

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas  
Carrera de Ingeniería en Mantenimiento Eléctrico

**MONITOREO DEL ESTADO ACTUAL DE UN SISTEMA ELÉCTRICO DE  
POTENCIA DE LABORATORIO**

Trabajo de grado presentado ante la Universidad Técnica del Norte previo a la  
obtención del título de Ingeniero en Mantenimiento Eléctrico

**AUTOR:**

Jonathan Fabricio Suárez Peralta

**DIRECTOR:**

Ing. Jhonny Javier Barzola Iza MSc.

Ibarra – Ecuador

2023



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN  
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

### IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO		
<b>CÉDULA DE IDENTIDAD:</b>	1003741665	
<b>APELLIDOS Y NOMBRES:</b>	Suárez Peralta Jonathan Fabricio	
<b>DIRECCIÓN:</b>	Padre Pio y Espinoza de los Monteros	
<b>EMAIL:</b>	<a href="mailto:jfsuarezp@utn.edu.ec">jfsuarezp@utn.edu.ec</a>	
<b>TELÉFONO FIJO:</b>	<b>TELÉFONO MÓVIL:</b>	+593986911283

DATOS DE LA OBRA	
<b>TÍTULO:</b>	Monitoreo del estado actual de un sistema eléctrico de potencia de laboratorio
<b>AUTOR:</b>	Jonathan Fabricio Suárez Peralta
<b>FECHA: DD/MM/AA</b>	28/02/2023
<b>SOLO PARA TRABAJOS DE GRADO</b>	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSTGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniería en mantenimiento eléctrico
<b>ASESOR / DIRECTOR</b>	Ing. Jhonny Javier Barzola Iza MSc.

## CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor a terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que se asume responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de marzo de 2023

EL AUTOR



---

Jonathan Fabricio Suárez Peralta

Cédula: 100374166-5

# CERTIFICACIÓN DE TRABAJO DE GRADO

## ACEPTACIÓN DEL DIRECTOR

En mi calidad de director del plan de trabajo de grado, previo a la obtención del título de Ingeniería en Mantenimiento Eléctrico, nombrado por el Honorable Consejo Directivo de la Facultad de Ingeniería en Ciencias Aplicadas.

### CERTIFICO:

Que una vez analizado el plan de grado cuyo título es **"MONITOREO DEL ESTADO ACTUAL DE UN SISTEMA ELÉCTRICO DE POTENCIA DE LABORATORIO"** presentado por el señor: **JONATHAN FABRICIO SUÁREZ PERALTA** con número de cédula 1003741665, doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte de los señores integrantes del jurado examinador que se designe.

En la ciudad de Ibarra, a los 28 días del mes febrero del 2023

Atentamente:



Ing. Jhonny Javier Barzola Iza MSc.

DIRECTOR DEL TRABAJO DE GRADO

# DEDICATORIA

Le dedico este trabajo de tesis a dos personas quienes fueron mi inspiración, me brindaron su apoyo incondicional, siempre velando por mi felicidad con un inmenso cariño y un gran corazón, que me ayudaron cuando más necesite y a pesar de los tropiezos en mi vida siguen creyendo en mí, gracias, padre el Sr. Euler Suárez y mi madre la Sr. Margarita Peralta esto es por y para ustedes.

A mis hermanos Daniela, Jhoel y José, por todos sus consejos, palabras de aliento en cada momento, convirtiéndose en mis confidente y apoyo moral fundamental durante el transcurso de mi carrera.

A mi hijo Iker, siendo mi fuente de inspiración diaria, mi motor y mi paño de lágrimas, lo logramos hijo salimos adelante a pesar de todas las adversidades y problemas en la vida, gracias por llegar a mi vida y darme una razón de luchar incondicionalmente.

# AGRADECIMIENTO

Agradezco a Dios por permitirme lograr un sueño y una meta que tenía en mi vida, a mi familia quienes confiaron y estuvieron incondicionalmente.

Le agradezco a mi tutor, Ing. Jhonny Barzola MSc, por su paciencia, guía y confianza, y mi asesor, Ing. Hernán Pérez MSc, que con su predisposición y tiempo dedicado a la guía de mi trabajo de grado pude llegar a culminarlo. A todos los docentes quienes me brindaron los conocimientos necesarios, durante mi periodo académico.

Agradecerles a todos mis amigos quienes se convirtieron en mis cómplices y hermanos. Gracias por todos los momentos de alegrías y apoyo moral que me brindan incondicionalmente.

Agradecer a la Universidad Técnica del Norte por formarme académicamente para poder ser un profesional con valores.

# TABLA DE CONTENIDO

IDENTIFICACIÓN DE LA OBRA .....	II
CONSTANCIA .....	III
CERTIFICACIÓN DE TRABAJO DE GRADO .....	IV
DEDICATORIA .....	V
AGRADECIMIENTO .....	VI
TABLA DE CONTENIDO .....	VII
ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABLAS .....	X
RESUMEN .....	XI
ABSTRACT .....	XII
INTRODUCCIÓN .....	1
A.1. CONTEXTUALIZACIÓN .....	1
A.2. PLANTEAMIENTO DEL PROBLEMA .....	2
A.3. FORMULACIÓN DEL PROBLEMA .....	2
A.4. OBJETIVO GENERAL .....	3
A.5. OBJETIVOS ESPECÍFICOS .....	3
A.6. JUSTIFICACIÓN .....	3
A.7. ALCANCE .....	3
CAPÍTULO I .....	5
Marco Teórico .....	5
1.1 Sistemas eléctricos de potencia .....	5
1.1.1 Etapas de un sistema eléctrico de potencia .....	5
1.1.2 Variables eléctricas de medición .....	7
1.1.3 Flujos de potencia .....	8
1.1.4 Sistema de transporte de energía eléctrica .....	11
1.2 Monitoreo de sistemas eléctricos .....	12
1.2.1 Estabilidad de un sistema eléctrico de potencia .....	12
1.2.2 Curvas de potencias de voltaje .....	14
1.3 Estado actual de un sistema eléctrico de potencia .....	15
1.4 Partes que conforman un módulo de monitoreo .....	16
1.4.1 Controladores usados para la adquisición de datos .....	17
1.4.2 Sensores para un módulo de monitoreo .....	18
1.4.3 Pantalla LCD para la visualización de datos del módulo de monitoreo .....	21
1.4.4 Entorno de desarrollo Integrado de Arduino .....	21
CAPÍTULO II .....	23

Diseño del módulo para el monitoreo de variables de estado de un sistema eléctrico de potencia .....	23
2.1 Metodología.....	23
2.2 Parámetros para el monitoreo del sistema eléctrico de potencia .....	24
2.3 Selección de componentes.....	24
2.3.1 Controladores para el monitoreo .....	24
2.3.2 Sensores para el monitoreo.....	25
2.3.3 Conversor para el monitoreo .....	26
2.3.4 Pantalla LCD para el monitoreo.....	26
2.3.5 Componentes seleccionados para el módulo .....	27
2.4 Flujograma del proceso del módulo .....	28
2.5 Diagrama esquemático de las conexiones del módulo .....	29
2.6 Diseño electrónico del módulo del sistema eléctrico de potencia.....	30
2.7 Estimación de estado de un sistema de potencia .....	32
CAPITULO III.....	39
Implementación del módulo para monitoreo de variables de estado de un sistema eléctrico de potencia .....	39
3.1 Ensamblaje del módulo.....	39
3.1.1 Fijación de los componentes eléctricos del sistema .....	40
3.1.2 Fijación de los componentes electrónicos del sistema.....	41
3.2 Creación de la interfaz SCADA .....	44
3.3 Adquisición de datos de todos los parámetros medidos .....	46
3.3.1 Cálculo de ángulos de Desfase .....	48
3.4 Resultados.....	49
CONCLUSIONES .....	58
RECOMENDACIONES .....	59
ANEXOS.....	68
Anexo A: Programación de Pzem-016 para la adquisición de datos .....	68
Anexo B: Programación para HMI.....	77
Anexo C: Programación de cálculo de error.....	98

## ÍNDICE DE FIGURAS

Figura 1. Diagrama unifilar.....	10
Figura 2. Diagrama de transporte de energía.....	11
Figura 3. Monitoreo de un sistema eléctrico.....	12
Figura 4. Curva Q-V.....	14
Figura 5. Curva P-V.....	15
Figura 6. Arduino Mega, Uno, Nano.....	17
Figura 7. Diagrama sensor de corriente SCT.....	19
Figura 8. Sensor de corriente SCT-013.....	20
Figura 9. Sensor PZEM-016.....	20
Figura 10. Pantalla LCD 12x6 y módulo I2C.....	21
Figura 11. Interfaz Arduino IDE.....	22
Figura 12. Interfaz Arduino IDE.....	23
Figura 13. Diagrama de flujo del sistema.....	29
Figura 14. Diagrama de bloques del sistema.....	30
Figura 15. Diseño de conexiones del diagrama electrónico del módulo.....	31
Figura 16. Diagrama unifilar.....	35
Figura 17. Pruebas del funcionamiento de los sensores.....	39
Figura 18. Colocación de barras.....	40
Figura 19. Fijación de los componentes electrónicos usados para la adquisición de datos.....	41
Figura 20. Montaje de controladores Arduino maestro y esclavos.....	42
Figura 21. Montaje de sensores Pzem-016.....	43
Figura 22. Montaje de convertidores.....	43
Figura 23. Colocación de pantallas LCD.....	44
Figura 24. Interfaz del sistema SCADA.....	45
Figura 25. Comprobación de medición y comunicación del sensor.....	46
Figura 26. Visualización de datos en las pantallas.....	47
Figura 27. Conexión para los sensores y demás componentes.....	47
Figura 28. Comprobación de valores en HMI.....	48
Figura 29. Código de creación del histórico de datos.....	50
Figura 30. Valores obtenidos para el histórico de datos.....	51
Figura 31. Datos obtenidos de las mediciones.....	51
Figura 32. Valores del histórico de datos.....	52
Figura 33. Transformación a por unidad.....	52
Figura 34. Varianza.....	53
Figura 35. Ecuaciones del sistema.....	53

Figura 36. Matriz jacobiana .....	54
Figura 37. Comparación de error .....	55
Figura 38. Sentencia comparativa.....	55
Figura 39. Nuevas ecuaciones.....	56
Figura 40. Nueva matriz jacobiana.....	56
Figura 41. Resultado obtenido mediante la simulación en Matlab.....	57

## ÍNDICE DE TABLAS

Tabla 1. Tipos de estabilidad de voltaje .....	13
Tabla 2 Características de las tecnologías de Arduino.....	24
Tabla 3. Características de los sensores de medición.....	25
Tabla 4. Características de los conversores.....	26
Tabla 5. Características de los conversores.....	26
Tabla 6. contenido de materiales usados en el monitoreo.....	27

## RESUMEN

El monitoreo de un sistema eléctrico de potencia se basa en un proceso de supervisión continua de las diferentes variables eléctricas, con la finalidad de garantizar la fiabilidad y eficiencia del suministro de energía eléctrica. El análisis del comportamiento de un sistema eléctrico de potencia es fundamental para estimar el rango moderado de operación. Para ello se debe conocer los valores tales como corriente, voltaje, potencia, factor de potencia, frecuencia de funcionamiento en tiempo real, y lograr determinar anomalías durante el tiempo de funcionamiento. El objetivo de realizar la estimación del estado actual del sistema eléctrico de potencia es comprobar el óptimo funcionamiento del sistema, mediante el monitoreo local y remoto recolectando valores medidos, por medio de sensores específicamente usados para este tipo de sistemas y así contar con la mayor cantidad de mediciones en las diferentes barras. Almacenándolas en un histórico de datos para llegar a una estimación del sistema comprando los valores almacenados con los bases que tiene en todo el proceso. Llegando a encontrar errores en las mediciones, mediante un algoritmo de Matlab que ayuda a detectar estas eventualidades, y así poder dar una solución o sugerencias para tener un sistema en óptimas condiciones de operación.

**Palabras clave:** Sistema eléctrico de potencia, variables, monitoreo local, monitoreo remoto, error en las mediciones.

## **ABSTRACT**

The monitoring of an electrical power system is based on a process of continuous supervision of the different electrical variables, with the purpose of guaranteeing the reliability and efficiency of the electrical energy supply. The analysis of the behavior of an electrical power system is fundamental to estimate the moderate range of operation. For this purpose, values such as current, voltage, power, power factor, operating frequency must be known in real time, and anomalies must be determined during the operating time. The objective of estimating the current state of the electrical power system is to verify the optimal operation of the system, through local and remote monitoring by collecting measured values, by means of sensors specifically used for this type of systems and thus having the largest number of measurements in the different busbars. Storing them in a historical data to arrive at an estimate of the system by comparing the stored values with the bases it has throughout the process. Getting to find errors in the measurements, by means of an algorithm of Matlab that helps to detect these eventualities, and thus to be able to give a solution or suggestions to have a system in optimal conditions of operation.

**Key words:** Electrical power system, variables, local monitoring, remote monitoring, measurement errors.

# INTRODUCCIÓN

## A.1. CONTEXTUALIZACIÓN

Los Sistemas Eléctricos de Potencia forman parte esencial en lo que respecta al campo de la ingeniería eléctrica siendo así esta una de las opciones para llegar a nuevos estudios, ya que en la actualidad se usan nuevas tecnologías para mejorar y crear energía eléctrica de una manera más eficiente y usando tecnologías renovables para ellos se ve necesario realizar un análisis de flujos de potencia en el sistema eléctrico.

El estudio de los flujos de potencia es relevante para el análisis y la implementación de sistemas de potencia además de lograr una correcta operación de estos según mencionan Changotasig & Beltrán (2017) que para ello se debe tomar cuenta la información que se va a obtener ya sea el voltaje de fase, potencia activa y reactiva de cada una de las líneas que se tenga conectadas a la red.

El llegar a controlar un proceso de pequeña o gran magnitud, se requiere el uso de sistemas de comunicación entre varios dispositivos ya sean estos sensores tales como PLC (Controlador Lógico Programable), entre otros estos pueden llegar a ser digitales o analógicos ya sea para la extracción de datos o a su vez para mando a control automatizado (Vilañez & Alban, 2015).

La forma de recolección de datos en la actualidad ha hecho que el operador pase a ser un analista de información, con ello se pueda adaptar a nuevas formas de operación en sistema en tiempo real (Gallegos, 2017).

En la estimación de estado, se debe mencionar que los dispositivos de medición pueden proporcionar información la cual llega a tener errores ya que cada uno de estos dispositivos introduce un error aleatorio, para ello se debe implementar un mecanismo el cual llegue a mitigar este error en estado estable (Vaahedi, 2014).

Si bien es cierto además de lo SEP, los sistemas SCADA (Supervisión, Control y Adquisición de Datos) también son parte fundamental en el campo eléctrico, ya que este tipo de tecnologías nos ayudan a facilitar procesos industriales y su respectivo monitoreo el cual puede ser las 24

horas del día obteniendo datos relevantes a cada momento, en este caso nos ayuda a optimizar procesos de control en un sistema eléctrico de potencia específicamente en la inyección de potencias activas y reactivas de un generador aun sistema de potencia (Torres & Rivera, 2017).

