certificado SSL, para lo cual se utilizará la herramienta openSSL para la creación de este certificado, mediante los siguientes comandos, los cuales apuntarán a la ruta de almacenamiento del certificado en apache y lo crearán.

```
mkdir /etc/apache2/certificate  
cd /etc/apache2/certificate  
  
openssl req -new -newkey rsa:4096 -x509 -sha256 -days...  
365 -nodes -out apache-certificate.crt -keyout apache.key
```

Finalmente, se realiza la instalación de dicho certificado en la configuración del sitio, para lo cual se accede al archivo de configuración que despliega las rutas de administración de la aplicación:

```
<VirtualHost *:80>
    ServerAdmin root@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on
    SSLCertificateFile /etc/apache2/certificate/cer.crt
    SSLCertificateKeyFile /etc/apache2/certificate/key.key
</VirtualHost>
```

Y se reinicia el servidor para aplicar los cambios mediante el comando:

```
service apache2 restart
```

Una vez recargado de nuevo el servidor, el protocolo HTTPs quedará correctamente configurado con el certificado SSL de raíz creado.

Capítulo 6

Aplicativo web

La interfaz gráfica es una parte bastante importante del proyecto, ya que es el elemento que permite juntar y visualizar todo lo hecho hasta el momento. Si bien la captura y almacenamiento de los datos puede brindar bastante información al usuario, una imagen vale más que mil palabras; por lo tanto, los datos deben visualizarse, para hacer que llegue el mensaje al usuario de una forma más sencilla y concreta.

En este capítulo se mostrarán todos los elementos que forman parte del aplicativo web, desde el inicio cuando el usuario desea registrarse o iniciar sesión, hasta cuando decide salirse de la interfaz. Cabe resaltar que tal y como se mencionó en capítulos anteriores, la interfaz presenta diferentes privilegios para los tres tipos de usuarios que puedan presentarse: Estudiantes, docentes y supervisores. El siguiente link corresponde al enlace del aplicativo web:

<https://gitud.udistrital.edu.co:8080/>

1. Página de inicio de sesión

La página de inicio de sesión, corresponde a la página de inicio de la interfaz, misma que redirige el enlace mostrado anteriormente. En esta página es donde el usuario ingresa a su cuenta si ya posee una o puede dirigirse a la página de registro presionando el botón correspondiente. También puede dirigirse a una página auxiliar en caso de que el usuario no recuerde su contraseña y desee reemplazarla por una nueva contraseña.

En la figura 6.1 se muestra la vista de escritorio de la página de inicio del aplicativo web, también se evidencian los elementos clave de la página de inicio, los cuales son: Los cuadros de texto que permiten que el usuario ingrese sus credenciales de acceso, los botones de inicio de sesión y de crear cuenta nueva, el hipervínculo de inicio que redirige a la página de inicio o el hipervínculo que redirige a la página de modificación de la contraseña.



Figura 6.1: Captura de la vista de escritorio de la página de inicio de la interfaz.

En la página de inicio pueden darse tres diferentes casos:

- No tener una cuenta de usuario. En este caso el usuario debería presionar el botón verde “Crear cuenta nueva” para dirigirse a la página de registro y poder crear su cuenta exitosamente.
- Ingresar exitosamente a su cuenta utilizando sus credenciales de acceso sin ningún problema.
- Que no se reconozcan sus credenciales de usuario o que sus datos sean incorrectos. En este caso la página mostrará un mensaje de alerta como el que se muestra en la figura 6.2 que le pide al usuario que confirme e ingrese sus datos nuevamente. Si este caso se repite, el usuario debería dirigirse a la página que le permite modificar la contraseña haciendo clic sobre el hipervínculo “¿Has olvidado tu contraseña?”.

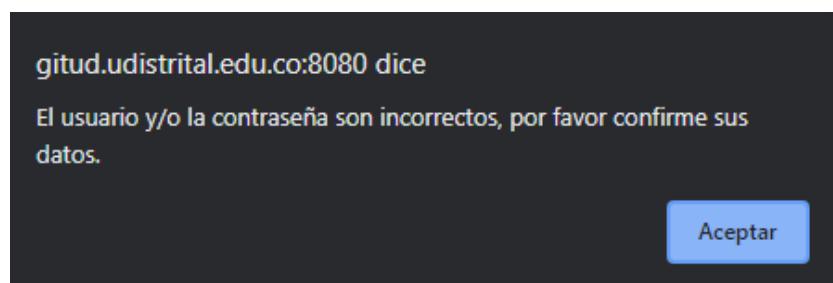


Figura 6.2: Mensaje de alerta por parte de la página de inicio cuando se ingresan las credenciales de acceso incorrectas.

En caso de que el ingreso del usuario con sus credenciales de acceso sea exitoso, la página se redirigirá automáticamente a una de las páginas principales donde se presentan todas las gráficas de los datos capturados. Esta página de las gráficas depende directamente de la ocupación asignada por el usuario al momento del registro, esto se realiza con el fin de determinar los privilegios que se le otorgan al usuario.

2. Página de registro de usuarios

La página de registro no es la principal del aplicativo web, sin embargo, es la página más importante, ya que es la página que permite crear los usuarios y activar las credenciales de acceso; sin esta parte no se puede acceder correctamente a la página principal, por lo que no podrían visualizar las gráficas de los datos capturados. En la figura 6.3 se evidencia la vista de escritorio de la página de registro y creación de cuentas del aplicativo web.

La imagen muestra una captura de pantalla de un formulario de registro de usuario. El formulario tiene un diseño moderno con un fondo azul oscuro y un efecto de circuito impreso en la parte superior. El título "REGISTRO DE USUARIO" está centrado en la parte superior. El formulario se divide en dos columnas principales:

- APELLIDOS:** Campo de texto para ingresar los apellidos.
- NOMBRES:** Campo de texto para ingresar los nombres.
- USUARIO:** Campo de texto para ingresar el código estudiantil.
- CORREO INSTITUCIONAL:** Campo de texto para ingresar el correo institucional.
- CONTRASEÑA:** Campo de texto para ingresar la contraseña, con un icono para alternar visibilidad.
- CONFIRMAR CONTRASEÑA:** Campo de texto para confirmar la contraseña, con un icono para alternar visibilidad.
- OCCUPACIÓN:** Campo desplegable para seleccionar la ocupación.
- PREGUNTA DE SEGURIDAD:** Campo desplegable para seleccionar la pregunta de seguridad.
- RESPUESTA:** Campo de texto para ingresar la respuesta a la pregunta de seguridad.

En la parte inferior del formulario hay un botón azul con el texto "REGISTRARSE".

Figura 6.3: Captura de la vista de escritorio de la página de registro de la interfaz.

En la figura 6.3 se muestran los elementos claves que componen la página, como por ejemplo el hipervínculo de inicio que se redirige hacia la página de inicio en cualquier momento, el botón de registrarse, el cual permite enviar los datos ingresados y también los campos de texto, los cuales se deben llenar con la respectiva información del usuario. Estos campos reciben los datos del usuario, tales como: Apellidos, nombres, usuario, correo institucional, contraseña, ocupación, pregunta de seguridad y respuesta de esa pregunta.

Para la creación de un nuevo usuario, se deben llenar todos los campos de texto con la respectiva información por parte del usuario, es requerido que todos los campos sean llenados, ya que la base de datos tiene una estructura que almacena todos los datos y no puede almacenar datos faltantes. En caso de presionar el botón de registrarse sin llenar todos los campos, la página muestra un mensaje de alerta como el mostrado en la figura 6.4 que solicita al usuario que complete todos los campos de texto de la página.

Además de llenar todos los campos de texto, se tienen que cumplir ciertos requisitos, por ejemplo, el correo institucional, debe ser un correo válido por la universidad, es decir, el correo debe poseer el dominio “correo.udistrital.edu.co” o el dominio “udistrital.edu.co”, de lo contrario, la página de registro mostrará un mensaje de alerta como el que se muestra en la figura 6.5 que solicita al usuario que ingrese un correo institucional con un dominio válido.

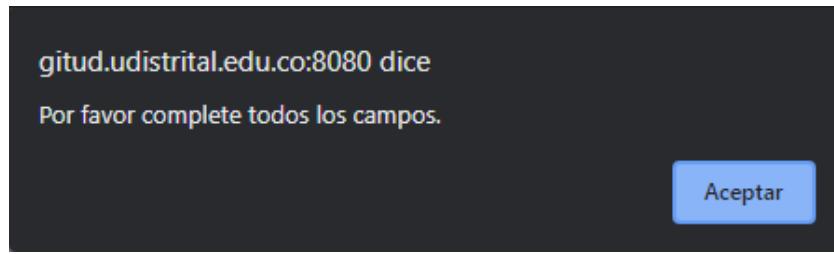


Figura 6.4: Mensaje de alerta por parte de la página de registro cuando se presiona el botón de registrarse sin haber llenado todos los campos de texto.

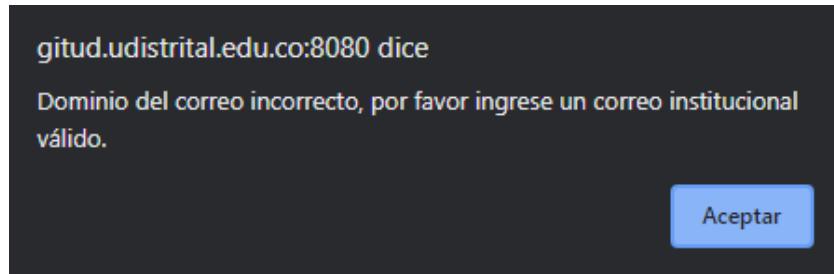


Figura 6.5: Mensaje de alerta por parte de la página de registro cuando se ingresa un correo con un dominio no válido.

Por un tema de seguridad, la página de registro revisa si la contraseña ingresada fue correctamente confirmada, para ello, se emplea el campo de confirmación de contraseña y se proporcionan dos botones que permiten mostrar y ocultar el texto escrito en el campo de contraseña o confirmación de contraseña. Sin embargo, si la página detecta que las contraseñas no son idénticas, se mostrará un mensaje de alerta como el mostrado en la figura 6.6 que solicita al usuario que ingrese y confirme correctamente su contraseña.