## **A.2. PLANTEAMIENTO DEL PROBLEMA**

Las centrales y pequeñas generadoras, así como subsistemas, cuentan con la información de parámetros de funcionamiento en tiempo real de forma local. Esto, provoca la necesidad de monitorear todo el sistema de potencia de forma remota con el objetivo de conocer la situación actual del mismo.

Aplicar un monitoreo a estos sistemas eléctricos de potencia es complicado, ya que el acceso a las centrales generadoras es muy limitado solo se puede acceder a visitas técnicas. Estos casos son limitados a casos de estudio o mediante pruebas realizadas en laboratorio, sin embargo, la manipulación de estos equipos es de vital importancia para entender su comportamiento y resolución de problemas durante operación.

El análisis del estado actual de un sistema de potencia es de gran importancia en la planeación y en el diseño de la expansión futura de los sistemas eléctricos, así como en la determinación de las mejores condiciones de operación de los sistemas existentes.

Siendo así el monitoreo de un sistema eléctrico de potencia a escala el encargado de verificar los voltajes en el sistema como también las potencias lo cual corroborará el funcionamiento normal y la operación dentro de los límites establecidos.

## **A.3. FORMULACIÓN DEL PROBLEMA**

¿Cómo monitorear un sistema eléctrico de potencia de laboratorio para la estimación de su estado actual?

#### **A.4. OBJETIVO GENERAL**

Monitorear un sistema eléctrico de potencia de laboratorio para la estimación de su estado actual en la carrera de Electricidad.

#### **A.5. OBJETIVOS ESPECÍFICOS**

- a) Analizar las técnicas y parámetros de monitorización de un sistema eléctrico de potencia.
- b) Determinar las variables de estado y barras que intervienen en el monitoreo de un sistema eléctrico de potencia de laboratorio.
- c) Monitorear el estado actual del sistema eléctrico de potencia de laboratorio a través de mediciones realizadas usando Arduino y programaciones en Matlab.

#### **A.6. JUSTIFICACIÓN**

El incremento de la demanda de energía eléctrica, la expansión del sistema eléctrico nacional y la salida repentina de elementos e incidentes en el sistema vuelve apropiado el análisis del comportamiento de un sistema eléctrico de potencia.

Para realizar una estimación del rango moderado de operación de un sistema eléctrico de potencia o cualquier estudio posterior, sea de confiabilidad, calidad de servicio o estabilidad, se requiere conocer el funcionamiento del sistema y su estado actual. Para esto, es importante la obtención de valores en tiempo real de corriente, voltaje y factor de potencia.

Mediante el estado actual de un sistema de potencia podemos analizar los datos obtenidos en las diferentes barras conectadas para conocer si el sistema está funcionando correctamente o cuenta con fallas para ello se comparará los valores medidos.

#### **A.7. ALCANCE**

La finalidad del proyecto es monitorear las variables de interés dentro del sistema de potencia de forma remota como lo son el voltaje, potencia y corriente en cada barra. Esto, se realizará por

medio de un sistema de adquisición de datos a las diferentes barras del sistema. Además, se desarrollará una guía de prácticas para el análisis de sistemas de potencia y un almacenamiento de datos histórico de funcionamiento.

Desarrollar el monitoreo de un módulo de sistema eléctrico de potencia a escala y proceder al análisis de datos de medición de este de forma remota abrirá la posibilidad del desarrollo de prácticas de laboratorio y estudios posteriores relacionados con el análisis de sistemas de potencia, como son la operación y estabilidad del sistema.

El análisis se realizará a un sistema eléctrico de potencia de laboratorio de 4 barras monofásico alimentado con 120 voltios, en el cual se permitirá estimar el estado actual del mismo, a través de la recopilación de datos tanto de potencia, corriente y voltaje eficaces. Estos datos serán medidos de forma local mediante visualizadores en las diferentes barras y de manera remota mediante sensores que estarán conectados a un Arduino Mega 2560 en comunicación con un computador para posterior utilización del usuario.

La forma en la cual se realizará la recopilación de estos será obtener las mediciones durante un lapso ajustable para obtener un estimado del estado del sistema

# **CAPÍTULO I**

## **Marco Teórico**

El presente trabajo de grado se refiere al monitoreo del estado actual de un sistema eléctrico de potencia de laboratorio en el que se monitorea las variables voltaje, corriente y potencia en cada barra además de los componentes que ayudan a la adquisición de datos de estos parámetros.

### **1.1 Sistemas eléctricos de potencia**

Los sistemas eléctricos de potencia cuentan con tres etapas la cual se puede encontrar dentro de la naturaleza diferentes tipos de energía primaria, para posterior convertirle en electricidad, mediante un generador eléctrico, una vez realizado este proceso se llega a la transmisión y distribución a los diferentes usuarios conectados a la red. Por ello se debe tomar en consideración todo lo que respecta el monitoreo de este sistema desde su generación hasta llegar al consumidor, para lograr que la eficiencia energética que llega a los clientes sea la adecuada evitando con la suspensión del servicio (Carakenio73, 2019).

#### **1.1.1 Etapas de un sistema eléctrico de potencia**

Un sistema eléctrico de potencia consta de 3 tipos de subsistemas los cuales son generación, transmisión, y carga.

##### **a) Generación**

La etapa de generación implica la obtención de energía eléctrica a través del uso de centrales eléctricas, las cuales utilizan fuentes de energía primaria disponibles en la naturaleza, como la energía solar, eólica, mareomotriz, hidráulica, biomasa y térmica. En esta etapa se lleva a cabo el proceso de transformación de la energía primaria en energía eléctrica mediante el uso de tecnologías como turbinas, generadores y paneles solares. Esta etapa es crucial para el suministro de energía eléctrica a los consumidores finales y es una de las primeras etapas en la cadena de suministro de electricidad.

Uno de los componentes comúnmente utilizados para generar energía eléctrica en las centrales eléctricas son los generadores síncronos de imanes permanentes y los generadores asíncronos de inducción.

Los generadores síncronos de imanes permanentes utilizan imanes permanentes para producir un campo magnético que gira dentro de una bobina, generando así corriente eléctrica. Estos generadores son altamente eficientes (Watt, 2020).

Al momento de la excitación del generador se mantiene el voltaje y el control de los flujos de potencia en este caso potencia reactiva del generador causando que aumente el voltaje (Postes & Rendon, 2019).

## **b) Transmisión**

La etapa de transmisión se encarga del traslado de la energía eléctrica desde las centrales eléctricas hasta los puntos de distribución, los cuales pueden ser estaciones transformadoras, subestaciones o redes de distribución. En esta etapa, la electricidad se desplaza mediante líneas de transmisión de alta tensión, las cuales pueden abarcar extensas distancias, que van desde cientos de kilómetros hasta miles de kilómetros.

Esta energía transportada antes de llegar a los consumidores finales pasa por diferentes niveles de voltaje, por lo que se reduce este nivel por medio de un transformador el cual distribuye a un sector específico por medio de los medidores ya sean residenciales o comerciales (Chiguano, 2018).

## **c) Carga**

La carga tiene características entre estas el factor de demanda, factor de carga, factor de diversidad, factor de utilización, todo esto conlleva a que se pueda dividir en categorías tales como:

- a) Residencial.
- b) Comercial

- c) Industrial
- d) Alumbrado Público

En donde la carga residencial se basa primordialmente en todos los aparatos electrónicos que se cuenta en el domicilio y en donde el consumo de estos es mínimo como pueden ser los siguientes.

- a) Hornos.
- b) Refrigeradores.
- c) Batidoras.
- d) Licuadoras.
- e) Duchas.

Mientras que la carga comercial se basa primordialmente en consumos que se encuentran dentro del rango moderado, es decir no sobrepasa la tarifa para colocar un transformador, la carga industrial es aquellas que cuentan con equipos de mayor consumo por ello exceden de la carga establecida por lo que necesitan muchas veces de un transformador independiente según el consumo que estas realices (Pilatásig, 2020).

### **1.1.2 Variables eléctricas de medición**

Las diferentes variables de medición que se obtendrán del sistema eléctrico de potencia son voltaje, corriente y potencia tomando en cuenta que estos son los valores bases para la obtención de datos.

#### **a) Voltaje**

El voltaje, también conocido como tensión eléctrica, es una medida de la diferencia de potencial eléctrico entre dos puntos en un sistema eléctrico de potencia. En términos simples, el voltaje indica la fuerza con la que la electricidad fluye a través del sistema.

El voltaje, también conocido como tensión eléctrica, es un parámetro crítico en un sistema eléctrico de potencia que debe ser cuidadosamente monitoreado y controlado

para garantizar la confiabilidad y seguridad del sistema. Las fluctuaciones o cambios en el voltaje pueden ocasionar problemas como fallas en el equipo, interrupciones en el suministro eléctrico, y en situaciones extremas, daños a los dispositivos eléctricos y peligros para las personas (Oleas, 2019).

## **b) Corriente**

La corriente es un parámetro fundamental en los sistemas eléctricos de potencia, ya que es la responsable de transportar la energía eléctrica a los dispositivos y equipos que la requieren para operar. La capacidad de un circuito para transportar corriente está limitada por la sección del conductor y la capacidad de los dispositivos de protección del sistema. Es crucial mantener la corriente dentro de un rango seguro para prevenir sobrecargas en los dispositivos y equipos eléctricos, así como para evitar situaciones peligrosas para las personas (Postes & Rendon, 2019).

## **c) Potencia**

La potencia eléctrica es la demanda que de energía eléctrica que necesita o absorbe un equipo para su funcionamiento, esta está relacionada con lo es corriente ( $I$ ) y voltaje ( $v$ ), (Amaguaña, 2021).

### **1.1.3 Flujos de potencia**

Es la potencia que se trasmite en un sistema eléctrico, puede encontrar ya sea en una instalación a gran o menor escala, un flujo de potencia da como resultado parámetros de la red como son los voltajes y ángulos de las barras de conexión y las potencias activas y reactivas que se transmiten por el sistema eléctrico (Duncan, Jaramillo, & Gomez, 2017).

Las razones por las cuales llevan a cabo un estudio flujo de potencia son los siguientes:

- a) Flujo en la línea.
- b) Voltajes y sus ángulos.

- c) El agregar nuevos circuitos conectados al sistema de carga.
- d) Despacho económico.
- e) Perdidas en el sistema.
- f) Transformadores de funcionamiento económico.
- g) Cambio en el tamaño de los conductores y repotenciaciones.

La mayoría de los estudios de los flujos de potencia, se llegan a usar en una red monofásica, ya que este sistema esta equilibrado y las redes se simulan red para soluciones de flujo estable.

Una ayuda puntual para el estudio de estos flujos de potencia son los computadores los cuales actualmente poseen gran velocidad y almacenamiento para las simulaciones que se realizan del sistema. En lo que respecta a las ecuaciones de red se toma en cuenta método de nodo voltaje esto sucede cuando las admitancias nodales obteniendo ecuaciones simultaneas en términos nodales o de bus.

#### a) Análisis de flujos de potencia

El Análisis de estos flujos por lo general se lo realiza en estado estacionario para ellos se usa un diagrama unifilar como se muestra en a Figura1, para lo cual se debe tener los datos reales que se cuentan con el sistema para el análisis numérico de estos flujos (Tiberio, Jaime, Luis, & Victor, 2017).

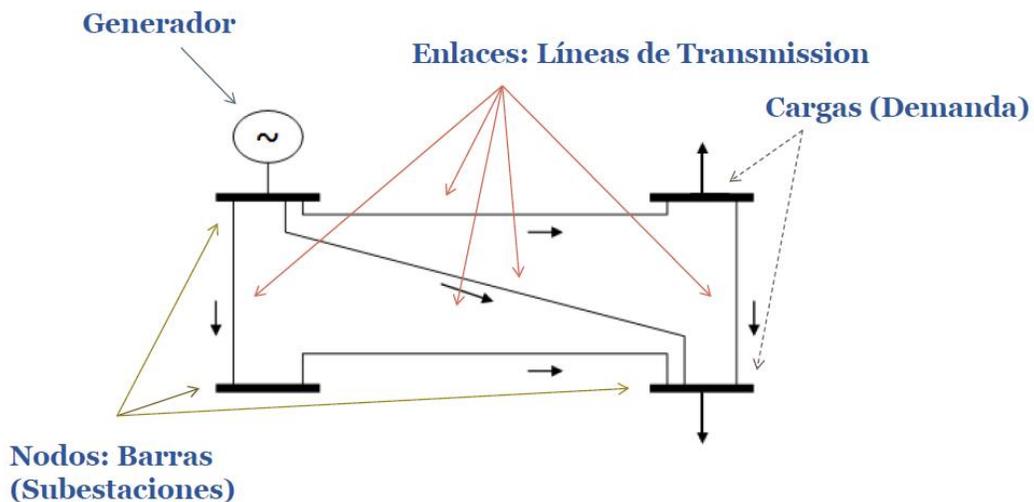


Figura 1. Diagrama unifilar

Fuente: (Tiberio, Jaime, Luis, & Víctor, 2017)

Un flujo de potencia implica encontrar la solución a una serie de ecuaciones simultáneas no lineales, y que tienen las relaciones, entre el voltaje y la corriente en cada barra, entre el requerimiento de potencia activa y reactiva en una barra de carga o la capacidad que tiene de potencia generada y en el tamaño de voltaje especificado en una barra de generación (Tenelema, 2019).

## **b) Criterios de flujos de potencia**

Para operar un SEP (Sistema Eléctrico de Potencia) en estado estable hay que considerar algunos criterios.

- a) La generación debe abastecer a toda la demanda más las pérdidas de potencia que se generan en las líneas de transmisión por efecto Joule.
- b) Los valores de voltaje en las barras deben permanecer dentro del rango valor nominal.
- c) Los generadores no deben operar fuera de los límites especificados de potencia activa y reactiva.
- d) Los transformadores y las líneas de transmisión deben operar dentro de sus límites, es decir no tienen que estar sobrecargados. (Duncan & Mulukutla, 2012).

El estudio de flujos de carga es de gran interés en la planeación de la red eléctrica y en el diseño de futuras expansiones de los sistemas eléctricos, también en la mejora de las circunstancias de operación de los sistemas eléctricos ya existentes. Este estudio se lo realiza principalmente en la red de transmisión, ya que para el análisis se necesita considerar un sistema que esté operando de forma estable y balanceado. En aquel momento para el análisis de flujos se debe tener en cuenta que las cargas del sistema

son de valores constantes. El planteamiento de las ecuaciones de flujos de potencia se hace de acuerdo con tres tipos de barras del sistema, para las que se especifican valores iniciales que son barras de carga PQ, barras de voltaje controlado PV y barras de compensación SLACK (Tenelema, 2019).