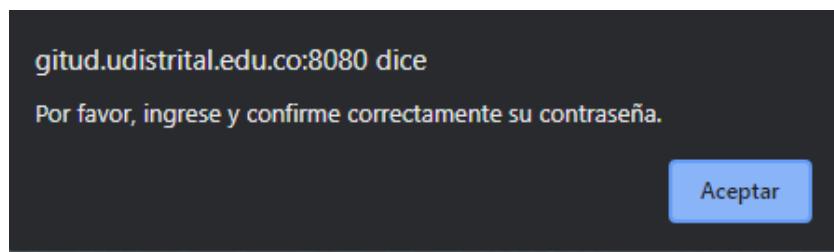
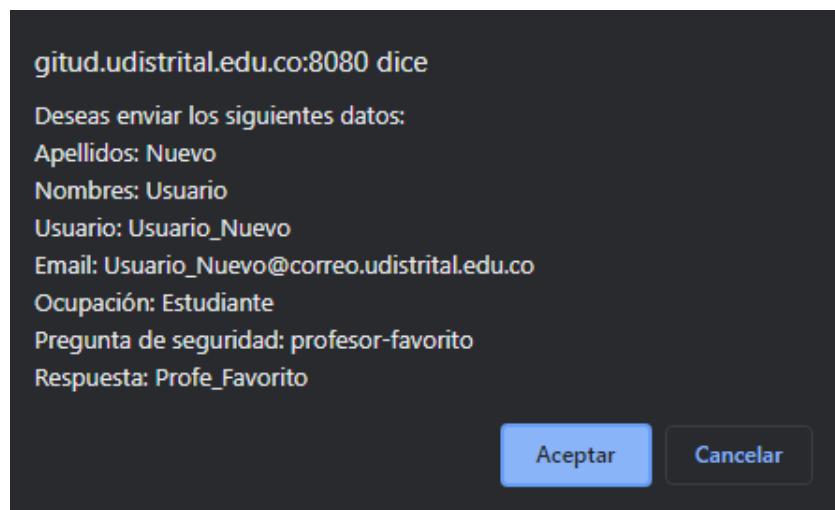


Figura 6.6: Mensaje de alerta por parte de la página de registro cuando se detecta que la contraseña y la confirmación no coinciden.

Una vez se han llenado todos los campos de texto con la información adecuada y correcta, lo siguiente es presionar el botón de registrarse, en este caso la página mostrará un mensaje de confirmación con los datos ingresados, tal y como se muestra en la figura 6.7. El mensaje de confirmación presenta dos opciones, si el usuario no se siente seguro o conforme con los datos ingresados, puede presionar “Cancelar” y luego ingresar todos los datos nuevamente. En caso contrario, si se está seguro de los datos ingresados, el usuario debe presionar “Aceptar” para enviar los datos al servidor.



Por otra parte, si el usuario ya existe en la base de datos, inmediatamente el servidor devuelve una respuesta al usuario indicando que ya existe un usuario registrado con ese nombre. Este respuesta se muestra al usuario a través de un mensaje de alerta en la página de registro, tal y como se muestra en la figura 6.9, en la cual se le pide al usuario que intente nuevamente su proceso de registro.

En algún caso crítico que ocurra un error en el servidor o la base de datos esté presentando errores de almacenamiento, la página de registro mostrará un mensaje de alerta como el mostrado en la figura 6.10, en el cual le indica al usuario que se ha presentado algún tipo de error y que intente nuevamente completar su proceso de registro.

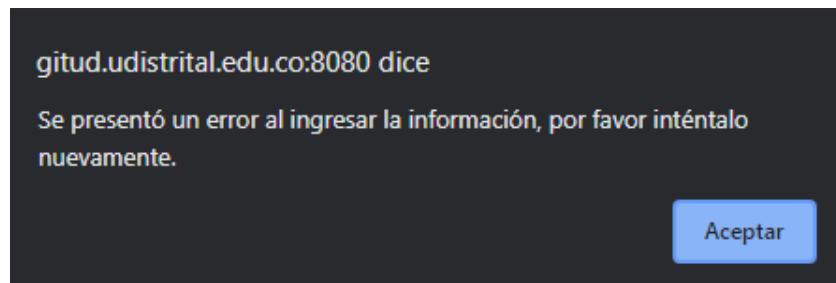


Figura 6.10: Mensaje de alerta por parte de la página de registro cuando se presentan errores de almacenamiento en el servidor o en la base de datos.

3. Página de modificación de contraseña

La página que le permite al usuario modificar su contraseña en caso de que la haya olvidado, es bastante similar a la página de registro mostrada en la figura 6.3, sin embargo, esta página posee varias diferencias en su contenido y su comportamiento. La vista de escritorio de la página de modificación de contraseña se muestra en la figura 6.11.

A screenshot of a web page titled "MODIFICAR CONTRASEÑA" (Change Password). The page has a dark blue background with a hexagonal pattern. It contains several input fields and dropdown menus. At the top right, there is a link "INICIO". The form fields include:

- USUARIO:** A text input field with placeholder "Ingrese su código estudiantil:".
- CORREO INSTITUCIONAL:** A text input field with placeholder "Ingrese su correo institucional:".
- NUEVA CONTRASEÑA:** A text input field with placeholder "Ingrese su contraseña:" and a visibility icon.
- CONFIRMAR CONTRASEÑA:** A text input field with placeholder "Confirme su contraseña:" and a visibility icon.
- OCCUPACIÓN:** A dropdown menu with placeholder "Seleccione su ocupación:".
- PREGUNTA DE SEGURIDAD:** A dropdown menu with placeholder "Seleccione la pregunta de seguridad:".
- RESPUESTA:** A text input field with placeholder "Ingrese su respuesta:".

At the bottom center of the form is a blue button labeled "ACTUALIZAR CONTRASEÑA" (Update Password).

Figura 6.11: Captura de la vista de escritorio de la página de modificación de contraseña de la interfaz.

Como se puede observar en la figura 6.11, la página para modificar la contraseña posee un diseño similar a la página de registro, esto se debe a que en este caso también se debe llenar los campos de texto con la información del usuario. Sin embargo, en este caso, se piden los mismos datos al usuario a excepción de los apellidos, nombres y contraseña, ya que estos datos son innecesarios en este instante.

Los elementos claves de la página, son los mismos que en la página de registro, lo que varía es la cantidad de dichos elementos. En esta página la contraseña que se ingresa será la que quede almacenada en la base de datos en lugar de la anterior; por lo que una vez ingresados los datos correctamente, se debe presionar el botón “Actualizar contraseña” para enviar los datos al servidor y que se actualice la base de datos. Cabe mencionar que los mensajes de alerta mostrados en las figuras 6.4, 6.5, 6.6 y 6.10 también pueden aparecer en esta página si los datos no son ingresados adecuadamente en los campos de texto de la página o en la tabla de la base de datos del servidor. Si los datos fueron ingresados correctamente, se mostrará un mensaje de confirmación similar al de figura 6.7, esta vez con los datos ingresados en esta página, tal y como se muestra en la figura 6.12.

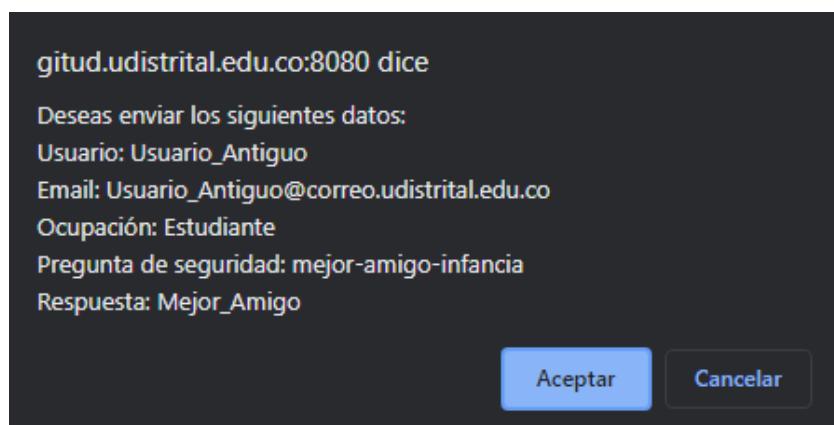


Figura 6.12: Mensaje de confirmación por parte de la página de registro mostrando los datos ingresados.

Una vez confirmados los datos ingresados, estos son enviados al servidor donde se confirma si corresponden con los datos almacenados en la base de datos, dependiendo de este proceso el servidor devuelve una respuesta hacia la página informando al usuario lo sucedido a través de un mensaje de alerta.

Según el proceso realizado en el servidor, se puede obtener alguna de estas respuestas:

- La contraseña se modificó exitosamente. Si este es el caso, la página recibe la respuesta por parte del servidor y muestra un mensaje de alerta como el expuesto en la figura 6.13.
- El usuario ingresado no existe en la base de datos. En este caso, la página muestra un mensaje de alerta como el expuesto en la figura 6.14.
- Los datos ingresados para ese usuario no coinciden con los datos almacenados. En este caso, la página muestra un mensaje de alerta como el expuesto en la figura 6.15.

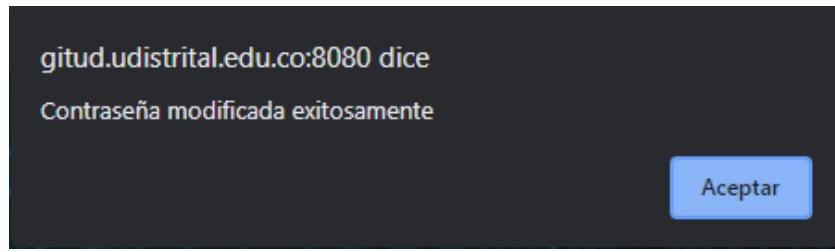


Figura 6.13: Mensaje de alerta por parte de la página de registro cuando se modifica la contraseña del usuario exitosamente.

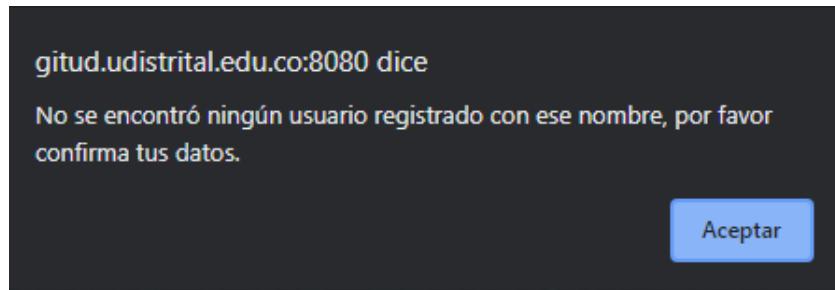


Figura 6.14: Mensaje de alerta por parte de la página de registro cuando el usuario ingresado no existe en la base de datos.

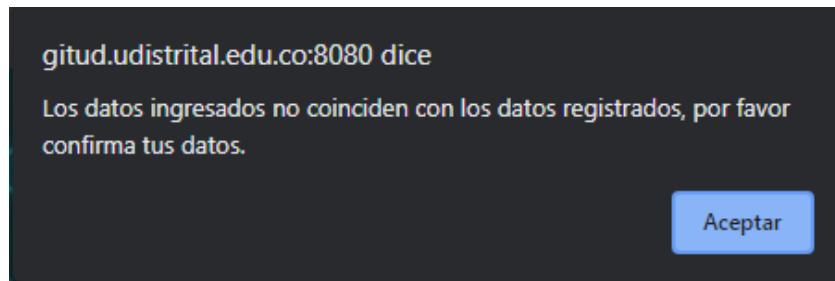


Figura 6.15: Mensaje de alerta por parte de la página de registro cuando los datos ingresados no coinciden con los almacenados en la base de datos.

4. Página principal para estudiantes

Tal y como es mencionado en la primera sección de este capítulo, la página que presenta las gráficas de los datos capturados posee distintos privilegios dependiendo la ocupación de cada usuario (estudiante, docente o supervisor). Estos privilegios son principalmente evidenciados en el diseño y el comportamiento de la página principal. Una vez que un usuario identificado como estudiante ha realizado correctamente el proceso de inicio de sesión utilizando sus credenciales de acceso, la página de inicio se redirige automáticamente a la página principal de estudiantes, la vista de escritorio de esta página se puede evidenciar en la figura 6.16.



Figura 6.16: Captura de la vista de escritorio de la página principal para estudiantes del aplicativo web.

La página principal, independientemente de la ocupación del usuario, se divide en tres secciones principales que son equivalentes a cada uno de los tres medidores. Dentro de cada una de estas secciones, existen dos sub-secciones equivalentes a los dos chips que posee cada medidor. Además de esto, dentro de cada sub-sección se encuentran las gráficas equivalentes a cada uno de los parámetros capturados: Potencia, tensión, corriente, frecuencia, temperatura, distorsión armónica total en tensión, distorsión armónica total en corriente y fase, respectivamente. Además, la página cuenta con un menú desplegable que le permite al usuario desplazarse por la página de una forma más rápida y sencilla. Este menú se divide al igual que la página y dependiendo el elemento del menú que se selecciona, la pagina se dirige a la posición del elemento inmediatamente, ahorrando el tiempo que se gastaría desplazándose con la barra de navegación. Este menú se puede evidenciar de una mejor forma en la figura 6.17.

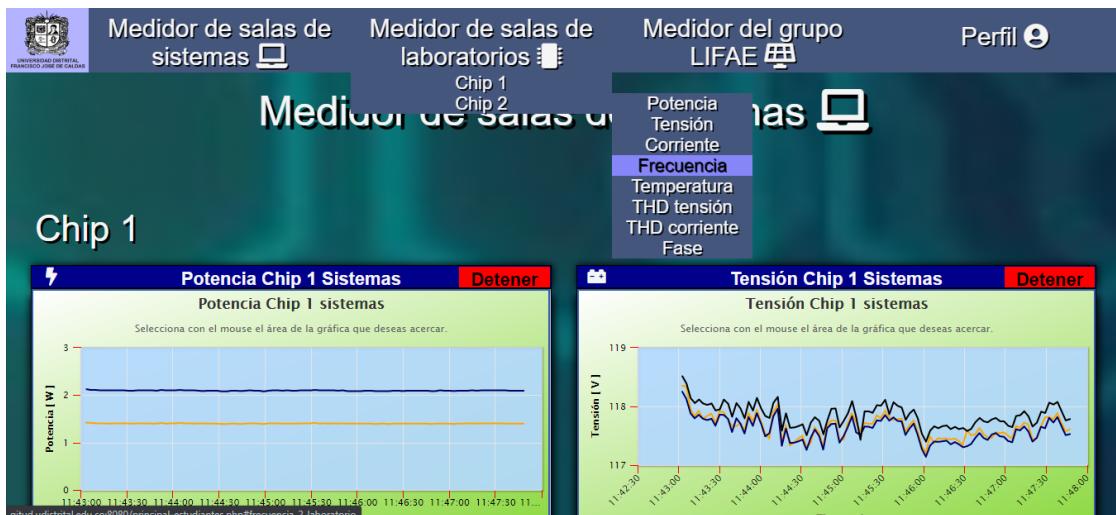


Figura 6.17: Captura de la vista de escritorio del menú desplegable de la página principal del aplicativo web.

Como se puede evidenciar en la figura 6.16, se observan las gráficas de potencia y tensión para el chip 1 del medidor ubicado en salas de sistemas. Cada gráfica se realiza para un único parámetro y es completamente independiente de los demás parámetros, sin mencionar que se muestran simultáneamente las tres curvas para las variables que poseen tres líneas.

Cada uno de los contenedores que incluyen una gráfica, se encuentra en constante actualización de sus datos. Esto se debe, a que el objetivo del aplicativo web es precisamente visualizar los datos de una forma dinámica y automática; por ese motivo, estos gráficos se actualizan cada 5 minutos, mostrando los datos capturados y almacenados en los últimos 5 minutos. Es por esta misma razón que cada uno de los contenedores posee un botón de colocar rojo con la palabra “Detener” en la esquina superior derecha del contenedor, ya que si se desea visualizar los datos graficados con mayor detenimiento, no sería posible esta tarea por la constante actualización. Por lo tanto, este botón lo que hace es detener la actualización el tiempo que desee el usuario. Para reiniciar las actualizaciones basta con presionar nuevamente el mismo botón; sin embargo, cuando se quiere reiniciar las actualizaciones, el botón ha cambiado su diseño y color, en ese caso el botón que activa las actualizaciones es de color verde, con la palabra “Iniciar” en el centro. En la figura 6.18 se puede evidenciar mejor este cambio en el botón, una vez se detuvieron las actualizaciones.



Figura 6.18: Contenedor del gráfico de tensión para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para estudiantes.

Como se puede evidenciar en la figura 6.18 el botón de la esquina superior derecha ha cambiado con respecto al que se puede observar en la figura 6.16, esto se debe a que al presionarlo para detener o iniciar las actualizaciones, cambia su diseño más no su tamaño o ubicación.

Un factor clave que es de suma importancia con respecto a estos contenedores, es su capacidad de adaptabilidad a los datos. Estas gráficas se realizan utilizando librerías y bibliotecas de desarrollo Front-End, las cuales son responsivas, adaptables e interactivas con el usuario. En la gráfica mostrada en la figura 6.18 se muestran los valores capturados y el tiempo de captura de los datos de tensión, esta variable posee 3 líneas por lo que permite mostrar las 3 curvas simultáneamente, pero además de eso, permite también desactivar cualquiera de las curvas o incluso todas, para poder tener una vista con mucho mayor detalle para solo la curva o las curvas de interés. Como en la figura, en la que se desactivó la curva de tensión de la línea B, para poder observar mejor las otras dos.

Se dice que estas gráficas son responsivas y adaptables, porque dependen únicamente de los datos, es decir, según el rango y el dominio de los datos ingresados, la gráfica ajusta sus tamaños y etiquetas automáticamente; reduciendo en gran medida la intervención por parte del usuario en términos de ajuste de visualización. También se dice que son interactivas porque permiten al usuario desactivar las curvas irrelevantes, tal y como se dijo anteriormente y además permite realizar acercamientos dentro de las gráficas, esto con el fin de visualizar solo un rango de interés determinado. En la figura 6.19 se evidencian los cambios en el gráfico cuando se realizan acercamientos.



Figura 6.19: Contenedor del gráfico de tensión acercado para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para estudiantes.

Como se puede observar en la figura 6.19, cuando se realizan acercamientos en las curvas de los datos dentro de un gráfico, automáticamente se ajustan los tamaños de los ejes, de la curva, las cuadrículas y las etiquetas de los ejes según los datos mostrados. El máximo acercamiento permitido es de 4 marcas o puntos de la curva y para restaurarlo a su estado original, basta con presionar el botón “Reset zoom”, los ejes y las curvas vuelven a sus tamaños y etiquetas iniciales.

Es importante mencionar que el botón de detener las actualizaciones es independientemente del contenedor que lo incluya, ya que si se presiona cualquiera de los botones, las actualizaciones se detienen para todas las gráficas. Lo mismo pasa en sentido contrario, si se presiona el botón de iniciar las actualizaciones, no importa cual se presione, se activan para todos los contenedores.

Como factor general independiente del tipo de usuario, para salir de la página principal se debe seleccionar el elemento “Cerrar sesión” en la sección del menú desplegable “Perfil”, esta acción hará que se salga de la cuenta del usuario y la página retorne automáticamente a la pagina de inicio.

5. Página principal para docentes

La página principal para los docentes, es muy similar a la página para los estudiantes, de hecho, a simple vista se podría decir que son idénticas. Sin embargo, como ya se ha mencionado, los privilegios de un tipo de usuario con respecto a otro cambian. Por lo que la principal diferencia de la página de docentes con respecto a la de los estudiantes, radica en el diseño y comportamiento de los gráficos dentro de los contenedores, principalmente después de un evento en particular. Una vez que un usuario identificado como docente ha realizado correctamente el proceso de inicio de sesión utilizando sus credenciales de acceso, la página de inicio se redirige automáticamente a la página principal de docentes, la vista de escritorio de esta página se puede evidenciar en la figura 6.20.