#### 1.1.4 Sistema de transporte de energía eléctrica

Un sistema de transporte de energía eléctrica lo comprenden las centrales de generación eléctrica, quienes producen la energía a un voltaje de 3 a 36 kV, las redes de transmisión a un nivel de voltaje de 138, 230 y 500 kV, redes de subtransmisión a un nivel de voltaje de 69 y 34,5 kV, y redes de distribución de 13,8, 22, 6.3 kV. En la Figura 2 se observa la estructura de transporte de energía eléctrica (Walter Brokering, 2008).

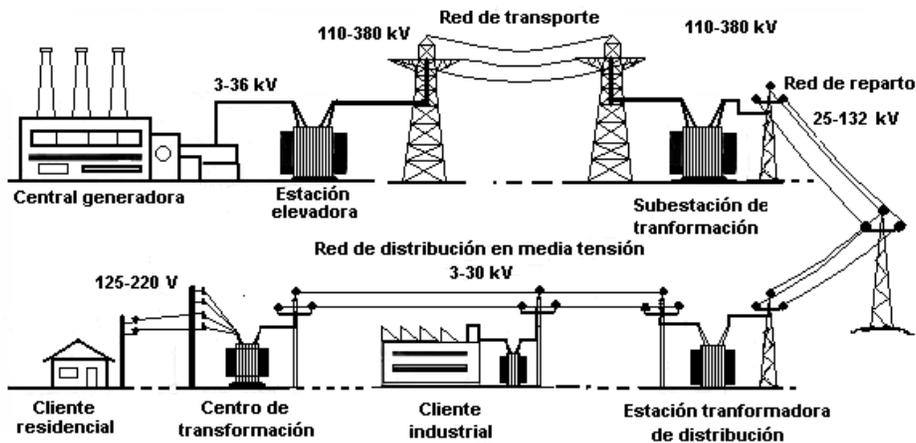


Figura 2. Diagrama de transporte de energía

Fuente: (Walter Brokering, 2008)

## 1.2 Monitoreo de sistemas eléctricos

El monitoreo de un sistema eléctrico de potencia se basa en la recolección de datos como se muestra en la Figura 3, en donde una vez obtenidos los datos del sistema se analiza la información obtenida y se la usa para tener el monitoreo constante y evitar fallas dentro del sistema; estos datos se los puede observar mediante pantallas o a su vez monitores integrados este tipo de monitoreo se los realiza de forma remota en el campo (Landy, 2015).

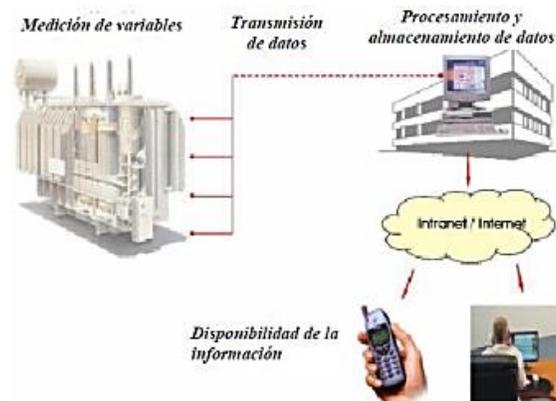


Figura 3. Monitoreo de un sistema eléctrico

Fuente: (Costa, Martinez, Alves, & Araujo)

### 1.2.1 Estabilidad de un sistema eléctrico de potencia

Con lo que respecta a la estabilidad se refiere a su capacidad de mantenerse en un estado de operación estable y equilibrado, cumpliendo con las normas estándares admisibles durante su funcionamiento normal. Esta propiedad es esencial para garantizar un funcionamiento confiable y seguro del sistema eléctrico de potencia y evitar situaciones peligrosas para las personas y los equipos.

Para llegar la evaluación de la estabilidad, se da en el interés del comportamiento de este sistema cuando este sujeto a alguna perturbación ya sea pequeña o grandes. Estas estabilidades se clasifican ante pequeños, grandes perturbaciones de corto plazo y largo plazo los que se detallan en la Tabla 1.

Tabla 1. Tipos de estabilidad de voltaje

<b>Estabilidad</b>	<b>Definición</b>
<b>Pequeños disturbios</b>	Esto se define con la capacidad de mantener al sistema con voltaje constante después de pequeños disturbios como por ejemplo los incrementos en la carga.
<b>Grandes disturbios</b>	Esto se define con la capacidad de mantener al sistema con voltaje constante después de grandes disturbios como por ejemplo fallas en el sistema de transmisión, pérdidas de generación o cortocircuitos.
<b>Corto plazo</b>	Esta estabilidad involucra la dinámica en el tiempo de los elementos rápidos de la carga, este periodo es en el orden de varios segundos, y para analizarle es en función de ecuaciones diferenciales del sistema.
<b>Largo plazo</b>	Esta estabilidad involucra la dinámica en el tiempo de los elementos lentos tales como limitaciones de la corriente del generador, cargas controladas por temperatura. Este periodo es se puede extender a varios minutos, y para analizarle se requiere simulaciones a largo plazo.

**Fuente:** Tomado de (Patiño & Limas, 2018)

Las inestabilidades del sistema se pueden dar por varias formas, pudiendo ser influenciados por un amplio rango de factores, por lo cual se ha llegado a clasificar entre apropiadas con el fin de facilitar sus análisis, además de identificar factores los cuales contribuyen a las pérdidas y también a formar métodos para mejorar la operación estable, los aspectos a considerar son los siguientes: (Tapia, 2015)

- a) Naturaleza física de la inestabilidad resultante.
- b) Tamaño de la perturbación considerada.
- c) Dispositivos, procesos y espacio de tiempo que debe ser tomado en consideración para determinar estabilidad.
- d) El más apropiado método de cálculo y predicción de estabilidad.

## 1.2.2 Curvas de potencias de voltaje

Las curvas de potencia son una herramienta fundamental en los sistemas eléctricos de potencia, ya que permiten garantizar el correcto funcionamiento de los equipos y asegurar la transmisión adecuada de la potencia eléctrica. La potencia eléctrica es una variable crítica que afecta directamente el voltaje en los sistemas eléctricos de potencia, por lo que resulta indispensable contar con herramientas que permitan su monitoreo y control. En este sentido, existen dos métodos principales para la representación de las curvas de potencia: el método de las curvas Q-V y el método de las curvas P-V. Ambos métodos permiten obtener información relevante sobre el comportamiento de los sistemas eléctricos de potencia y resultan fundamentales para la toma de decisiones en el diseño, operación y mantenimiento de los mismos.

### a) Método de las curvas Q-V

El método de las curvas Q-V muestra el colapso del voltaje el cual nos indica que hay un balance inadecuado de potencia reactiva. El método consiste en aumentar el consumo, la potencia reactiva en cada barra de la red hasta colapsar el voltaje, este método gráfico indica la cantidad de potencia reactiva que debe ser inyectada a una barra para mantener un nivel de voltaje deseado esto se representa en la Figura 4. (Patiño & Limas, 2018)

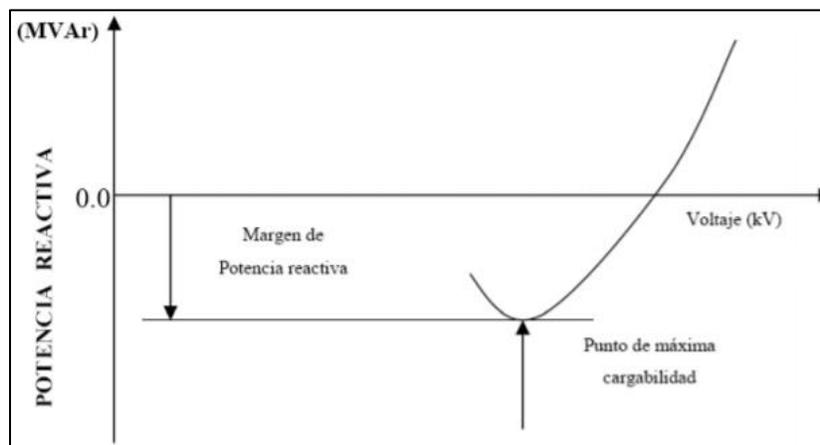


Figura 4. Curva Q-V

Fuente: (Patiño & Limas, 2018)

## b) Método de las curvas P-V

El método de la curva P-V simula aumentos de cargas de muchas formas esto dependerá de que barra se elija para aumentar la carga activa o reactiva esto no es más que elegir un vector en el espacio multidimensional de la red para realizar los cálculos. La curva P-V que se observa en la Figura 5 indica cuando hay un aumento de potencia activa transmitida mientras que los voltajes decaen lentamente.

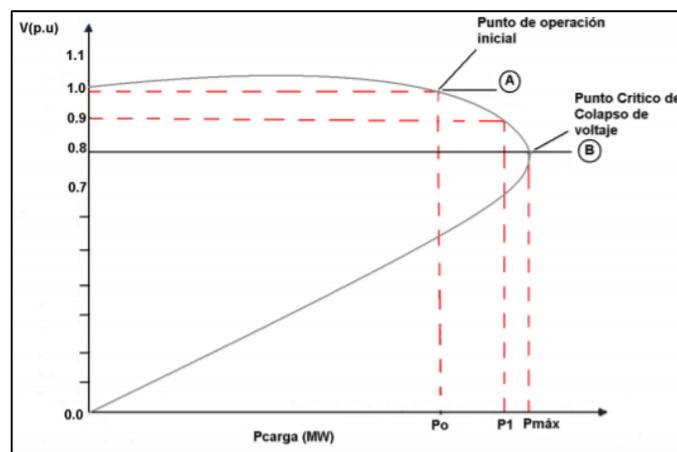


Figura 5. Curva P-V

Fuente: Tomado de (Patiño & Limas, 2018)

En la Figura 5 también se observa el punto crítico el cuál es el límite entre la estabilidad y la inestabilidad de voltaje esto quiere decir que si el punto de operación se encuentra ubicado en “A” hay un pequeño incremento en la transferencia de potencia activa y esto lleva a una pequeña disminución del voltaje. Para el otro caso ocurre lo contrario si la operación cambia del punto inicial “A” al punto crítico “B” hay un incremento de potencia lo que lleva una gran disminución de voltaje y por último si la potencia transferida sobrepase el punto “B” esto provocaría una disminución drástica en los voltajes y en la potencia transferida (Patiño & Limas, 2018).

### 1.3 Estado actual de un sistema eléctrico de potencia

La finalidad de llegar a obtener la información del estado actual del sistema por medio de sensores, dispositivos de adquisición de datos, software para la obtención de mediciones y evaluación del sistema es útil ya que se puede llevar los datos obtenidos durante todo el proceso del sistema, manteniendo así de manera simultánea la veracidad de información y la seguridad de los datos del sistema, para ello el método de los mínimos cuadrados ayuda en el cálculo de la estimación del estado de los sistemas eléctricos de potencia (Viscaino, 2015).

#### **a) Método de mínimos cuadrados para la estimación de estados**

Para la estimación de estados en los sistemas eléctricos de potencia debe tener como referencia el concepto en donde se debe asignar un valor a una variable de estado, esta variable debe ser de un sistema ya conocido para poder basarse en dichas medidas partiendo de este sistema.

Generalmente los sistemas nos envían mediciones imperfectas que son muy redundantes, este criterio se basa en un sistema estadísticos el cual estima el valor real de todas las variables involucradas de tal manera que se pueda minimizar el criterio de elección de datos. En los sistemas eléctricos de potencia, se toman generalmente las variables de estado como son la corriente, ángulos de fase en cada uno de los nodos del sistema. (Castro, 2107).

Hoy en día existe gran variedad de dispositivos eléctricos y electrónicos que ayudan a la resolución de cálculos, mediciones de variables como voltaje, corriente, potencia etc. Con el fin de facilitar al usuario el monitoreo de los sistemas eléctricos de potencia, pero esto conlleva elevados costos, sin embargo, hay componentes de fácil adquisición con prestaciones similares a los de gama alta.

### **1.4 Partes que conforman un módulo de monitoreo**

Para monitorear sistemas eléctricos de potencia se tiene en cuenta que estos están conformados por un controlador, sensores, una comunicación, y en algunos casos actuadores.

- a) Controlador: Este sistema es el encargado de realizar los algoritmos para ejecutar las acciones que están programadas por el usuario el cual utiliza los sensores como entradas y los actuadores como salidas, por lo general se utilizan microcontroladores que son programables esto definirá el monitoreo del módulo (Mera & Ruano, 2019).
- b) Sensores: Son los elementos que proporcionan información al sistema de control para que realice las acciones necesarias para ejecutar una orden que pueden ser de tipo digitales y analógicas; por ejemplo, sensores de corriente, voltaje, potencia (Mera & Ruano, 2019).
- c) Comunicación: Los sistemas de comunicación sirven para transmitir datos e interactuar con uno o más dispositivos a la vez permitiendo la interconexión de varios nodos, estas comunicaciones pueden ser por cable o inalámbricas (Mera & Ruano, 2019).

#### 1.4.1 Controladores usados para la adquisición de datos

Hoy en día existe una gran variedad de dispositivos electrónicos programables que cuentan con variedad de microcontroladores, pero se debe tener en cuenta que cada tarjeta sirve para su propósito, entre las plataformas electrónicas más comunes está Arduino la cual tiene varias placas programables como son Arduino Mega, Arduino Uno, Arduino nano como se muestra en la Figura 6, (Torres, 2017).

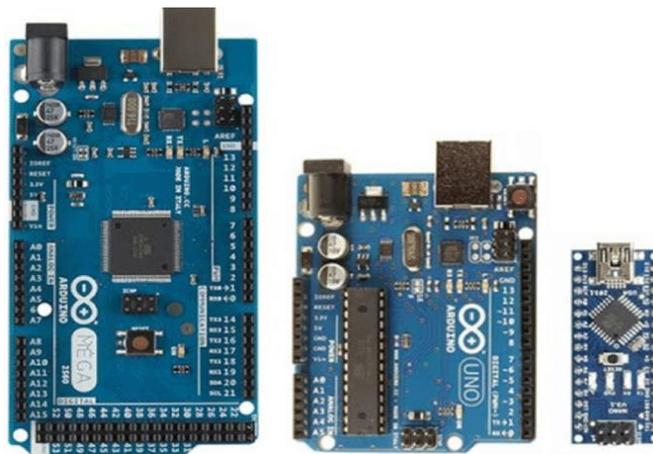


Figura 6. Arduino Mega, Uno, Nano

#### **a) Arduino Mega**

Esta tarjeta de desarrollo está basada en una placa microcontrolador ATmega2560 el cual es utilizado para proyectos en donde se requiera varias entradas y salidas analógicas y digitales ya que este es una de las placas con más entradas digitales. Por lo que se utiliza en procesos más complejos. El Arduino Mega ya viene cargado un gestor de arranque lo que permite cargar códigos sin necesidad de un programador por hardware externo. (Arduino, 2022)

#### **b) Arduino Uno**

Esta tarjeta de desarrollo es la más utilizada para programar, se usa en la mayoría de los trabajos ya que dispone de entradas y salidas analógicas y digitales suficientes como para proyectos complejos. Su modelo es compacto no ocupa mucho tamaño y usa un microcontrolador ATmega328P el cual consta de 14 pines digitales y 6 análogos. (Arduino, 2022)

#### **c) Arduino Nano**

La tarjeta de desarrollo Arduino Nano es una de las pequeñas placas que se basa en el ATmega328, este componente electrónico se utiliza por personas que inician en el campo de la programación electrónica es un modelo compacto el cual es utilizada para proyectos pequeños o placas que necesiten poco espacio posee voltajes de 5 y 3,3 voltios suficientes para proyectos básicos. (Arduino, 2022)

### **1.4.2 Sensores para un módulo de monitoreo**

Existe una gran variedad de sensores sofisticados, pero mucho más costosos, todo con el fin de facilitar la medición de parámetros como voltaje, corriente, potencia entre otras para ello se tiene un sensor de corriente SCT-013 y un medidor de voltaje y corriente PZEM-004T.

### a) Sensor no invasivo de corriente SCT-013

Estos sensores cumplen su función como transformadores ya que la corriente que circula por el cable que se va a medir, actúa como un devanado primario e interiormente se tiene un devanado secundario. El número de espiras ya dependerá del modelo que se vaya a usar, esto podemos observar en la Figura 7. (Naylamp, 2021)

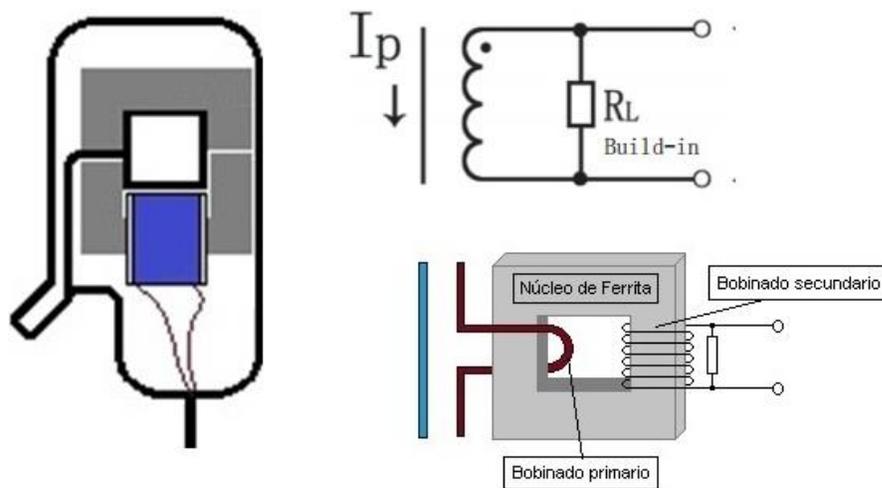


Figura 7. Diagrama sensor de corriente SCT

Fuente. Tomado de (Naylamp, 2021)

Una de las ventajas de este sensor es que no se necesita cortar el cable donde se va a medir la corriente ya tiene una compuerta que abre y cierra para colocar el cable donde se quiere hallar la corriente como se observa en la Figura 8, este sensor soporta medidas de corriente de hasta 100 amperios. (Naylamp, 2021)



Figura 8. Sensor de corriente SCT-013

Fuente. Tomado de (Naylamp, 2021)

## b) Sensor de voltaje y corriente PZEM-016

Este sensor tiene ventajas de medir cuatro distintas variables las que son potencia activa instantánea (W), voltaje (V), amperaje (A) y potencia activa acumulada (Wh). Este dispositivo soporta hasta 100 amperios al igual que tiene un integrado de 8 pines que es una memoria externa EEPROM que almacena los kWh, junto al conector blanco están 2 opto acopladores para la comunicación serial como se observa en la Figura 9. (PDAControl, 2018)



Figura 9. Sensor PZEM-016

Fuente. Tomado de (PDAControl, 2018)

### 1.4.3 Pantalla LCD para la visualización de datos del módulo de monitoreo

Esta pantalla LCD como se muestra en la Figura 10, cuenta con dos filas y a su vez también con 16 columnas el cual permite imprimir mensajes que no sobrepasen dentro del rango admisible caso contrario se puede imprimir resultados o mensajes en recorrido, resultados obtenidos por medio de la programación realizada en Arduino estos resultados pueden ser en tiempo real o pueden ser mensajes que se vayan alternando según el requerimiento del usuario. (Holger, Alonso, & Hernan, 2017)

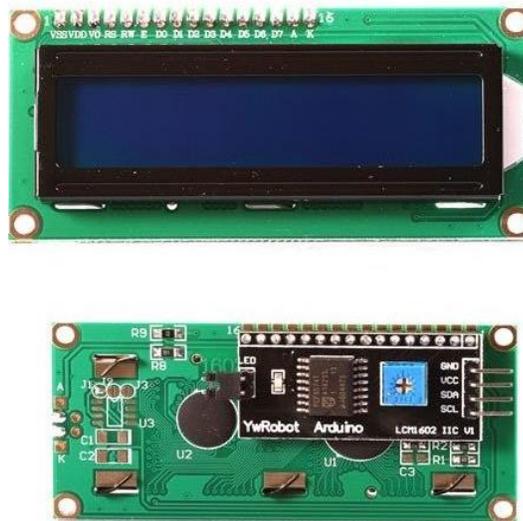


Figura 10. Pantalla LCD 12x6 y módulo I2C

**Fuente:** Tomado de (Garcia, 2013)

El módulo I2C ayuda a la reducción de cables de la LCD ya que se reduce los pines que se conectan con el Arduino que son VCC de 5v, tierra GND y dos pines utilizados para la comunicación SDA y SCL el cual por medio de librerías utilizadas en el Software de desarrollo de Arduino facilita la comunicación entre estos dos elementos (Giraldo, 2019).

### 1.4.4 Entorno de desarrollo Integrado de Arduino

La plataforma de Arduino es un software de uso libre y código abierto basada en hardware y software, por lo que Arduino es un pequeño ordenador que ejecuta una serie de códigos que son introducidos por un usuario desde un ordenador.



```
Arduino 1.8.15
DigitalReadSerial
Archivo Editar Programa Herramientas Ayuda
DigitalReadSerial
/*
  DigitalReadSerial

  Reads a digital input on pin 2, prints the result to the Serial Monitor

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/DigitalReadSerial
*/

// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}
```

Figura 11. Interfaz Arduino IDE

Fuente. (Autor)

Este programa se llama IDE, que significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). El cual debe estar instalado en un computador, su interfaz es muy sencilla de usar y en ella se escribe el programa que se necesita para que Arduino ejecute las acciones solicitadas. Una vez escrito, se carga a través del USB y el Arduino comenzará a trabajar de forma autónoma (León, 2019). En la Figura 11 se puede observar la interfaz de Arduino IDE donde se inserta variables secciones de acción y comentarios para un mayor entendimiento del programa a ejecutar.

## CAPÍTULO II

### Diseño del módulo para el monitoreo de variables de estado de un sistema eléctrico de potencia

A continuación, se presenta los diferentes elementos eléctricos y electrónicos para el monitoreo de un sistema eléctrico de potencia a escala de laboratorio para las cuales se muestra las especificaciones técnicas de todos los materiales a usar con el fin de cumplir con los objetivos propuestos.

#### 2.1 Metodología

La metodología que se utilizó en este trabajo de grado es de tipo bibliográfica, la cual fue fundamental para la recolección de información sobre el monitoreo de los sistemas eléctricos de potencia; además de que permitió adquirir la información necesaria de todo los dispositivos eléctricos y electrónicos que intervienen dentro de este sistema para así llegar a dimensionar los elementos a menor escala del sistema de monitoreo real.

Una vez recolectados los datos que sirvieron como fundamento teórico para esta investigación, se continua con la metodología experimental la cual tiene procesos que permitan desarrollar distintas pruebas para adquirir datos base para la identificación de errores dentro del sistema, el proceso a seguir es el que se muestra en la Figura 12.

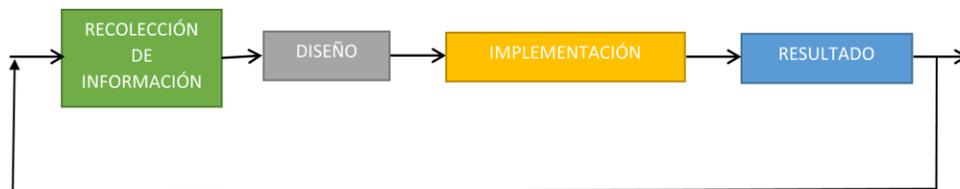


Figura 12. Interfaz Arduino IDE

Fuente. (Autor)

## 2.2 Parámetros para el monitoreo del sistema eléctrico de potencia

Para la implementación del monitoreo del sistema eléctrico de potencia se debe tomar en cuenta los siguientes parámetros:

- a) La selección de elementos con características necesarias para la medición de variables para un módulo a escala.
- b) Adquisición de equipos y materiales necesarios para el desarrollo del proyecto.
- c) Realización de la programación base para cada uno de los dispositivos.

Con estos parámetros se realiza el proyecto tomando como base la adquisición de los equipos y materiales que se usaran para el monitoreo. Posterior a ello se realiza una programación en Arduino los cuales sirven para la adquisición de datos necesarios y el control de los elementos.

## 2.3 Selección de componentes

Para realizar el monitoreo de las variables de estado se consideró componentes de fácil adquisición, bajo costo, y el uso de software libre a la hora de programar, para ello se detalla los componentes a utilizarse como los controladores, sensores, comunicación.

### 2.3.1 Controladores para el monitoreo

A continuación, se detalla las características de los controladores Arduino ya que son dispositivos electrónicos programables para lo cual se tuvo en cuenta aspectos como velocidad de procesamiento, rendimiento y cantidad de entradas analógicas y digitales por lo que se detalla en la Tabla 2 las siguientes características.

Tabla 2 Características de las tecnologías de Arduino

Arduino	Entradas y salidas digitales	Entradas y salidas analógicas	Voltaje (v)	Corriente (mA)	Voltaje de operación (v)	Clock Speed (MHz)
---------	------------------------------	-------------------------------	-------------	----------------	--------------------------	-------------------

<b>Mega</b>	54	12	7 - 12	93	3.3 – 5	16
<b>Uno</b>	14	6	7 - 12	60	3.3 – 5	16
<b>Nano</b>	22	8	7 - 12	19	3.3 – 5	16

**Fuente.** Tomado de (Roda, 2020), (Llamas, 2017), (Meier, 2020)

En la Tabla 2 se observa algunos controladores de Arduino funcionan de 7 a 12 voltios para lo cual se optó por el Arduino Mega el cual es controlador principal y también el Arduino Uno que serán los que funcionen como esclavos ya que cuenta con varias entradas analógicas y digitales.

### 2.3.2 Sensores para el monitoreo

Para la selección del sensor de medición se tiene dos tipos como es el sensor invasivo SCT-013, el sensor PZEM-004T para ello se detalla algunas características para tener en cuenta como se muestra en la Tabla 3.

Tabla 3. Características de los sensores de medición

<b>Sensores</b>	<b>Corriente (A)</b>	<b>Voltaje de entrada (V)</b>	<b>Parámetros de medición</b>
<b>Invasivo SCT-03</b>	30	80 a 660	Corriente
<b>PZEM-016</b>	100	80 a 260	Corriente, voltaje, potencia activa y energía

**Fuente.** Tomado de (Dielect, 2022), (Hernández, 2022)

Como se observa en la Tabla 3 el sensor PZEM-016 cuenta con más parámetros de medición que son de corriente, voltaje, potencia activa y energía al igual que trabaja con una corriente de 100 Amperios por lo que es apto para utilizar en el proyecto.

### 2.3.3 Conversor para el monitoreo

El convertidor RS485 a TTL se trata de un módulo donde se puede realizar la conversión mutua entre la señal TTL o serial y la comunicación RS485 bidireccional, pero también hay el MAX485 que es una comunicación semidúplex, lo que permite realizar una comunicación bidireccional algunas características se muestran en la Tabla 4.

Tabla 4. Características de los conversores

<b>Conversor</b>	<b>Voltaje de operación</b>	<b>Temperatura</b>	<b>Velocidad de transmisión</b>	<b>Tipo de comunicación</b>
<b>RS485</b>	3.3V - 5V	-49°C a + 85°C	35 Mbit/s	Bidireccional
<b>MAX485</b>	3.3V - 5V	-49°C a + 85°C	10 Mbit/s	Bidireccional

Fuente. Tomado de (Naylamp, 2020)

Como se observa en la tabla 4 ambos módulos se diferencian en la velocidad de transmisión por lo que se opta por utilizar el módulo conversor RS485 ya que maneja más velocidad y brinda los requerimientos necesarios para cumplir con el monitoreo de un sistema eléctrico de potencia.

### 2.3.4 Pantalla LCD para el monitoreo

Las pantallas LCD sirve para mostrar datos sin necesidad de estar conectados a un computador por lo que ahorra espacio dependiendo del tamaño algunos tipos se presenta en la Tabla 5.

Tabla 5. Características de los conversores

<b>LCD</b>	<b>Voltajes de operación</b>	<b>Tamaño</b>
<b>16X2</b>	5	2 líneas y 16 caracteres
<b>20X4</b>	5	4 líneas y 20 caracteres

Fuente. Tomado de (Diosdado, 2019)

Cómo se observa en la Tabla 5 se opta por utilizar el LCD de 20x4 ya que tiene mayor tamaño y se programa para observar los parámetros de voltaje, corriente, potencia, factor de potencia entre otros.

### 2.3.5 Componentes seleccionados para el módulo

Una vez seleccionado los componentes para la realización del módulo se incluye los elementos que brindan el voltaje para que operen los demás dispositivos son transformadores, resistencias al igual que otros elementos que aportaran al diseño del módulo por lo que se detalla en Tabla 6 dichos componentes.

Tabla 6. contenido de materiales usados en el monitoreo

Elementos	Nombre	Cantidad	Especificación
	Arduino mega	1	Atmega2560 54 pines digitales, 5v 16 pines analógicos
	Arduino uno	3	AtMega328P 14 pines digitales, 5v 6 pines analógicos
	Medidor Pzem-016	3	Pzem-016 Tensión: 80-260 VAC Corriente: 0 a 100a Frecuencia: 45-65Hz Fp: 0-1
	Conversor Rs485	3	RS485, Voltaje: 5v Corriente de op:120uA – 500uA
	Pantalla LCD 20x4	3	LCD 2004 I2C, 5v Luz de fondo azul

	Resistencia de tiza	6	Resistencia: 4,7 ohm Potencia: 10w
	Barra de cobre	2	
	Transformador reductor	1	Vin: 120v / 220v Vout: 12v / 24V Corriente: 3amp
	Tostadora	1	Voltaje: 120v Potencia: 750w Frecuencia: 60 Hz

Fuente. (Autor)

En la Tabla 6 se muestran los componentes que usaron dentro del monitoreo de un sistema eléctrico de potencia a escala de laboratorio, estos componentes servirán para alimentación de cada uno de los dispositivos como el controlador, sensores, etc.