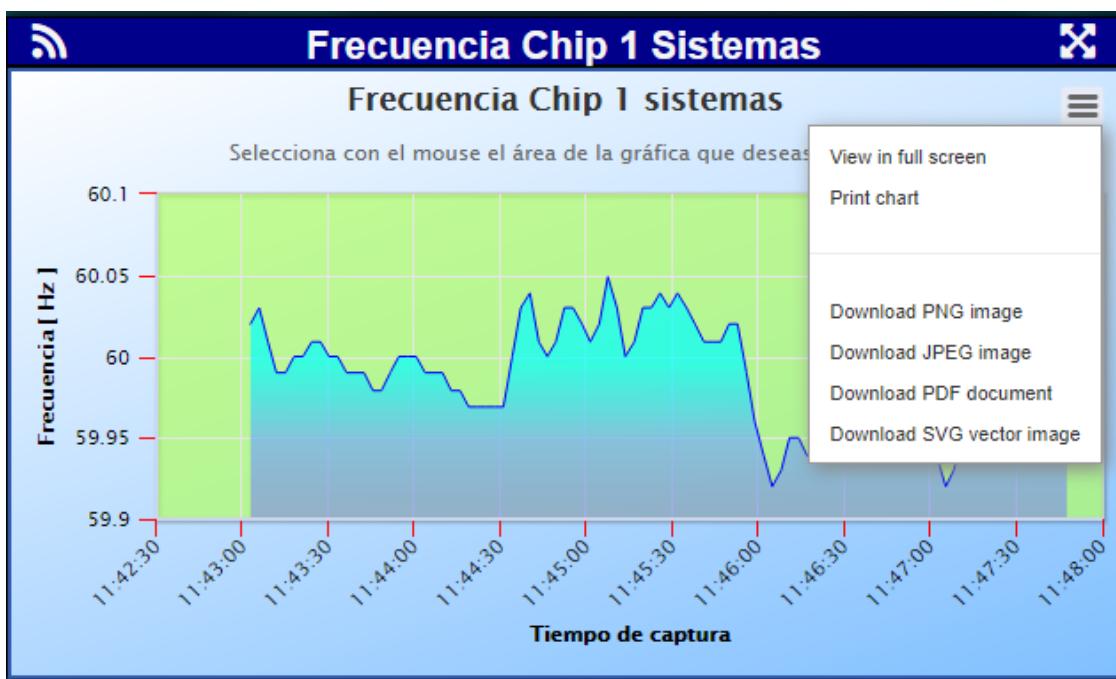


Figura 6.20: Contenedor del gráfico de frecuencia para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para docentes.

Como se puede observar en la figura 6.20, el contenedor mostrado expone la gráfica de la frecuencia, la cual, es una gráfica de área a diferencia de las gráficas de línea creadas para las variables que dependen de las tres líneas. Un contenedor de la página principal de docentes posee dos diferencias clave con respecto a un contenedor de la página principal de estudiantes como el que se muestra en la figura 6.18. La primera diferencia se encuentra en el botón gris con 3 líneas horizontales ubicado cerca de la esquina superior derecha del gráfico; este botón permite realizar diferentes acciones con el gráfico, como por ejemplo: Visualizarlo en pantalla completa, imprimir el gráfico o descargarlo en diferentes formatos (PNG, JPEG, PDF o SVG).

Por otra parte, la otra diferencia se encuentra en el botón de la parte superior derecha del contenedor. Como se observa, ya no existe el botón para detener las actualizaciones, en su lugar existe un botón mucho más pequeño, pero el cual, al ser presionado permite expandir el gráfico junto con la cantidad de datos mostrados. Si bien en la página principal se pueden evidenciar gráficos de los datos capturados y almacenados en los últimos 5 minutos, cuando se presiona el botón para maximizar el gráfico, se muestran los datos capturados y almacenados en el último día. Este gráfico se puede evidenciar en la figura 6.21 para el caso de la potencia del chip 1 del medidor de sistemas.

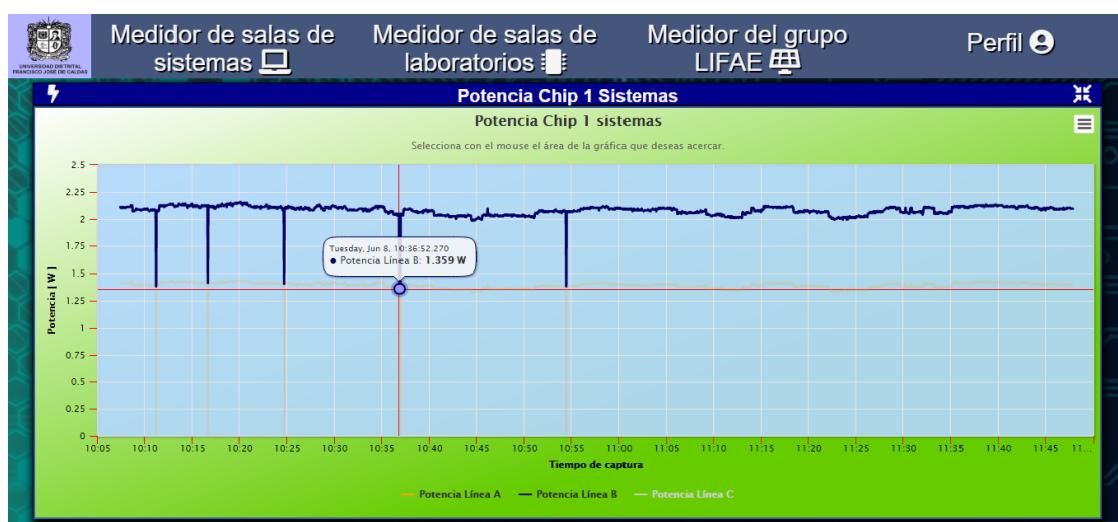


Figura 6.21: Gráfico maximizado de potencia para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para docentes.

Como se puede observar en la figura 6.21, se tiene un mayor rango de valores que en la parte inicial, a pesar de no ser un día completo de datos, muestra los de el último día, es decir, muestra todos los datos que se encuentran en la base de datos, del último día medido. Al igual que con todos los demás gráficos, se puede desactivar o activar las líneas de interés y se puede realizar acercamientos a las secciones de tiempo que se desee observar con mayor detalle. También se pueden realizar las mismas acciones que en los contenedores iniciales, expandir el gráfico a pantalla completa o descargarlo en diferentes formatos. Como se observa en la figura, al colocar el mouse sobre alguno de los datos, se genera un cursor en el eje X y en el Y, que le permite conocer el dato del valor de la variable y del tiempo de captura con mayor precisión.

Cabe mencionar que una vez se presiona el botón de expandir el gráfico en cualquier contenedor, se detiene automáticamente las actualizaciones para todas las variables en todos los medidores, pues no es necesario que se actualicen dichos datos si no se pueden visualizar. Esto se debe a que al maximizar un gráfico, solo se puede visualizar ese gráfico, todos los demás permanecen ocultos hasta que el usuario minimice el gráfico expandido; de igual forma pasa con los accesos directos del menú desplegable, pues al no tener ningún contenedor al cual dirigirse, se mantienen en la misma posición. Para realizar esta acción, basta con presionar nuevamente el mismo botón, el cual ahora presenta un nuevo símbolo, indicando que ahora realizará la acción de minimizar.

6. Página principal para supervisores

Como se ha mostrado con las diferentes páginas principales, los privilegios se otorgan al usuario en orden jerárquico, es decir, la página principal de docentes posee todos los privilegios de la página principal de estudiantes y algunos extras como se explicó en la sección anterior. De la misma forma, la página principal de supervisores cuenta con todos los privilegios de la página principal de docentes junto con algunos elementos extra que son únicamente para los usuarios identificados como supervisores. Las diferencias de una página con respecto a la otra se evidencian cuando se expande alguno de los gráficos iniciales, en la figura 6.22 se evidencia el cambio de diseño en la gráfica expandida de alguna de las variables, en este caso, para la temperatura del chip 1 del medidor de sistemas.

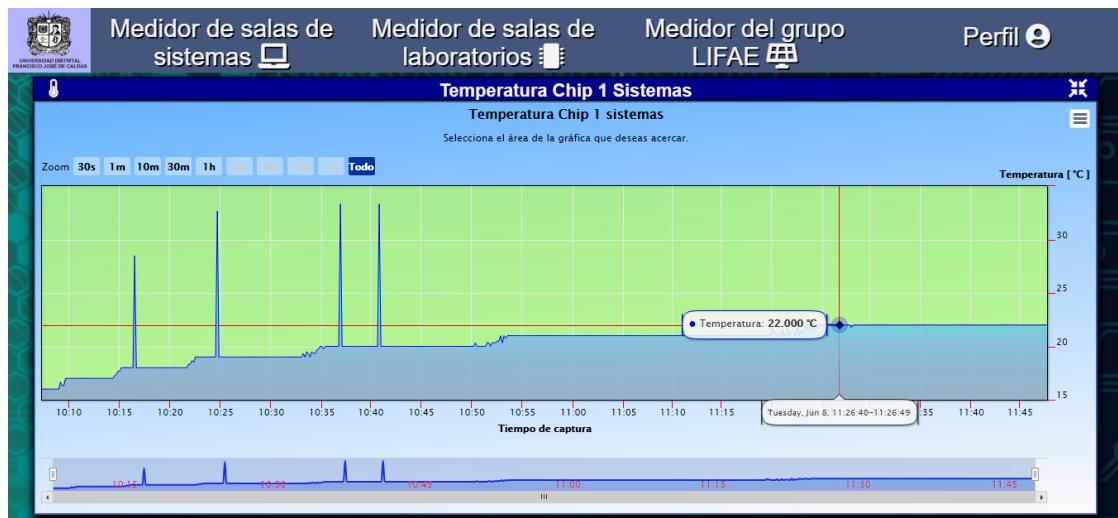


Figura 6.22: Gráfico maximizado de potencia para el chip 1 del medidor de sistemas mostrado en la interfaz gráfica en la página principal para supervisores.

Como se puede observar en la figura 6.22, la gráfica de temperatura y de frecuencia comparten la característica de ser gráficas de área, al depender de una única línea y no de tres líneas como los demás parámetros. A diferencia de las gráficas expandidas en la página principal de docentes, estos gráficos poseen característica adiciones, como por ejemplo, botones con zoom automático específicos y un navegador adicional del gráfico.

Los gráficos expandidos de esta página también muestran los datos capturados y almacenados durante el último día y también permiten realizar acercamientos manuales de las secciones de interés. Sin embargo, los botones de zoom específicos permiten realizar acercamientos específicos de tiempo, por ejemplo, 30 segundos, 1 minuto, 10 minutos, 30 minutos, etc, como también existe el botón que permite observar todo el gráfico. Dependiendo la cantidad de datos mostrados se pueden activar botones con un zoom más grande, por ejemplo, 3 horas, 6 horas, 12 horas, etc.

Por otra parte, el navegador que se encuentra en la parte inferior del gráfico muestra una pequeña gráfica de todos los datos en una menor escala. Este elemento permite seleccionar manualmente rangos de zoom distintos a los establecidos en los botones, también se pueden mover sus límites a conveniencia del usuario, para visualizar secciones de mayor interés sin problemas. Todos estos gráficos pueden visualizarse en pantalla completa y descargarse en los diferentes formatos.

Al finalizar de cada página principal, para todos los usuarios se encuentra un pie de página con información del proyecto y los logos de los grupos de investigación relacionados en el proyecto. Además, en la figura 6.23 se muestra el botón que permite mostrar la tabla de estados para el parámetro en cuestión. Este botón solo es visible para los usuarios identificados como supervisores y se mantiene oculto hasta que se expande alguno de los gráficos, de esta manera se asegura que solo se seleccione un parámetro.

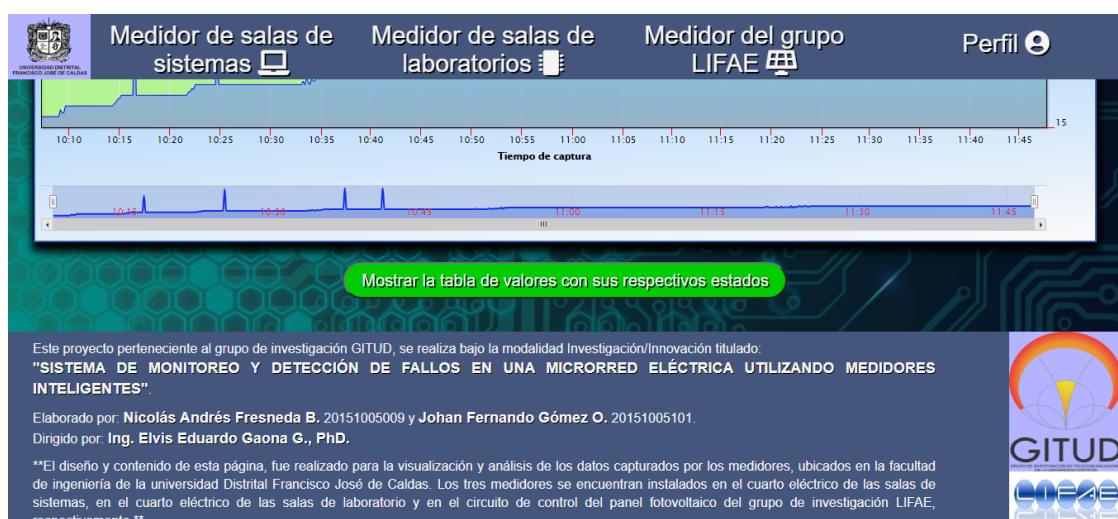


Figura 6.23: Pie de página de la interfaz gráfica y botón de mostrar tabla de estados en la página principal para supervisores.

En la figura 6.23 se puede observar que este botón se encuentra debajo de la gráfica expandida, por lo que solo es visible al deslizar la barra de navegación hacia abajo. Al presionar este botón se hace visible una tabla con los valores tal y como se encuentran almacenados en las bases de datos. La cantidad de registro o filas mostradas equivalen a la misma cantidad de datos que se muestran en el gráfico expandido; esto se realiza con el fin de que el supervisor pueda consultar el estado en el que se encontraba cada dato del parámetro en su momento de captura. La tabla generada se muestra en la figura 6.24.

CICLO	VALOR	TIEMPO	ESTADO
1	16	2021-06-08 10:07:23.226907	Normal
2	16	2021-06-08 10:07:26.275818	Normal
3	16	2021-06-08 10:07:29.247944	Normal
4	16	2021-06-08 10:07:32.429832	Normal
5	16	2021-06-08 10:07:35.293347	Normal
6	16	2021-06-08 10:07:38.409252	Normal
7	16	2021-06-08 10:07:41.431725	Normal
8	16	2021-06-08 10:07:44.618496	Normal
9	16	2021-06-08 10:07:47.512504	Normal
10	16	2021-06-08 10:07:50.691256	Normal
11	16	2021-06-08 10:07:53.666052	Normal
12	16	2021-06-08 10:07:56.651895	Normal
13	16	2021-06-08 10:07:59.706175	Normal
14	16	2021-06-08 10:08:02.842567	Normal
15	16	2021-06-08 10:08:05.690512	Normal
16	16	2021-06-08 10:08:08.770488	Normal
17	16	2021-06-08 10:08:11.746325	Normal
18	16	2021-06-08 10:08:14.931727	Normal
19	16	2021-06-08 10:08:17.918779	Normal
20	16	2021-06-08 10:08:21.116099	Normal
21	16	2021-06-08 10:08:24.087551	Normal

Figura 6.24: Tabla de valores y botón de mostrar/ocultar la tabla de estados en la página principal para supervisores para la temperatura del chip 1 del medidor de sistemas.

Como se puede observar en la figura 6.24, la estructura de los datos en la tabla de estados es la misma que la mostrada en la tabla 5.4 creada en MySQL para el almacenamiento de los datos de temperatura en el servidor. En la figura se muestran poco más de 20 registros, sin embargo esta tabla tiene aproximadamente 2000 registros equivalentes a los datos mostrados en el gráfico expandido. También se puede evidenciar que el botón en la parte superior cambio el texto que tenía internamente. Inicialmente el texto era “Mostrar la tabla de valores con sus respectivos estados” tal y como se muestra en la figura 6.23, mientras que ahora es “Ocultar la tabla de valores con sus respectivos estados”; este cambio de texto se realiza en ambos sentidos cuando se presiona el botón. Como se mencionó anteriormente, este botón solo es visible cuando se expande alguno de los gráficos, por lo que si se minimizan a su estado inicial el botón y la tabla se ocultan inmediatamente. Al igual que con las gráficas en la página principal de docentes, los gráficos detienen sus actualizaciones cuando alguno de los contenedores se expande y de la misma forma, se activan las actualizaciones cuando el gráfico expandido se minimiza.

En el menú desplegable de “Perfil”, se encuentra la opción de “Tabla de Fallas críticas”, que redirige hacia la tabla de fallas críticas capturadas por el sistema. Allí se evidencia un listado con las últimas 20 fallas de baja impedancia registradas por el medidor. La información se encuentra ordenada por fecha, desde la más reciente a la más antigua, junto con la ubicación y el estado de la falla, es decir, el medidor, el chip, el tipo de falla y el tiempo exacto de detección. Una vez se ingresa a la página que muestra el listado, aparece en pantalla un mensaje de alerta que muestra los datos y características de la última falla registrada en la base de datos, esto con el fin, de que el usuario pueda visualizar rápidamente la falla crítica más reciente y pueda tomar acciones correctivas si es necesario. En la figura 6.25 se evidencia una captura de pantalla de la página de visualización de las fallas críticas.

ID	MEDIDOR	CHIP	FALLA	TIEMPO
30	labs	2	Falla trifásica	18:59 oct
29	labs	2	Falla trifásica	18:59 oct
28	labs	2	Falla trifásica	18:59 oct
27	labs	2	Falla trifásica	18:59 oct
26	labs	2	Falla trifásica	18:59 oct
25	labs	2	Falla trifásica	18:59 oct
24	labs	2	Falla trifásica	18:59 oct
23	labs	2	Falla trifásica	18:59 oct
22	labs	2	Falla trifásica	18:59 oct
21	labs	2	Falla trifásica	18:59 oct
20	labs	2	Falla trifásica	18:59 oct
19	labs	2	Falla trifásica	18:59 oct
18	labs	2	Falla trifásica	18:59 oct
17	labs	2	Falla trifásica	18:59 oct
16	labs	2	Falla trifásica	18:59 oct
15	labs	2	Falla trifásica	18:59 oct
14	labs	2	Falla trifásica	18:59 oct
13	labs	2	Falla trifásica	18:59 oct
12	labs	2	Falla trifásica	18:59 oct
11	labs	2	Falla trifásica	18:59 oct

Figura 6.25: Página de visualización del listado de fallas críticas detectadas por el sistema.

Como se observa en la figura 6.25, el listado muestra solo 20 registros, esto se decide como parámetro de diseño, para no alargar el listado innecesariamente y estar mostrando siempre la misma información. También se ajusta este parámetro, en términos de rapidez, ya que estas fallas son las más relevantes para el usuario, es necesario que se muestren de la manera más rápida posible. Debido a que se actualiza la tabla cada vez que se abre la página, no se pierde tiempo consultando y cargando datos demasiado antiguos, sino por el contrario, solo los 20 más recientes, de esta manera se agiliza la búsqueda y se aligera la carga de datos para mostrar, permitiendo al usuario tomar acciones más eficaces.

Capítulo 7

Implementación, pruebas y resultados

Este capítulo tiene como finalidad explicar el procedimiento realizado, durante la implementación física del proyecto sobre una red trifásica, la cual fue el objetivo principal de este.

Inicialmente, se muestra la topología de la red implementada, la cual esta compuesta por una línea de alimentación trifásica y un neutro, los cuales son conectados al medidor, el cual se encargará de enviar los datos al servidor web y el servidor de base de datos, con el fin de visualizar en pantalla los valores tomados, la figura 7.1, muestra este comportamiento.

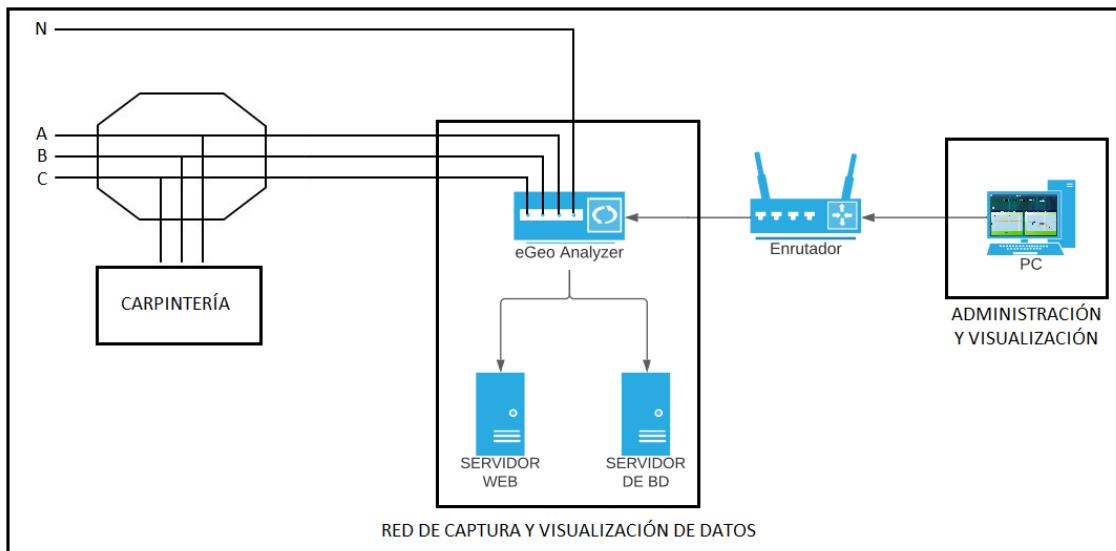


Figura 7.1: Topología de red de la prueba trifásica implementada

Seguidamente, se presenta la implementación real del medidor, la cual fue realizada en una carpintería del sector de Engativa, esta tiene una red trifásica que refiere una fase a neutro, la figura 7.2 muestra el montaje realizado.