## 2.4 Flujograma del proceso del módulo

Una vez obtenido todos los componentes que integran el módulo para el monitoreo de las variables de estado de un sistema eléctrico de potencia se realiza el diagrama de flujo como se muestra en la Figura 13 acorde a los requerimientos que se necesita para su correcto funcionamiento.

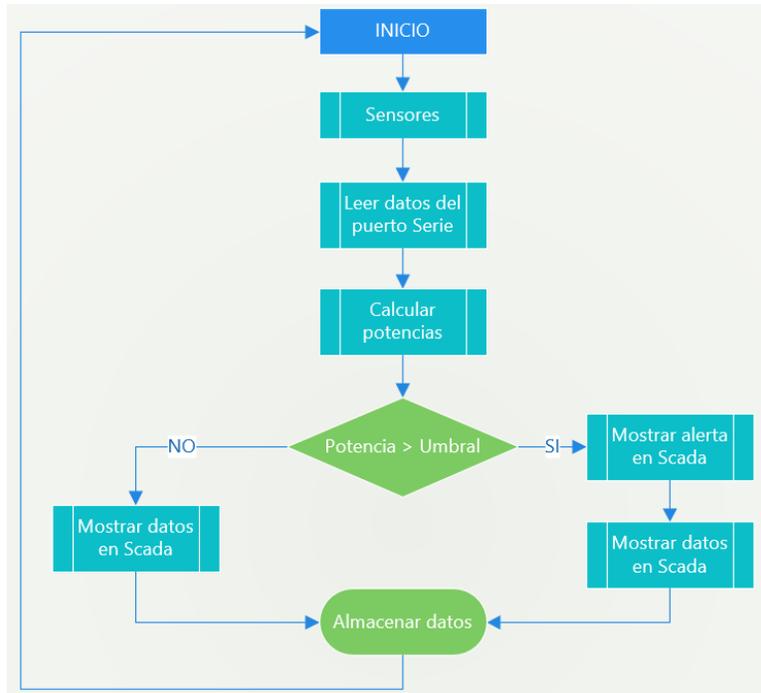


Figura 13. Diagrama de flujo del sistema  
Fuente. (Autor)

En la Figura 13 se muestra el funcionamiento del sistema, desde la lectura de datos hasta llegar a la programación del Arduino. Donde se calcula la potencia para procesarla y transferir los datos en la interfaz del sistema SCADA, en donde se visualiza los valores reales y a su vez el error en las mediciones en caso de existir. Una vez completado todo este proceso el sistema guardara los datos almacenados en un histórico para su posterior uso.

## 2.5 Diagrama esquemático de las conexiones del módulo

En el diagrama mostrado en la Figura 14 muestra el diagrama de bloques del sistema el cual está conformado por los sensores los cuales son los encargados de la adquisición de los datos para posterior ser enviados al Arduino enviados por las entradas analógicas del mismo.

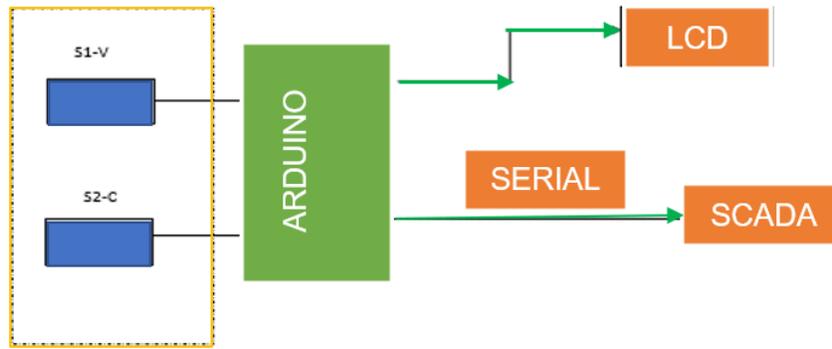


Figura 14. Diagrama de bloques del sistema

Fuente. (Autor)

Una vez enviado los datos de al Arduino los datos se mostrarán en la pantalla LCD y a su vez también serán enviados al SCADA para el monitoreo de todo el sistema.

## 2.6 Diseño electrónico del módulo del sistema eléctrico de potencia

La siguiente Figura 15 muestra el circuito base para comenzar con la simulación del circuito base del sistema.

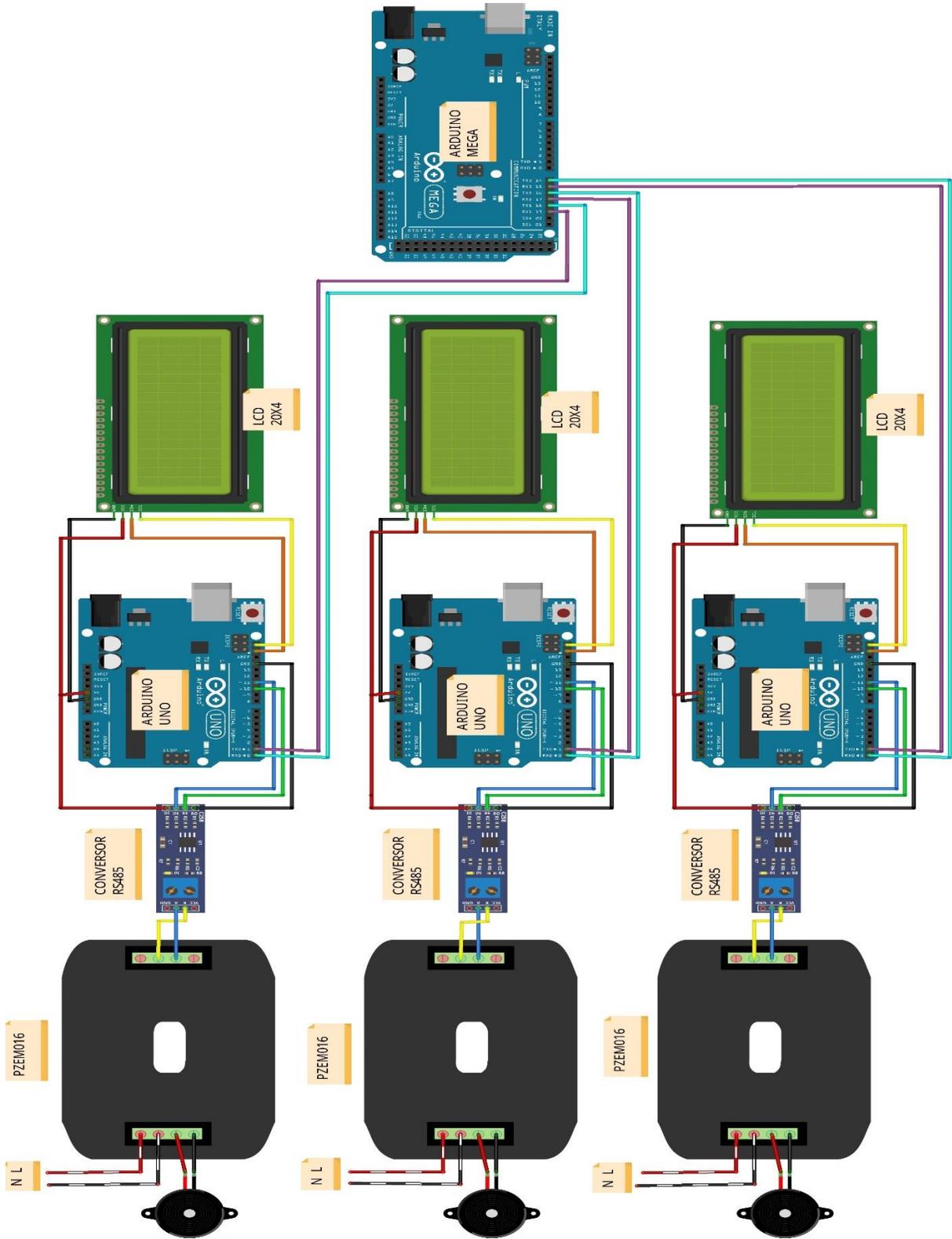


Figura 15. Diseño de conexiones del diagrama electrónico del módulo

Fuente. (Autor)

Para el diseño electrónico se utiliza un Arduino Mega el cual va a ser utilizado como maestro y 3 Arduino Uno que son utilizados como esclavos, también se tiene las pantallas LCD de 20x4 en las cuales se visualiza las mediciones de corriente, voltaje, potencia, factor de potencia, etc. Estos datos se obtienen del módulo PZEM016 ya que a este dispositivo se conecta la carga a medir.

Por medio del conversor RS485 se recibe los datos para procesarlos en cada Arduino Uno para luego enviar toda la información mediante el puerto serie al Arduino Mega, el cual envía toda la información al Matlab para procesar la información en una interfaz, donde se monitorea todas las variables de estado del sistema eléctrico de potencia.

## **2.7 Estimación de estado de un sistema de potencia**

La estimación del sistema de potencia de corriente alterna se realiza a través de la medición de las magnitudes de voltaje y los ángulos de fase en cada una de las barras. No obstante, no es posible medir los ángulos de fase relativos de los voltajes de barra directamente, por lo que se requiere el procesamiento de datos en tiempo real obtenidos del sistema para su cálculo. El proceso de estimación se lleva a cabo mediante un estimador de estado, un programa de computadora que calcula las magnitudes de voltaje y los ángulos de fase relativos de las barras del sistema. A pesar de que el estimador de estado arroja resultados similares a los obtenidos mediante programas convencionales de flujos de potencia, los procedimientos para el suministro de datos y cálculos son considerablemente diferentes (Stevenson & Grainger, 2011).

El estimador de estado se basa en dos tipos de entradas en tiempo real: la información de datos y la información de estado. El estado activo o no activo de los dispositivos de interrupción, como interruptores, cuchillas des conectadoras y derivaciones de transformadores, es lo que determina la configuración de la red, que se modifica cada vez que estos dispositivos se activan o desactivan. Para registrar estos cambios, se utilizan unidades terminales remotas distribuidas por todo el sistema que, además, realizan la verificación de estos.:

- 1) Datos analógicos en la forma de flujos de mega watts y megavars en todas las líneas grandes.
- 2) Cargabilidad de P y Q de generadores y transformadores.

### 3) Magnitudes de voltaje de la mayoría de las barras del sistema.

Las unidades terminales remotas se someten a una verificación periódica que se realiza cada poco segundo, y el conjunto completo de mediciones se recopila a distancia en el centro de control de energía. De esta manera, el estimador de estado se basa en una base de datos más amplia que la utilizada en el flujo de potencia convencional, tal como se describió anteriormente (Stevenson & Grainger, 2011).

En la práctica de la estimación de estado, el número de mediciones reales es significativamente mayor que el número de entradas de datos requeridas para el flujo de potencia de tipo de planeación.

La redundancia en la cantidad de ecuaciones es necesaria, debido a que las mediciones pueden ser incorrectas o no estar disponibles debido a errores en los sistemas de adquisición de datos. El uso directo de las mediciones reales no es aconsejable y se requiere un filtrado de datos para utilizarlos en los cálculos. La estimación de estado realiza esta función de filtrado procesando el conjunto completo de mediciones, de manera que se obtiene un estimado promedio de todas las variables de estado del sistema. Por lo tanto, se logra el mejor ajuste del conjunto completo de datos de entrada en lugar del procesamiento individual de cada medición (Stevenson & Grainger, 2011).

En un sistema de potencia AC, las ecuaciones que describen las mediciones de las variables eléctricas no son lineales y, por lo tanto, no pueden resolverse de manera analítica. Para obtener soluciones numéricas precisas, se utilizan métodos iterativos como el procedimiento de flujos de potencia Newton-Raphson. En este proceso, se parte de un valor inicial para las variables de estado desconocidas y se realizan iteraciones sucesivas hasta que se alcanza una solución que satisface las ecuaciones de las mediciones.

Un ejemplo de las ecuaciones de las mediciones son la Ecuación 1, 2, 3, 4, que son expresiones matemáticas que relacionan las variables eléctricas medidas, como los voltajes y las corrientes, con las variables de estado desconocidas, como los ángulos de fase y las magnitudes de los voltajes en cada nodo de la red eléctrica. El proceso de estimación de estado

utiliza estas ecuaciones y los datos de entrada para calcular los valores de las variables de estado desconocidas, que permiten conocer el estado real del sistema eléctrico en tiempo real

$$e_1 = z_1 - h_1(x_1, x_2) \quad \text{Ecuación 1}$$

$$e_2 = z_2 - h_2(x_1, x_2) \quad \text{Ecuación 2}$$

$$e_3 = z_3 - h_3(x_1, x_2) \quad \text{Ecuación 3}$$

$$e_4 = z_4 - h_4(x_1, x_2) \quad \text{Ecuación 4}$$

Donde, en general,  $h_1, h_2, h_3$  y  $h_4$  son funciones no lineales que expresan las cantidades medidas en términos de las variables de estado, y  $e_1, e_2, e_3$  y  $e_4$  son los términos de ruido de las variables aleatorias gaussianas. Los valores verdaderos de  $x_1$  y  $x_2$  no se conocen y se tienen que estimar de las mediciones de  $z_1, z_2, z_3$  y  $z_4$ .

Se empezará por la formación de la suma ponderada de cuadrados de los errores con ponderaciones  $w_j$ , seleccionados como iguales a los recíprocos de las varianzas de error correspondientes  $\sigma_j^2$ , lo que da como resultado la Ecuación 5.

$$f = \sum_{j=1}^4 \frac{e_j^2}{\sigma_j^2} = \frac{(z_1 - h_1(x_1, x_2))^2}{\sigma_1^2} + \frac{(z_2 - h_2(x_1, x_2))^2}{\sigma_2^2} + \frac{(z_3 - h_3(x_1, x_2))^2}{\sigma_3^2} + \frac{(z_4 - h_4(x_1, x_2))^2}{\sigma_4^2} \quad \text{Ecuación 5}$$

Por medio de esta Ecuación 5 se utilizó para calcular los estimados de estado acorde al diagrama que se muestra en la Figura 16.