Figura 7.2: Implementación real del medidor para la red trifásica a analizar.

De la prueba realizada en la implementación física, se muestran los resultados obtenidos de los parámetros de la red eléctrica tomados por el medidor inteligente. Primero, se muestra una captura de los valores tomados en tiempo real sobre el video de evidencia, tomado durante la prueba. La figura 7.3, muestra los valores obtenidos en tiempo real para dicha red. Como puede observarse en la figura, únicamente se presentan valores de corriente para la fase B de la red, dado que esta era la única que estaba referida a neutro, así mismo correspondería a los valores de THD de corriente, únicamente se obtuvieron valores para la fase B. Correspondiente a los demás registros, obtenemos valores que están dentro de los rangos normales de funcionamiento y estos son imprimidos en pantalla y graficados en la interfaz web, la evidencia de esta captura se presenta en las siguientes figuras.

THD1 Linea B chip 2	: 73.210	%, Tiempo: 2021-11-06 18:46:59.210881 , Estado: Anormal
Potencia Linea A chip 2	: 0.000	[mW], Tiempo: 2021-11-06 18:46:59.263586 , Estado: Normal
Potencia Linea B chip 2	: 0.000	[mW], Tiempo: 2021-11-06 18:46:59.315134 , Estado: Normal
Potencia Linea C chip 2	: 0.000	[mW], Tiempo: 2021-11-06 18:46:59.369776 , Estado: Normal
Fase Linea A chip 2	: 0.000	DEG, Tiempo: 2021-11-06 18:46:59.425105 , Estado: Normal
Fase Linea B chip 2	: 312.700	DEG, Tiempo: 2021-11-06 18:46:59.482334 , Estado: Normal
Fase Linea C chip 2	: 119.800	DEG, Tiempo: 2021-11-06 18:46:59.534426 , Estado: Normal
Tensión linea A chip 1	: 123.951	[V], Tiempo: 2021-11-06 18:46:59.588613 , Estado: Normal
Tensión linea B chip 1	: 126.204	[V], Tiempo: 2021-11-06 18:46:59.916231 , Estado: Normal
Tensión linea C chip 1	: 126.695	[V], Tiempo: 2021-11-06 18:46:59.976569 , Estado: Sobretensión
Corriente linea A chip 1	: 0.000	[mA], Tiempo: 2021-11-06 18:46:59.033746 , Estado: Sobretensión
Corriente linea B chip 1	: 0.000	[mA], Tiempo: 2021-11-06 18:46:59.081924 , Estado: Normal
Corriente linea C chip 1	: 0.000	[mA], Tiempo: 2021-11-06 18:46:59.151788 , Estado: Normal
Frecuencia chip 1	: 59.970	[Hz], Tiempo: 2021-11-06 18:46:59.208477 , Estado: Normal
Temperatura chip 1	: 21	[C], Tiempo: 2021-11-06 18:46:59.313742 , Estado: Normal
THDv Linea A chip 1	: 0.000	%, Tiempo: 2021-11-06 18:46:59.360454 , Estado: Normal
THDv Linea B chip 1	: 9.070	%, Tiempo: 2021-11-06 18:46:59.563520 , Estado: Anormal
THDv Linea C chip 1	: 0.000	%, Tiempo: 2021-11-06 18:46:59.614981 , Estado: Normal
THD1 Linea A chip 1	: 0.000	%, Tiempo: 2021-11-06 18:46:59.665443 , Estado: Normal
THD1 Linea B chip 1	: 0.000	%, Tiempo: 2021-11-06 18:46:59.717373 , Estado: Normal
THD1 Linea C chip 1	: 0.000	%, Tiempo: 2021-11-06 18:46:59.848345 , Estado: Normal
Potencia Linea A chip 1	: 0.000	[mW], Tiempo: 2021-11-06 18:46:59.900528 , Estado: Normal
Potencia Linea B chip 1	: 0.000	[mW], Tiempo: 2021-11-06 18:46:59.956357 , Estado: Normal
Potencia Linea C chip 1	: 0.000	[mW], Tiempo: 2021-11-06 18:47:00.012460 , Estado: Normal
Fase Linea A chip 1	: 0.000	DEG, Tiempo: 2021-11-06 18:47:00.219099 , Estado: Normal
Fase Linea B chip 1	: 312.800	DEG, Tiempo: 2021-11-06 18:47:00.272752 , Estado: Normal
Fase Linea C chip 1	: 119.000	DEG, Tiempo: 2021-11-06 18:47:00.326757 , Estado: Normal
Tensión linea A chip 2	: 122.612	[V], Tiempo: 2021-11-06 18:47:00.383333 , Estado: Normal
Tensión linea B chip 2	: 124.370	[V], Tiempo: 2021-11-06 18:47:00.449633 , Estado: Normal
Tensión linea C chip 2	: 124.895	[V], Tiempo: 2021-11-06 18:47:00.505144 , Estado: Normal
Corriente linea A chip 2	: 0.000	[mA], Tiempo: 2021-11-06 18:47:00.566893 , Estado: Normal
Corriente linea B chip 2	: 16.016	[mA], Tiempo: 2021-11-06 18:47:00.625470 , Estado: Normal
Corriente linea C chip 2	: 0.000	[mA], Tiempo: 2021-11-06 18:47:00.682729 , Estado: Normal
Frecuencia chip 2	: 59.970	[Hz], Tiempo: 2021-11-06 18:47:00.682729 , Estado: Normal
Temperatura chip 2	: 20	[C], Tiempo: 2021-11-06 18:47:00.674156 , Estado: Normal
THDv Linea A chip 2	: 0.000	%, Tiempo: 2021-11-06 18:47:01.002683 , Estado: Normal
THDv Linea B chip 2	: 9.220	%, Tiempo: 2021-11-06 18:47:01.050685 , Estado: Normal
THDv Linea C chip 2	: 0.000	%, Tiempo: 2021-11-06 18:47:01.102651 , Estado: Anormal
THD1 Linea A chip 2	: 0.000	%, Tiempo: 2021-11-06 18:47:01.159639 , Estado: Normal
THD1 Linea B chip 2	: 74.380	%, Tiempo: 2021-11-06 18:47:01.204656 , Estado: Normal
THD1 Linea C chip 2	: 0.000	%, Tiempo: 2021-11-06 18:47:01.257885 , Estado: Anormal

Figura 7.3: Datos reales obtenidos sobre la consola del medidor correspondientes a la red trifásica

Como primer parámetro, se presenta la potencia medida sobre la red eléctrica, para la cual no se presentan valores medidos sobre dicha red, esto pudo deberse a que los dispositivos y maquinaria eléctrica estaban en baja producción, lo que causó que se evidencia muy poca energía durante el momento de la prueba, la figura 7.4, muestra los valores de potencia registrados sobre el medidor.

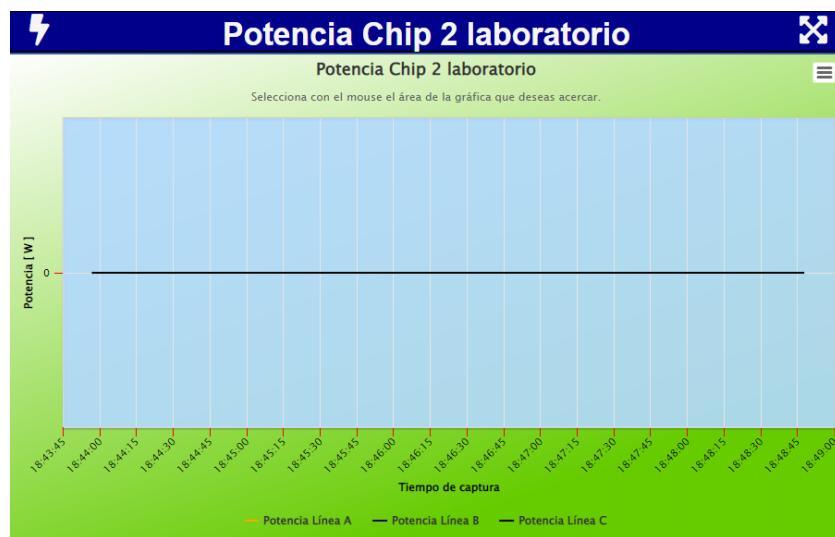


Figura 7.4: Gráfica de potencia registrada sobre las pruebas.

Seguidamente, obtenemos el valor de la tensión para las tres fases, la cual presenta un rango de operación entre 122V a 124V en promedio, durante las pruebas se pueden observar algunas variaciones de tensión, sin embargo, en gran parte del intervalo de tiempo medido, se observa estabilidad en la red, la figura 7.5 ilustra lo presentado anteriormente.

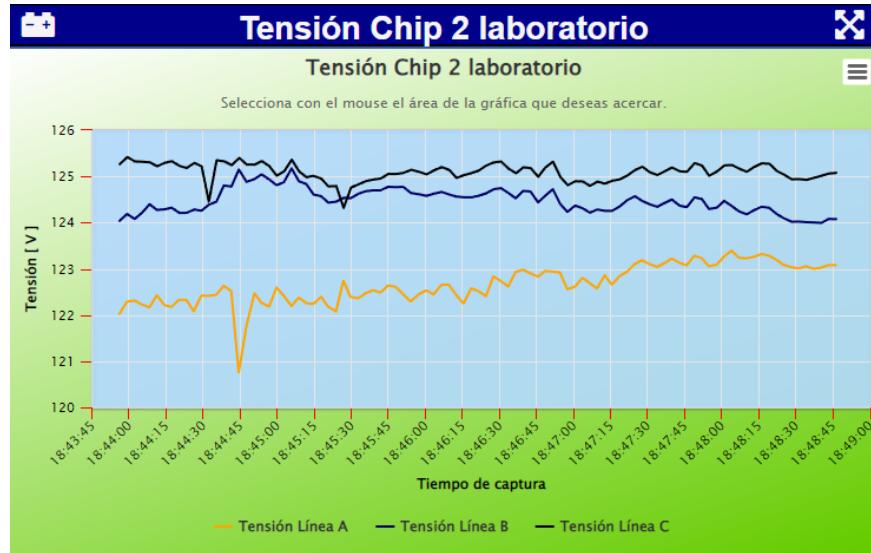


Figura 7.5: Gráfica de tensión registrada sobre las pruebas.