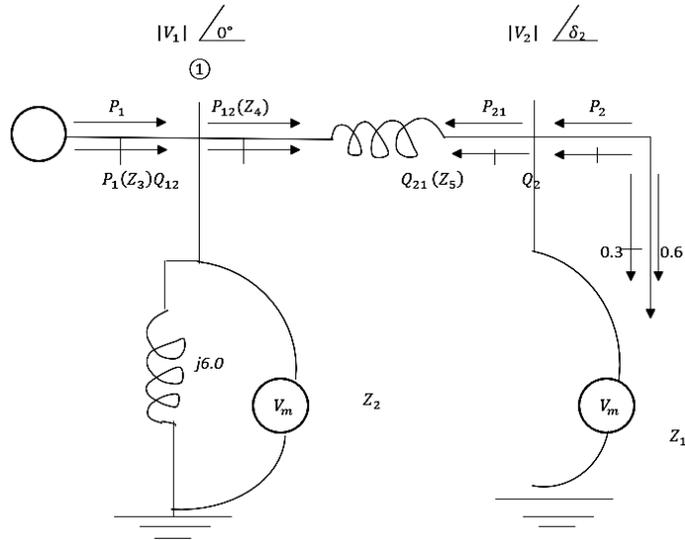


Figura 16. Diagrama unifilar

Fuente. (Autor)

Diagrama unifilar de la Figura 16 muestra las cantidades  $|V_2|$ ,  $|V_1|$ ,  $Q_1$ ,  $P_{12}$  y  $Q_{21}$  que constituye el conjunto de mediciones  $\{Z_1, Z_2, Z_3, Z_4, Z_5\}$  por lo que se llega a la Ecuación 6.

$$\begin{bmatrix} X_1^{(K+1)} \\ X_2^{(K+1)} \\ X_3^{(K+1)} \end{bmatrix} - \begin{bmatrix} X_1^{(K)} \\ X_2^{(K)} \\ X_3^{(K)} \end{bmatrix} = \left( H_x^{(K)T} R^{-1} H_x^{(K)} \right)^{-1} H_x^{(K)T} R^{-1} \begin{bmatrix} e_1^{(K)} \\ e_2^{(K)} \\ e_3^{(K)} \\ e_4^{(K)} \\ e_5^{(K)} \end{bmatrix} \quad \text{Ecuación 6}$$

Con la matriz de ponderaciones de varianzas recíprocas dada por la Ecuación 7.

$$R^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{\sigma_2^2} & \cdot & \cdot \\ \cdot & \cdot & \frac{1}{\sigma_4^2} & \cdot \\ \cdot & \cdot & \cdot & \frac{1}{\sigma_5^2} \end{bmatrix} \quad \text{Ecuación 7}$$

Las ecuaciones desarrolladas en el ejemplo precedente son ilustraciones importantes del procedimiento general y forman la base de los cálculos numéricos para estimar el estado del sistema.

Las mediciones realizadas a distancia en el sistema físico que corresponden a la Figura 16 muestran los valores en por unidad de las cinco cantidades medidas.

$$\begin{aligned} Z_1 = |V_2| &= 0.92 & Z_2 = |V_1| &= 1.02 \\ Z_3 = Q_1 &= 0.605 & Z_4 = P_{12} &= 0.598 \\ Z_5 = Q_{21} &= 0.305 \end{aligned}$$

Las varianzas de los errores de medición se especifican en por unidad que dan como resultado los siguientes datos.

$$\sigma_1^2 = \sigma_2^2 = (0.001)^2 \quad \sigma_3^2 = \sigma_5^2 = (0.002)^2 \quad \sigma_4^2 = (0.015)^2$$

Calculando los estimados de los mínimos cuadrados ponderados de las variables de estado dan como resultado la siguiente Ecuación 8.

$$X_1 = \delta_2, X_2 = |V_2| \text{ y } X_3 = |V_1| \quad \text{Ecuación 8}$$

Sustituyendo los datos  $X_1^{(0)} = 0^\circ$  y  $X_2^{(0)} = X_3^{(0)} = 1.0$  por unidad para las tres variables de estado, da como resultado  $\text{Sen } X_1^{(0)} = 0$ ,  $\text{Cos } X_1^{(0)} = 1$ . Para luego sustituir estos valores numéricos en las Ecuaciones 1, 2, 3, 4, se obtienen los siguientes errores de medición de la primera interacción.

$$e_1^{(0)} = Z_1 - X_2^{(0)} = 0.92 - 1.00 = -0.08$$

$$e_2^{(0)} = Z_2 - X_3^{(0)} = 1.02 - 1.00 = 0.02$$

$$e_3^{(0)} = Z_3 - \left( \frac{25}{6} X_3^{(0)2} - 4X_3^{(0)} - X_2^{(0)} \text{ COS} - X_1^{(0)} \right) = 0.605 - \left( \frac{1}{6} \right) = 0.4383$$

$$e_4^{(0)} = Z_4 - ( - 4X_3^{(0)} X_2^{(0)} \text{Sin } X_1^{(0)} ) = 0.598 - (0.0) = 0.5980$$

$$e_5^{(0)} = Z_5 - ( 4X_2^{(0)2} - 4X_3^{(0)} X_2^{(0)} \text{cos } X_1^{(0)} ) = 0.305 - (0.0) = 0.3050$$

Similarmente, se obtiene la forma numérica de la jacobiana.

$$H_X^{(0)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -4 & \frac{13}{3} \\ -4 & 0 & 0 \\ 0 & 4 & -4 \end{bmatrix}$$

La matriz de ganancias  $G_X^{(0)} = H_X^{(0)\top} R^{-1} H_X^{(0)}$ , se evalúa ahora de la siguiente forma.

$$G_X^{(0)} = \begin{bmatrix} 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -4 & 0 \\ 0 & 1 & \frac{13}{3} & 0 & -4 \end{bmatrix}$$

El método de Newton-Raphson se emplea para estimar el estado en sistemas de potencia de AC. Se empieza con valores iniciales para las variables de estado, que provienen de la solución del flujo de potencia convencional. Estos valores se utilizan para resolver las ecuaciones no lineales de las mediciones en una primera iteración. Los valores obtenidos se utilizan como datos de entrada en la siguiente iteración, y los cálculos continúan hasta que se logra la convergencia. Los valores de las variables de estado en la convergencia se eligen como los estimados del estado del sistema.

$$\hat{X}_1 = \delta_2 = -0.1762 \text{ radianes} = -10.0955^\circ \text{ grados}$$

$$\hat{X}_2 = |V_2| = 0.9578 \text{ por unidad}$$

$$\hat{X}_3 = |V_1| = 0.9843 \text{ por unidad}$$

Los estimados correspondientes de los errores de medición en por unidad son los siguientes.

$$e_1 = Z_1 - \hat{X}_2 = 0.92 - 0.9578 = -0.0378$$

$$e_2 = Z_2 - \hat{X}_3 = 1.02 - 0.9843 = 0.0357$$

$$e_3 = Z_3 - \left(\frac{25}{6} \hat{X}_3^2 - 4\hat{X}_3 \hat{X}_2 \cos \hat{X}_1\right) = 0.605 - 0.3240 = 0.2810$$

$$e_4 = Z_4 - (-4\hat{X}_3 \hat{X}_2 \sin \hat{X}_1) = 0.598 - (0.6610) = -0.0630$$

$$e_5 = Z_5 - (4\hat{X}_2^2 - 4\hat{X}_3 \hat{X}_2 \cos \hat{X}_1) = 0.305 - (-0.0430) = 0.3480$$

Y la suma ponderada de cuadrados de estos errores usando la Ecuación 5 da el siguiente resultado.

$$f = \sum_{j=1}^4 \left(\frac{\hat{e}_j}{\sigma_j}\right)^2 = \frac{(-0.0378)^2}{(0.01)^2} + \frac{(0.0357)^2}{(0.01)^2} + \frac{(0.2810)^2}{(0.02)^2} + \frac{(-0.0630)^2}{(0.015)^2} + \frac{(0.3480)^2}{(0.02)^2} = 545$$

En caso de que la distribución de  $\hat{f}$  siga una función de ji-cuadrado, el número de grados de libertad  $k$  sería 2, debido a que hay tres variables de estado que se deben estimar a partir de cinco mediciones. Para establecer un intervalo de confianza del 99% ( $\alpha = 0.01$ ), es necesario que  $\hat{f}$  sea menor o igual que  $\chi^2(k, \alpha)$ . En este contexto, se ha comprobado que para  $K = 2$  y  $\alpha = 0.01$ ,  $\chi^2(2, 0.01) = 9.21$ . Si bien se ha calculado que  $\hat{f}$  es igual a 545, este valor es muy elevado y sugiere la existencia de datos incorrectos. Por lo tanto, los valores estimados de las variables de estado no pueden considerarse precisos. Además, si hay dos o más errores significativos en el conjunto de mediciones, ya no es posible utilizar este procedimiento debido a que la redundancia se pierde cuando se descartan dos o más mediciones, lo que da como resultado  $N_m - N_s = 5 - 3 = 2$ .

Una vez analizado el ejemplo se relaciona con el módulo, de un sistema eléctrico de potencia que se va a analizar teniendo en cuenta que los valores de resistencias y demás componentes del sistema varían dependiendo la configuración.

## CAPITULO III

### Implementación del módulo para monitoreo de variables de estado de un sistema eléctrico de potencia

Este capítulo se explica la implementación del módulo de monitoreo de variables de estado de un sistema eléctrico de potencia con los componentes seleccionados en el segundo capítulo, también el montaje, comunicación con el sistema SCADA para poder realizar pruebas y ajustes necesarios para observar los resultados que permita llegar al correcto funcionamiento.

#### 3.1 Ensamblaje del módulo

Para el proceso de pruebas y corroboración del funcionamiento del sistema se procedió a armar y probar cada uno de los sensores, como se muestra en la Figura 17, tratando así de tener un orden establecido de cada uno de los componentes.



Figura 17. Pruebas del funcionamiento de los sensores

Fuente. (Autor)

Las pruebas realizadas con los sensores PZEM 0-16 se las realizaron en voltajes que se encuentran en las residencias que son alimentadas con 120V, para así estimar el correcto funcionamiento de estos.

### 3.1.1 Fijación de los componentes eléctricos del sistema

Se llevó a cabo la disposición de los componentes eléctricos del sistema en una base de madera con el propósito de prevenir cualquier tipo de fricción entre ellos y, por lo tanto, evitar posibles errores en las mediciones o lecturas de datos en el sistema. La base de prueba fue diseñada para separar los componentes de corriente de los de medición, a fin de evitar cortocircuitos o daños a cualquiera de los componentes.

#### a) Colocación de barras

Para la colocación de barras se distribuyó acorde al tamaño del tablero utilizado, este tablero tiene medidas de 77x65 cm, para ellos se colocó 4 barras que son la barra infinita, barra 1, barra 2 y barra 3 como se observa en la Figura 18 con una distancia de separación 30 cm entre la barra infinita y la barra 1, también una separación de 10 cm entre la barra 1 y barra 2 y por último 19 cm entre la barra 2 y barra 3.

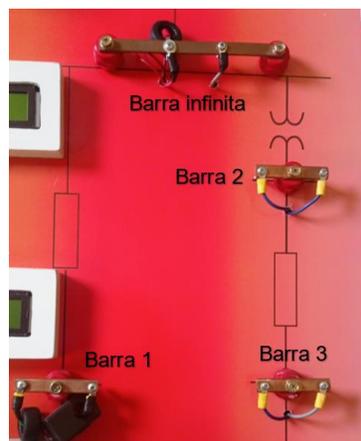


Figura 18. Colocación de barras

Fuente. (Autor)

#### b) Colocación del termomagnético

Con el fin de garantizar la seguridad de las personas y la integridad del sistema eléctrico, es importante contar con la protección contra sobrecargas y cortocircuitos que brindan los termomagnéticos. En este sentido, se ha instalado un termomagnético con una capacidad nominal de 120 voltios y 10 amperios de corriente, para proteger el circuito eléctrico contra una posible falla, en vista que los dispositivos electrónicos usados para la adquisición de datos solo soportan dicho amperaje.

### 3.1.2 Fijación de los componentes electrónicos del sistema

En la parte posterior del tablero se instaló la fijación de los componentes electrónicos del sistema, mientras que la pantalla LCD se colocó en la parte delantera como puede ser vista en la Figura 19.



Figura 19. Fijación de los componentes electrónicos usados para la adquisición de datos

Fuente. (Autor)

### a) Colocación de los controladores Arduino

Los controladores se ubicaron en la parte trasera del tablero con el fin de realizar todas las conexiones y evitar la visibilidad del cableado en la parte delantera, de la misma forma se colocó canaletas como protección del conductor y mejoramiento la parte estética como se observa en la Figura 20. Los controladores son los encargados de la adquisición de todos los datos medidos por los sensores para ser procesados a un sistema SCADA, para ello se tiene un controlador maestro, esclavo 1, esclavo 2, esclavo 3.

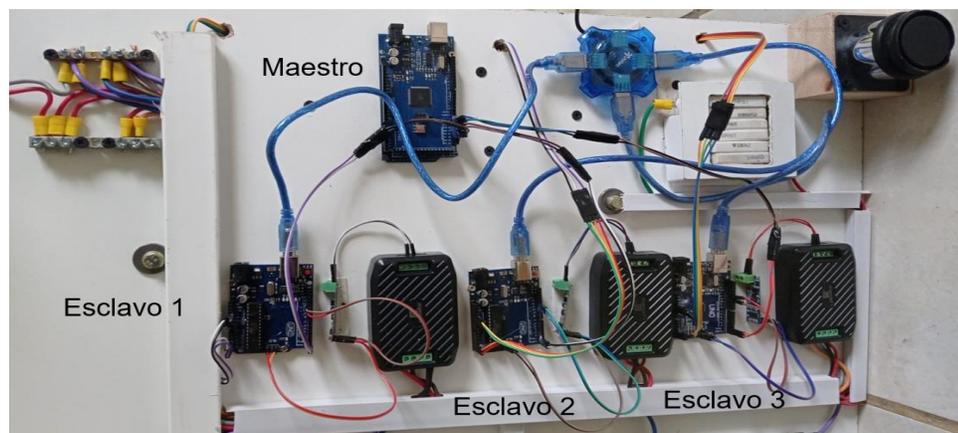


Figura 20. Montaje de controladores Arduino maestro y esclavos.

Fuente. (Autor)

### b) Colocación de los sensores Pzem-016

Los sensores son los dispositivos encargados de realizar las mediciones para cada barra independiente en el circuito. Para ello, se utilizaron 3 Pzem-016, cada uno con su respectivo sensor no invasivo de bobina cerrada y sensor invasivo de bobina abierta. Estos sensores funcionan como un transformador de corriente cerrado para medir la corriente eléctrica que fluye a través del cable conductor en el circuito al que está conectado. La señal de voltaje generada por la bobina cerrada se amplifica y procesa mediante el microcontrolador del medidor de energía para mostrar la información de medición en la pantalla LCD, como se muestra en la Figura 21.

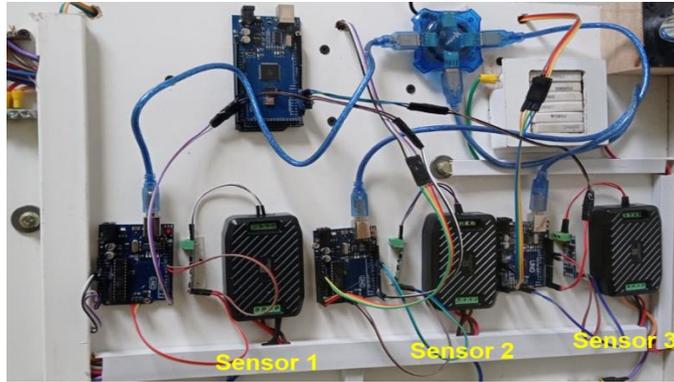


Figura 21. Montaje de sensores Pzem-016

Fuente. (Autor)

### c) Colocación de los conversores

Los conversores se situaron en la parte trasera del tablero como se indica en la Figura 22, con el fin de evitar interferencias en las mediciones ya que estos dispositivos utilizan una comunicación RS485 la que se encarga de transformar el lenguaje de comunicación de los sensores Pzem-016 a un lenguaje de comunicación tipo serie de Arduino.