Respecto a los valores de corriente, como ya fue explicado anteriormente, únicamente se tiene registro para la fase B, debido a que esta era la única fase que estaba referida a neutro, esta fase presentaba valores entre los 12.5mA y 15mA, aquí se evidencia que la resolución de medida del dispositivo permite obtener mediciones precisas para valores bajos de corriente, es decir, del orden de los mA. La figura 7.6 muestra dicho comportamiento.

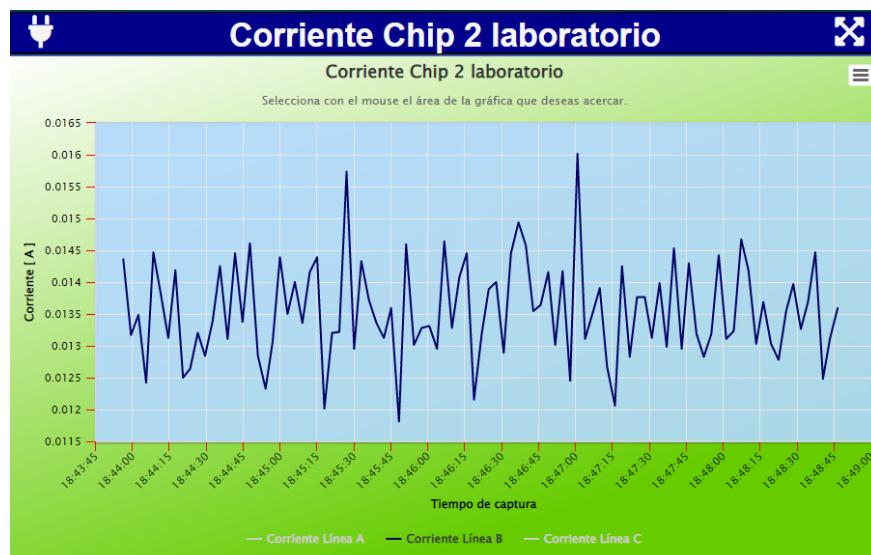


Figura 7.6: Gráfica de corriente registrada sobre las pruebas.

Para el valor de frecuencia obtenido durante la prueba, podemos evidenciar que dichos

valores no sobrepasan en más de un 5 % el valor estándar de 60 Hz para redes eléctricas en Colombia, se tienen valores registrados entre 59.925 Hz y 60.07 Hz, por lo cual, se puede afirmar que dichos parámetros se encuentran dentro de los rangos normales de funcionamiento. Este comportamiento se puede ver en la figura 7.7

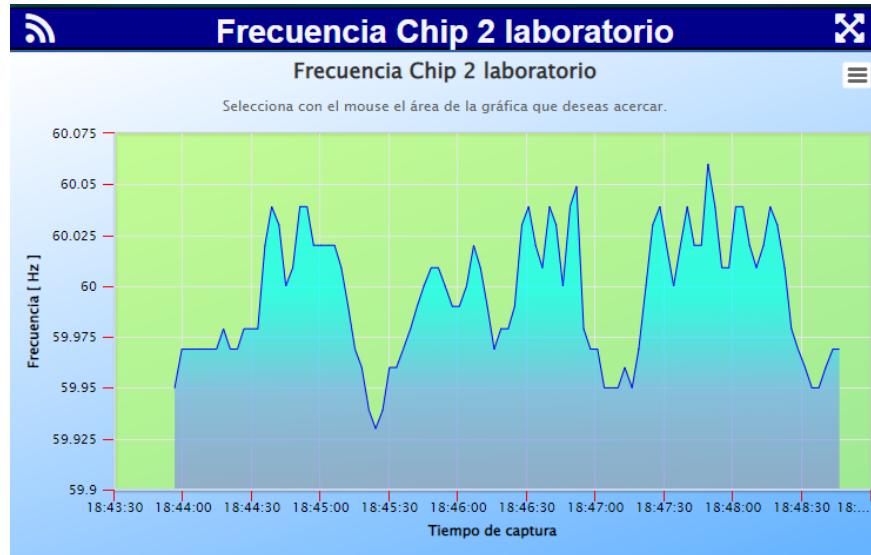


Figura 7.7: Gráfica de frecuencia registrada sobre las pruebas.

Con relación a la temperatura del medidor durante la prueba, se puede observar que este mantuvo una temperatura estable de 20 Celsius a lo largo de toda la prueba, es decir, las componentes de hardware del medidor no sufrieron estrés o sobrecarga en la ejecución de los scripts de medición. La figura 7.8 muestra la estabilidad de este parámetro.

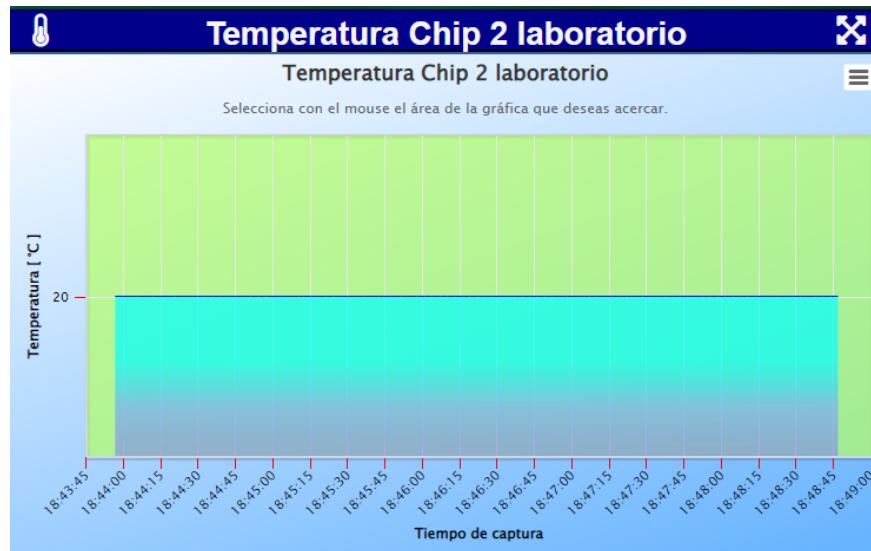


Figura 7.8: Gráfica de temperatura registrada sobre las pruebas.

Correspondiente a los valores de THD de tensión, se puede observar que el rango medido se encuentra dentro del 9.2 % a 9.5 % y presenta valores estables, esto puede

deberse a la baja operatividad de la maquinaria instalada en dicha red, es posible que al momento de aumentarla, se presenten mayores variaciones y/o mayores valores en este parámetro. La figura 7.9 ilustra los valores registrados durante la prueba para el parámetro THD de tensión.

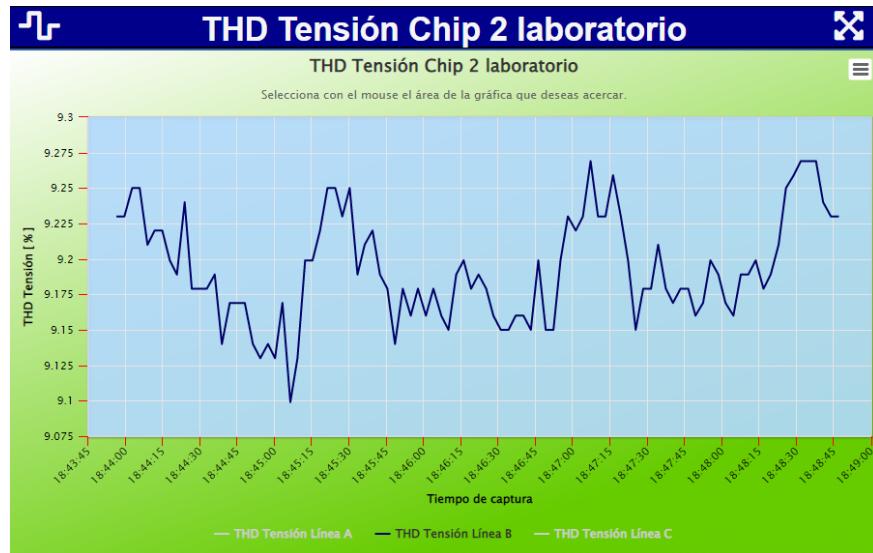


Figura 7.9: Gráfica de distorsión armónica de tensión registrada sobre las pruebas.

De igual manera que la corriente, únicamente se presentan valores en THD de corriente para la fase B, los cuales fluctúan entre el 60 % y 80 %, para estos valores si se evidencia una variación significativa a pesar de que la maquinaria no estaba en operatividad. La figura 7.10 muestra este comportamiento.

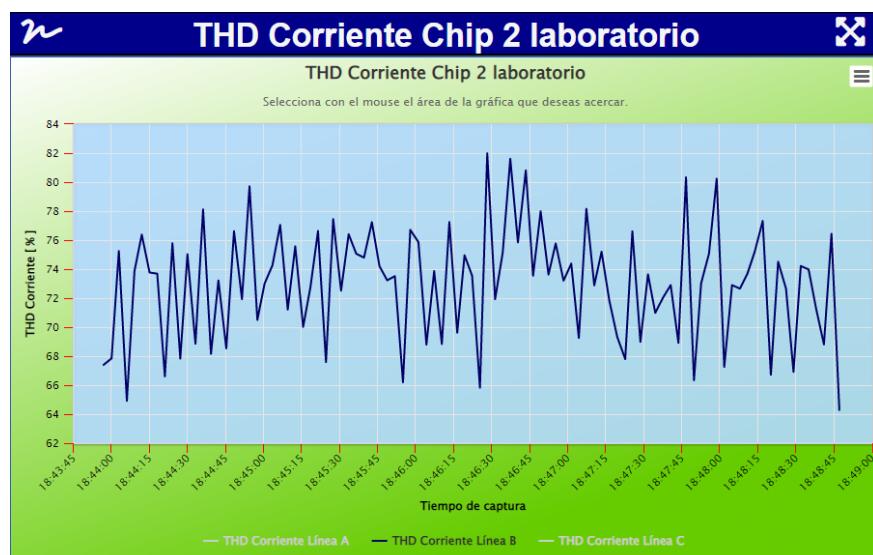


Figura 7.10: Gráfica de distorsión armónica de corriente registrada sobre las pruebas.

Finalmente, en la figura 7.11 se muestra la medición de las fases de la red trifásica, donde se puede observar que se tienen valores de 0° , 120° y 310° respectivamente para cada fase.

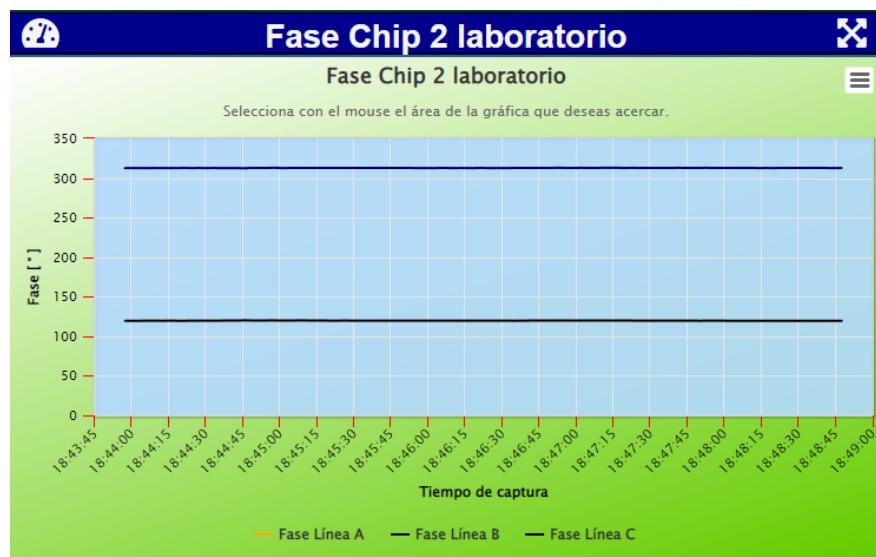


Figura 7.11: Gráfica de fases para la red trifásica registrada sobre las pruebas.

Capítulo 8

Conclusiones

Se pudo emplear un conjunto de medidores inteligentes, interconectados entre sí, para cuantificar parámetros propios de una red eléctrica, almacenar y visualizar dicha información; con el fin de llegar hasta usuario, se evalúan criterios de detección para determinar fallas leves y críticas en el sistema, mostrando al usuario el tiempo exacto y la ubicación de dicha falla. De esta manera, el usuario puede tomar acciones correctivas, en caso de ser necesarias, sin necesidad de perder tiempo encontrando la ubicación de la falla.

Utilizar medidores inteligentes como el eGEO Analyzer, permite tener una mayor confiabilidad en las medidas tomadas, en comparación con otros dispositivos de captura. Esto se debe a la alta precisión que maneja el medidor, junto con su compatibilidad con el sistema de Linux y el lenguaje Python, permitiendo que una amplia gama de librerías interactúen y faciliten todo el proceso de tratamiento de datos. Sin embargo, el controlador cuenta con una desventaja, la cual es emplear comunicación serial, pues a diferencia de otros dispositivos, este no permite realizar acciones en paralelo, mucho menos realizar interrupciones. Esto es un factor clave, ya que durante el proceso de captura y tratamiento de los datos, solo se puede tomar un parámetro a la vez; también mientras se envía una carga de información al servidor, no se pueden tomar medidas, por lo que existe un pequeño lapso de tiempo, en los que el sistema queda a ciegas. Aunque este lapso no es un periodo de tiempo muy largo, puede ser decisivo en una situación particular.

Establecer un tiempo de espera adecuado, permitió lograr una toma de datos limpia y confiable. Debido a que el protocolo de comunicación serial generaba problemas de solapamiento, fallos en las mediciones y corrimiento de los datos, al utilizar tiempos demasiado pequeños o tiempos considerables. Después de ajustar finamente este valor, se estableció un tiempo de espera igual a 300 ms sobre cada objeto de tipo serial en cada uno de los medidores. Además del afinamiento de este tiempo de espera, se decidió emplear dos objetos seriales que permitieran realizar las medidas de los diferentes parámetros de manera intercalada, con el fin de reducir aún mas la probabilidad de un corrimiento o ausencia de una medida.

El uso de contenedores sobre el servidor principal permite obtener una topología de red intercomunicada y a su vez independiente, que posee la capacidad de realizar diferentes consultas a través del aplicativo web, sin presentar conflictos a nivel de puerto o

peticiones. Esta topología de red garantiza que un posible error no se propague por todas las bases de datos del sistema, reduciendo la confiabilidad del aplicativo, sino que por el contrario, llegue únicamente a una de ellas y pueda detectarse y corregirse de una manera mucho más eficaz. Para poder llevar a cabo esta topología de red, se debe establecer una correcta conexión entre las bases de datos, re-mapear los puertos por defecto en el contenedor para evitar interferencia sobre sí mismo, y por último, se debe modificar los permisos a nivel de firewall sobre el servidor principal.

Con el fin de emplear un flujo de tráfico de datos cifrado y seguro sobre internet, dedicado al sistema en general, pero principalmente sobre el aplicativo web, se utilizó la configuración del protocolo SSL, sobre el aplicativo. De esta manera se puede trabajar sobre tráfico cifrado HTTPs, contando con un certificado SSL auto-firmado; adicionalmente, se realiza el cierre en la comunicación a nivel del aplicativo web mediante el protocolo no cifrado HTTP.

La limitante principal al momento de realizar las pruebas en la implementación física sobre la red trifásica, radica en que dicha red no contaba con un neutro referido a las 3 fases, es decir, se realizó la conexión de manera directa a las terminales de tensión de la red con el medidor, sin embargo, el neutro de la red fue tomado de una red monofásica. Debido a esto, no fue posible obtener valores completos de corriente, THD de tensión y THD de corriente sobre las fases A y C de la red.

Este proyecto tiene un enfoque a futuro bastante amplio, debido a que por cuestiones ajenas a nosotros y la Universidad Distrital, no pudo implementarse de la manera en la que se tenía planeada, con sus dos puntos de consumo y uno de generación. Sin embargo, el sistema de medición queda completamente funcional en caso de que se quiera seguir con él. Puede ser extendido, agregando una mayor cantidad de medidores y sensores, con el fin de tener una microrred inteligente mucho más grande. Incluso se pueden adicionar elementos de domótica, que permitan al usuario controlar e interactuar directamente con los elementos de la red, de esta manera reducir un poco el consumo de energía en la institución, disminuyendo también la energía desperdiciada y reduciendo gastos directos con la compañía prestadora del servicio.

Referencias

- [1] eSMARTCITY, “Micro redes eléctricas inteligentes.” <https://www.esmartcity.es/2011/03/08/micro-redes-electricas-inteligentes>, 2011.
- [2] M. L. G., “La micro-red inteligente: Una ciudad eficiente, en miniatura,” *Universitas Científica*, vol. 18, no. 1, pp. 24–29, 2015.
- [3] C. N. de Energías Renovables CENER, “Introducción a las microrredes.” <http://www.cener.com/introduccion-a-las-microrredes/>, 2017.
- [4] E. S. B. Milton Leandro Becerra Hernandez, *Integración de medidores electrónicos y electromecánicos al sistema AMI cumpliendo con los requerimientos de EPSA ESP*. Universidad de San Buenaventura, 2011.
- [5] Structuralia, “El papel de los smart meters o contadores inteligentes en el ahorro energético.” <https://blog.structuralia.com/el-papel-de-los-smart-meters-o-contadores-inteligentes-en-el-ahorro-energetico>, 05 de marzo del 2018.
- [6] eGEO, “Iot energy efficiency.” <https://www.egeo.co/features>, 2018.
- [7] eGEO, “Hardware.” <https://www.egeo.co/wiki/getting-started/hardware/>, 2018.
- [8] S. S. de medición y control, “Sensores efecto hall.” <https://www.siriored.com/efecto-hall.html>, 2021.
- [9] L. HEYI Electrical Co, “Kct series split core current transformer.” <http://www.heyiele.com/Catalogue/KCT-split-core-current-transformer-113.html>, 2019.
- [10] ONION, “The command line interface.” <https://docs.onion.io/omega2-docs/the-command-line-interface.html>, 2018.
- [11] ONION-DOCS, “Onion omega2+.” <https://docs.onion.io/omega2-docs/omega2p.html>, 2018.
- [12] G. Navarro, “Las fallas eléctricas más comunes y peligrosas.” <https://gruponavarro.pe/electricidad-domiciliaria/fallas-electricas/>, 2021.
- [13] D. H. Santana, *Fallas en las instalaciones eléctricas por calidad de la energía*. DHS Ingeniería, 2010.

- [14] IEEE SA, “3002.3-2018 - ieee recommended practice for conducting short-circuit studies and analysis of industrial and commercial power systems.” Available: https://standards.ieee.org/standard/3002_3-2018.html, 2018.
- [15] J. Das, *Understanding Symmetrical Components for Power System Modeling*. IEEE Press Series on Power Engineering, Wiley, 2016.
- [16] G. R. Barai, S. Krishnan, and B. Venkatesh, “Smart metering and functionalities of smart meters in smart grid - a review,” in *2015 IEEE Electrical Power and Energy Conference (EPEC)*, pp. 138–145, Oct 2015.
- [17] A. S. Metering, S. Visalatchi, and K. K. Sandeep, “Smart energy metering and power theft control using arduino gsm,” in *2017 2nd International Conference for Convergence in Technology (I2CT)*, pp. 858–961, April 2017.
- [18] B. K. Barman, S. N. Yadav, S. Kumar, and S. Gope, “Iot based smart energy meter for efficient energy utilization in smart grid,” in *2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)*, pp. 1–5, June 2018.
- [19] L. Susanti, D. Fatrias, D. Ichwana, H. Kamil, and M. V. Putri, “A configuration system for real-time monitoring and controlling electricity consumption behavior,” in *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 442–447, Oct 2018.
- [20] B. B. Huang, G. H. Xie, W. Z. Kong, and Q. H. Li, “Study on smart grid and key technology system to promote the development of distributed generation,” in *IEEE PES Innovative Smart Grid Technologies*, pp. 1–4, May 2012.
- [21] A. Singhal and R. P. Saxena, “Software models for smart grid,” in *2012 First International Workshop on Software Engineering Challenges for the Smart Grid (SE-SmartGrids)*, pp. 42–45, June 2012.
- [22] J. Liu, B. Zhao, J. Wang, Y. Zhu, and J. Hu, “Application of power line communication in smart power consumption,” in *ISPLC2010*, pp. 303–307, March 2010.
- [23] N. Anku, J. Abayatye, and S. Oguah, “Smart grid: An assessment of opportunities and challenges in its deployment in the ghana power system,” in *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, Feb 2013.