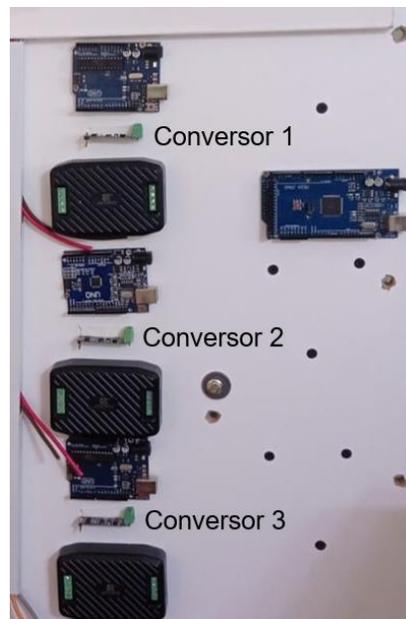


Figura 22. Montaje de conversores

Fuente. (Autor)

#### d) Colocación de las pantallas LCD

Para el monitoreo local se colocó 3 pantallas LDC de 20x4 a una distancia de separación de 11cm entre la pantalla 1 - 2 y 8 cm entre la pantalla 2 – 3 como se identifica en la Figura 23. De esta manera, se puede mostrar fácilmente los datos medidos en tiempo real que llegan desde el sensor.

La pantalla LCD actúa como una interfaz de usuario, y presenta información clara y legible como la corriente, voltaje, potencia y energía consumida en la red eléctrica.



Figura 23. Colocación de pantallas LCD

Fuente. (Autor)

### 3. 2 Creación de la interfaz SCADA

Para la visualización de los datos que se están adquiriendo durante el proceso, se realizó la creación de una interfaz del sistema SCADA en donde se puede visualizar cada uno de los valores de las diferentes variables como se muestra en la Figura 24, además de contar con un sistema de alerta para evidenciar el error en las mediciones.



Figura 24. Interfaz del sistema SCADA

Fuente. (Autor)

En la interfaz se cuenta con apartados para la conexión y desconexión del sistema, los valores mostrados en pantalla son en tiempo real para evidenciar las lecturas de los sensores. El diseño de la interfaz se lo realizo mediante un código en Matlab, donde se delimita la función de cada uno de los botones en la pantalla HMI del sistema.

Todo el sistema se lo desarrollo en la aplicación App Designer varios de los componentes usados son etiquetas, campos de edición numérica, paneles y lámparas. También cuenta con los botones los cuales tiene la función de conectar y desconectar. Esta aplicación de App Designer se usó para monitorear y controlar un sistema de potencia, ya que en la pantalla cuenta con tres paneles titulados “SENSOR1”, “SENSOR2” Y “SENSOR3”. Los paneles mostrados cuentan con campo para mostrar diferentes medidas, como voltaje, corriente, frecuencia y potencia activa y factor de potencia.

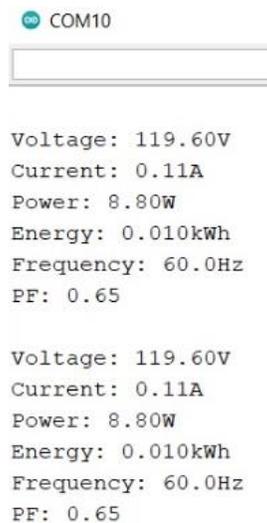
La propiedad “COM” usada en la programación es un objeto serial que se utiliza para lograr establecer la conexión con el puerto serie de la computadora. Esta propiedad generalmente se la usa para poder enviar y a su vez recibir datos a través del puerto serie.

Se cuenta también con otra propiedad muy importante la cual es “estado” es una variable la cual se usa para indicar el estado que se tiene en la conexión con el puerto serie. Es posible que esta variable se pueda usar para determinar si la conexión esta activa inactiva, ambas

propiedades mencionadas anteriormente son privadas lo que significa que solo se pueden ser accedidas dentro de la programación.

### 3. 3 Adquisición de datos de todos los parámetros medidos

Para realizar la adquisición de datos se utilizó las librerías “Wire, PZEM004Tv30 y LiquidCrystal\_I2C”, las cuales facilitaron la comunicación entre los dispositivos electrónicos donde se obtuvo los diferentes parámetros que son voltaje, corriente, potencia, energía, frecuencia y factor de potencia como se muestra en la Figura 25 en donde se muestra los valores reales que se encentra en el sistema.



```
COM10  
  
Voltage: 119.60V  
Current: 0.11A  
Power: 8.80W  
Energy: 0.010kWh  
Frequency: 60.0Hz  
PF: 0.65  
  
Voltage: 119.60V  
Current: 0.11A  
Power: 8.80W  
Energy: 0.010kWh  
Frequency: 60.0Hz  
PF: 0.65
```

Figura 25. Comprobación de medición y comunicación del sensor

Fuente. (Autor)

Teniendo los valores del sensor en tiempo real, dentro de la programación se añade el código para la impresión de datos en la pantalla para poder visualizar todas las variables sin necesidad del monitor serie.

Como resultado se verifica estos datos en la pantalla para monitorear si hay alguna variación en las mediciones durante el tiempo de ejecución del módulo como se muestra en la Figura 26.



Figura 26. Visualización de datos en las pantallas

Fuente. (Autor)

Las mediciones de cada sensor se visualizan en las diferentes pantallas LCD, el sensor se encuentra conectado a distintas barras, que son al final de la barra infinita, al inicio de la barra 1 y al final de la barra 1, esto se puede verificar en la Figura 27.



Figura 27. Conexión para los sensores y demás componentes

Fuente. (Autor)

Otra forma de monitorear las variables medidas es por medio del HMI que es el sistema SCADA que se comunica por medio de un puerto de comunicación serie con el Matlab como se observa en la Figura 28.



Figura 28. Comprobación de valores en HMI

Fuente. (Autor)

### 3.3.1 Cálculo de ángulos de Desfase

Para la adquisición de datos el sensor Pzem-016 envía los valores pico, para lo cual se transforma a valores eficaces, para ello se utiliza el cálculo de los ángulos de desfase, simulaciones, diseños de circuitos, modelo en funcionamiento. Usando la fórmula de los valores eficaces para la potencia se tiene la siguiente Ecuación 9.

$$P = V_{rms} * I_{rms} * \cos \varphi \quad \text{Ecuación 9}$$

Donde

- P: es la potencia activa censada por el sensor PZEM-016.
- $V_{rms}$ : es el voltaje medio censado.
- $I_{rms}$ : es la corriente media censada.
- $\Phi$ : es el ángulo de desfase que existe entre el voltaje y la corriente en un circuito que tiene componentes pasivos como capacitor, inductor, resistencia.

Para despejar  $\varphi$  en la fórmula, primero se debe dividir ambos lados de la ecuación por  $V_{rms} I_{rms}$ . Esto da como resultado la Ecuación 10.

$$\cos \varphi = P/(V_{rms} * I_{rms})$$

Ecuación 10

Luego, para obtener  $\varphi$ , se debe aplicar la función arcocoseno (arcos) a ambos lados de la ecuación y esto da como resultado la Ecuación 11.

$$\varphi = \arcsin (P/(V_{rms} * I_{rms}))$$

Ecuación 11

El ángulo  $\varphi$  representa el ángulo de fase entre la tensión y la corriente en un circuito eléctrico. En un circuito sinusoidal,  $\varphi$  es el ángulo de desfase entre la forma de onda de la tensión y la forma de onda de la corriente. En un circuito eléctrico, el ángulo de fase es importante porque determina la potencia aparente, activa y reactiva en el circuito.

### 3.4 Resultados

Una vez finalizado el módulo se realizó la ejecución del sistema iniciando con la energización, seguido se muestra las mediciones de los valores medidos por los sensores en cada pantalla LCD, luego se establece conexión con el sistema SCADA para monitorear las variables.

A continuación, se genera un histórico de datos que es un archivo de Excel donde se almacena 100 datos de cada variable por los 3 sensores, esto lo realiza cada minuto durante el tiempo que este en ejecución el módulo. Esta parte la ejecuta el código de Matlab como se muestra en la Figura 29.

```

if (count>100)
    folder = "C:\Users\Toshiba\Documents\Tesis datos";
    fecha = convertCharsToStrings(datestr(now,'dd-mm-yyyy_HH_MM_SS'))
    filename = sprintf("datos_%s.xls",fecha)
    filepath = fullfile(folder, filename)

    %tabla = array2table(DATA);
    tabla = array2table(DATA, 'VariableNames', {'Voltaje_1', 'Corriente_1', 'Potencia_1', 'Energia_1', 'Frecuencia_1', '
    writetable(tabla,filepath);

    count=1;
    DATA =[];

```

Figura 29. Código de creación del histórico de datos

Fuente. (Autor)

Donde:

- a) Folder: Es la ruta donde se va a guardar los archivos del histórico de datos.
- b) Fecha: Es la fecha de creación del archivo.
- c) Filename: Es una variable para crear el nombre del archivo.
- d) Filepath: Es una variable la cual abarca al comando fullfile, además sirve para generar un archivo nuevo teniendo en cuenta las variables de folder y filename.
- e) Table: Convierte la matriz data a una tabla y asigna los encabezados a los valores leídos en cada barra.
- f) Writetable: Escribe una tabla con los tributos que se tomaron del comando array2table y se los escribe en un archivo .xls en la dirección de filepath.
- g) La línea count=1 reinicia el contador mientras que Data=[] vacía la matriz de datos.

Una vez obtenido todos los valores del sensor, se procede generar el histórico de datos de todo el sistema en una hoja de Excel, teniendo 100 valores de voltaje, Corriente, energía, potencia, frecuencia, factor de potencia y ángulo de desfase, como se muestra en la Figura 30.

Voltaje_1	Corriente_1	Potencia_1	Energia_1	Frecuencia_1	Fp_1	Angulo_1
119,3	5,18	617,4	4,11	59,9	1	0,05
119,1	5,17	615,5	4,12	59,9	1	0,04
120	5,21	624,8	4,12	59,9	1	0,05
119,8	5,2	622,7	4,12	60	1	0,04
119,3	5,18	617,3	4,12	59,9	1	0,04
119,9	5,2	622,7	4,12	60	1	0,05
120	5,2	623,6	4,12	60	1	0,05
119,9	5,2	622,2	4,12	59,9	1	0,05
119,9	5,19	621,6	4,12	59,9	1	0,06
119,9	5,2	622,4	4,12	60	1	0,04

Figura 30. Valores obtenidos para el histórico de datos

Fuente: (Autor)

De manera estadística se representa los datos obtenidos del sensor 3, cabe recalcar que estos valores se atenúan levemente como se observa en la Figura 31.

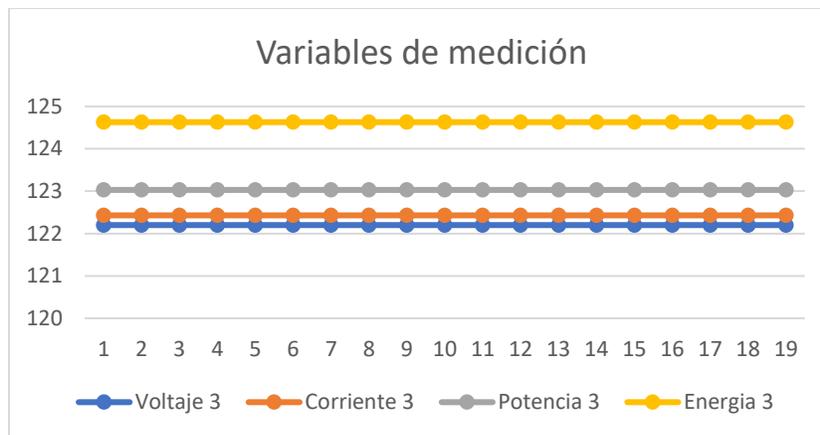


Figura 31. Datos obtenidos de las mediciones

Fuente: (Autor)

Con los datos obtenidos del módulo se ejecuta el código de programación en Matlab, primero colocando los datos censados en las variables M1, M2, M3, M4 como se observa en la Figura 32.

```

1      %VALORES NOMINALES DEL SISTEMA
2      M1=;
3      M2=;
4      M3=;
5      M4=;
6
7      VN=120; %Voltaje base
8      IN=5.10; %Corriente base
9      PN=750; %Potencia base

```

Figura 32. Valores del histórico de datos

Fuente: (Autor)

Donde:

- a) M1= Voltaje 2.
- b) M2= Voltaje 1.
- c) M3= Potencia1.
- d) M4= Potencia 2.
- e) VN= Voltaje base para pasar a por unidad.
- f) IN= Corriente base para pasar a por unidad.
- g) PN= Potencia base para pasar a por unidad.

Se estableció valores iniciales del estado del sistema dados en Por Unidad, teniendo en cuenta que el ángulo en la barra 1 es 0 y el voltaje1 es 1; mientras que el voltaje 2 es igual 1.

Teniendo en cuenta lo anterior se transforma los valores reales a en por unidad para proceder a insertarlos en el vector Z asignando en las posiciones del vector como se muestra en la Figura 33.

```

27 -      V2=M1/VN;           z (1,1)=V2;
28 -      V1=M2/VN;           z (2,1)=V1;
29 -      P1=M3/PN;           z (3,1)=P1;
30 -      P2=M4/PN;           z (4,1)=P2;
31
32 -      Nmediciones=size(z,1);

```

Figura 33. Transformación a por unidad

Fuente: (Autor)

Se establece una condición para que no exista valores negativos y en caso de existir imprimir un mensaje de error. Asimismo, se define un valor de varianza para cada matriz, y luego se coloca una condición para organizar en forma de matriz transformando a su inversa como se observa en la Figura 34.

```

for i = 1:Nmediciones
    if z(i) < 0
        error('Valor negativo encontrado en la medicion %d', i)
    end
end
%LOS VALORES DE LAS VARIANZAS PARA CADA MEDICIÓN N R_inv
VR_S=0.05;

VR(1)=(VR_S^2);
VR(2)=(VR_S^2);
VR(3)=(VR_S^2);
VR(4)=(VR_S^2);

for(i=1:Nmediciones)
    R(i,i)=VR(i);
    Rinv=(inv(R));
end

```

Figura 34. Varianza

Fuente: (Autor)

Se establece las ecuaciones iniciales con las que se realiza el proceso y se calcula el error con respecto a los valores iniciales en por unidad como se muestra en la Figura 35.

```

%LAS MEDICIONES CALCULADAS INICIALES

h(1,1)=X(2,1);
h(2,1)=X(3,1);
h(3,1)=((6.41*(X(3,1))*(X(2,1))))*sin(-X(1,1));
h(4,1)=((6.41*(X(3,1))*(X(2,1))))*sin(X(1,1));

%LOS ERRORES INICIALES MEDIDOS
err=z-h;

```

Figura 35. Ecuaciones del sistema

Fuente: (Autor)

La matriz H se forma de un conjunto de ecuaciones no lineales basadas en las mediciones y variables de estado del sistema, la función principal de la matriz es minimizar el error de las

mediciones del sistema basándose en el método de Newton Raphson como se muestra en la Figura 36.

$$Hx = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6.41 \cdot X(3,1) \cdot X(2,1) \cdot \cos(X(1,1)) & -6.41 \cdot X(3,1) \cdot \sin(X(1,1)) & -6.41 \cdot X(2,1) \cdot \sin(X(1,1)) \\ 6.41 \cdot X(3,1) \cdot X(2,1) \cdot \cos(X(1,1)) & 6.41 \cdot X(3,1) \cdot \sin(X(1,1)) & 6.41 \cdot X(2,1) \cdot \sin(X(1,1)) \end{bmatrix};$$

Figura 36. Matriz jacobiana

Fuente: (Autor)

Seguido se realiza la matriz de ganancia para introducir los valores de varianza dentro del sistema.

Se compara valores para verificar si el proceso sigue iterando o se ya lleo a su resultado final y se establece algunas condiciones que debe cumplirse teniendo en cuenta k, alfa y la suma ponderada de los errores, tiendo que k es el número de mediciones menos el número de estimados siendo igual a 1, por lo que le corresponde un alfa de 0.01 y un X=6.64, para verificar que la suma ponderada de los errores sea menor a X.

Si la suma ponderada de los errores es mayor a 6.64 sigue existiendo iteraciones hasta llegar a la convergencia, pero si la suma de los errores es menos a 6.64 significa que ya no existen mediciones erróneas dentro del sistema como se muestra en la Figura 37, este límite se lo delimita al momento de encontrar alfa y X.

```

%SE CALCULAN LOS NUEVOS VALORES DE X HASTA LA CONVERGENCIA
X=X+(inv(Gx)*Hx'*Rinv*err);

if(abs(X-Xi)<ERRMAX)
    fprintf('SE REALIZARON %d ITERACIONES PARA LA CONVERGENCIA\n',j);
    %SE CALCULA LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES
    for i=1:Nmediciones
        f=f+((err(i)^2)*(Rinv(i,i)));
    end
    fprintf('LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES ES: f=%d\n',f);
    fprintf('LA DISTRIBUCIÓN CON:\n K=%d Y ALFA=%d PERMITE UN f<=X^2(K,ALFA)<=%d\n',k,alfa,X_K_ALFA);
    if(f>X_K_ALFA)
        fprintf('POR LO QUE EXISTEN MEDICIONES ERRONEAS A SER IDENTIFICADAS\n',f);
        break
    else
        fprintf('POR LO QUE NO EXISTEN MEDICIONES ERRONEAS A SER IDENTIFICADAS\n',f);
        break
    end
end
end
Xi=X;
end

```

Figura 37. Comparación de error

Fuente: (Autor)

Se calculo la matriz de covarianza para luego generar una matriz quitando el error generado en las primeras iteraciones.

Una vez obtenido los resultados se obtiene el error máximo que existe realizando una comparación como se observa en la Figura 38, para luego recalculer las iteraciones, pero eliminando el error.

```

if min(ej) >= 0.00001 && min(ej)< 0.09
    fprintf('LUEGO DE CALCULAR LOS ERRORES ESTANDARIZADOS:\n EL ERROR MAS GRANDE ESTA EN:ej(%d)=%d\n',find(ej==abs(min(ej))),abs(min(ej)));
elseif max(ej)>0.1
    fprintf('LUEGO DE CALCULAR LOS ERRORES ESTANDARIZADOS:\n EL ERROR MAS GRANDE ESTA EN:ej(%d)=%d\n',find(ej==abs(max(ej))),abs(max(ej)));
end

```

Figura 38. Sentencia comparativa

Fuente: (Autor)

Se vuelven a escribir las ecuaciones para volver a recalculer los valores exceptuando la mediación errónea como se muestra en la en la Figura 39.

```

h(1,1)=X(2,1);
h(2,1)=X(3,1);
h(3,1)=((6.41*(X(3,1))*(X(2,1))))*sin(-X(1,1));
h(4,1)=((6.41*(X(3,1))*(X(2,1))))*sin(X(1,1));
%h(5,1)=(4*(X(2,1)^2)-(4*X(3,1)*X(2,1)*cos(X(1,1))));

%SE VUELVE h CALCULADO = z MEDIDO,, ASÍ? EL ERROR ES CERO PARA ELIMINAR EL DATO EN LOS CÁLCULOS
if min(ej) <= 0.1
    h(find(ej==abs(min(ej))),1)=z(find(ej==abs(min(ej))),1);;
elseif max(ej)>0.1
    h(find(ej==abs(max(ej))),1)=z(find(ej==abs(max(ej))),1);;
end
% h(find(ej==abs(max(ej))),1)=z(find(ej==abs(max(ej))),1);

```

Figura 39. Nuevas ecuaciones

Fuente: (Autor)

Se vuelve a escribir la matriz H con las mismas ecuaciones presentadas en la Figura 39, para calcular el error quitando la medición errónea encontrada anteriormente como se muestra en la Figura 40.

```

Hx2=[0                1                0
      0                0                1
      -6.41*X(3,1)*X(2,1)*cos(X(1,1))  -6.41*X(3,1)*sin(X(1,1))  -6.41*X(2,1)*sin(X(1,1))
      6.41*X(3,1)*X(2,1)*cos(X(1,1))    6.41*X(3,1)*sin(X(1,1))    6.41*X(2,1)*sin(X(1,1))    ];

%LA MATRIZ DEGRADACIONES CALCULADA
Gx=Hx'*Rinv*Hx;

%SE CALCULAN LOS NUEVOS VALORES DE X HASTA LA CONVERGENCIA
X=X+(inv(Gx)*Hx'*Rinv*err);
%Condicionales para quitar la medicion donde existe el mayor error
if(abs(X-Xi)<ERRMAX)
    if min(ej) >= 0.00001 && min(ej)< 0.09
        fprintf('\n \n AL RECALCULAR SIN LA MEDICIÓN Z%d:\n ',find(ej==abs(min(ej)))));
        fprintf('SE REALIZARON %d ITERACIONES PARA LA CONVERGENCIA\n',j);
    elseif max(ej)>0.1
        fprintf('\n \n AL RECALCULAR SIN LA MEDICIÓN Z%d:\n ',find(ej==abs(max(ej)))));
        fprintf('SE REALIZARON %d ITERACIONES PARA LA CONVERGENCIA\n',j);
    end

%SE CALCULA LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES
f=0;

```

Figura 40. Nueva matriz jacobiana

Fuente: (Autor)

Se vuelve a comparar los errores de la suma ponderada con X para determinar si no existen errores dentro del sistema.

Como resultado final que se obtiene todo el proceso realizado donde se detalla cada resultado obtenido como se muestra en la Figura 41.

```

SE REALIZARON 2 ITERACIONES PARA LA CONVERGENCIA
LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES ES: f=5.102948e+02
LA DISTRIBUCIÓN CON:
K=1 Y ALFA=1.000000e-02 PERMITE UN  $f \leq X^2(K, ALFA) \leq 6.640000e+00$ 
POR LO QUE EXISTEN MEDICIONES ERRONEAS A SER IDENTIFICADAS
LUEGO DE CALCULAR LOS ERRORES ESTANDARIZADOS:
EL ERROR MAS GRANDE ESTA EN:  $e_j(1)=1.006309e+00$ 

AL RECALCULAR SIN LA MEDICIÓN Z1:
SE REALIZARON 1 ITERACIONES PARA LA CONVERGENCIA
LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES ES: f=2.551474e-02
LA DISTRIBUCIÓN CON:
K=1 Y ALFA=1.000000e-02 PERMITE UN  $f \leq X^2(K, ALFA) \leq 6.640000e+00$ 
POR LO QUE NO EXISTEN MEDICIONES ERRONEAS A SER IDENTIFICADAS

```

Figura 41. Resultado obtenido mediante la simulación en Matlab

Fuente: (Autor)

Como se observa en la Figura 41 para realizar estos cálculos se requiere el modelo iterativo basado en Newton Raphson desacoplado, teniendo en cuenta variables de estado y mediciones del sistema, para encontrar el error y minimizarlo.

Estos cálculos se realizan tendiendo la sentencia “for” para recalcular en base a las iteraciones máximas establecidas, esto con el fin de obtener el valor máximo de error en alguna medición global del sistema y minimizarlo.

## CONCLUSIONES

- a) Se determino que las principales variables que se va a analizar en la monitorización en tiempo real de un sistema eléctrico de potencia las cuales son el voltaje, potencia, frecuencia, energía, factor de potencia y ángulos desfase, siendo estas las más factibles para el análisis en cada una de las barras que conforman un el sistema.
- b) Para el monitoreo de las barras del sistema eléctrico de potencia se hace uso de las variables de estado como son voltaje, corriente y potencia, que intervienen en el módulo de laboratorio, siendo estas las que tienen una variación dentro del sistema, en cada punto de medición.
- c) Usando programaciones en Arduino para la adquisición de datos de las diferentes variables, las cuales se complementan con el software Matlab se llega a determinar las diferentes variaciones y errores calculados en las mediciones en cada barra, teniendo como referencia un valor nominal.

## RECOMENDACIONES

- a) Para estudios futuros se sugiere integrar los módulos de generación y transmisión eléctrica de laboratorio para así obtener un monitoreo general de un sistema eléctrico de potencia completo.
- b) El monitoreo del sistema eléctrico de potencia, este sujeto a variaciones de las diferentes variables por lo que se recomienda el rediseño de la parte estructural del sistema dependiendo de las barras que se vayan a acoplar y variables que se requiera la medición.
- c) Para realizar el acoplamiento del módulo a un nuevo sistema, se debe realizar el modelado y análisis matemático de todas las cargas, variables y barras que se vayan a usar dentro del monitoreo para así poder evitar que existan fallas al momento de calcular el error en las mediciones del sistema.
- d) Uno de los errores que se evidenció fue al momento de correr el programa App designer, se generó un error en el puerto de comunicación, debido a que se encuentra en uso por el monitor serie, para corregir el problema se debe dar clic en el botón de desconector del App designer seguido de ello correr nuevamente el programa.

## REFERENCIAS

- Amaguaña, L. (2021). Estudio para el mejoramiento del factor de potencia en los puntos de interconexión de emelnorte con el sistema nacional interconectado. 76.
- ANSI/NEMA MG 1. (2016). *Motors and Generators Supplements*.
- Arduino. (2022). *Arduino*. Recuperado el 10 de Abril de 2022, de <https://arduinodehtics.weebly.com/tipos-de-arduino.html>
- Arias Calle, F. S., & Bravo López, A. S. (2016). Estudio para la modernización de la Central Hidroeléctrica Saucay. *Tesis previa a la otención del Título de Ingeniero Eléctrico*. Universidad de Cuenca, Cuenca.
- Boldea, I. G. (2016). *Synchronous Generator*. New York: Taylor & Francis Group.
- Carakenio73. (25 de marzo de 2019). *dademuchconnection*. Obtenido de <https://dademuch.com/2019/03/25/definicion-de-sistema-electrico-de-potencia/>
- Castillo, O., Katerin, P., Sánchez, L., & Dominic, A. (2019). Sistema "FlextHand" para mejorar la motricidad de las manos en pacientes con accidente cerebrovascular del Hospital Belén, Trujillo. 193.
- Castro, R. J. (21 de mayo de 2107). *Sector electricidad*. Obtenido de <https://www.sectorelectricidad.com/17778/una-introduccion-a-la-estimacion-de-estados-en-sistemas-de-potencia/>
- Chapman, S. J. (2012). *Maquinas Eléctricas* (Quinta ed.). Mexico: The McGraw-Hill.
- Chiguano, B. .: (2018). Estimación de los Parámetros Eléctricos de un Generador. *Revista Técnica Energía*, 43.
- Chimarro Alomoto, A. P. (2019). Trabajo de titulación previo a la otención del título de ingeniera eléctrica. *Interfaz gráfica para el monitoreo de la operación en estado estable de las unidades de generación eléctrica de la central coca codo sinclair*. Escuela Politécnica Nacional, Quito.
- CONELEC. (2013). PPlan Maestro de Electrificación 2013-2022. En CONELEC, *Perspectiva y expansión del sistema eléctrico Ecuatoriano* (pág. 55). Ecuador.
- Costa, M., Martinez, A., Alves, M., & Araujo, D. (s.f.). *treetech*.

- CT, P.-0. E. (2023). *Mikroelec.com*. Obtenido de <http://www.mikroelec.com/product/1668/pzem-016-energy-meter-ac-100a-voltage-current-energy-hz-power-factor-rs485-modbus-module-with-ct>
- Cummins. (2007). *slideplayer.es*. Obtenido de Sistemas de generadores de excitación: <https://slideplayer.es/slide/17103/>
- Danilo, S. (2019). Metodología para la evaluación y mejora de la estabilidad de voltaje de un sistema eléctrico de potencia. 101.
- Dielect. (2022). Recuperado el 19 de Octubre de 2022, de <https://ssdielect.com/magnitudes-electricas-1/168-md-pzem-004t.html>
- Diosdado, R. (5 de Mayo de 2019). *Maker Zona*. Obtenido de <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/tipos-de-lcd-para-arduino>
- Duncan, G., & Mulukutla, S. (2012). *Sistemas de Potencia Análisis y Diseño*.
- Duncan, O., Jaramillo, W., & Gomez, A. (2017). Flujo de potencia óptimo para redes radiales y enmalladas empleando programación semidefinida. 15.
- Fernández Diego, I., & Robles Díaz, A. R. (2012). El Generador Síncrono. *Centrales de Generación Eléctrica*. Universidad de Cantabria, Cantabria.
- Gallego Rendón, R. A., Escobar Zuluaga, A. H., & Granda Echeverri, M. (2016). Flujo de carga en sistemas de transmisión Modelamiento y análisis. *Coleccion Textos Académicos Facultad de Tecnología*. Universidad Tecnológica de Pereira, Pereira.
- García, A. (08 de febrero de 2013). *Panamahitek*. Obtenido de <http://panamahitek.com/uso-de-pantalla-lcd-con-arduino/>
- geekfactory. (29 de mayo de 2017). *geekfactory*. Obtenido de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/lcd-16x2-por-i2c-con-arduino/>
- Geoff Klempner, I. K. (2018). *The Handbook of Large Turbo-Generator Operation and Maintenance*. United States of America.: IEEE.

- Gerrero Hernández, A., Araque Gallardo, J., & Gallo Nieves, M. (2016). Implementación de módulos didácticos para sistemas electrónicos de potencia. *Revista Educación en Ingeniería*, 9-13.
- Giraldo, S. A. (04 de Septiembre de 2019). *Control Automatico Education* . Obtenido de <https://controlautomaticoeducacion.com/microcontroladores-pic/lcd-i2c-pic/>
- Gomez, A., Conejo, A., & Canizares, C. (2018). *Electric Energy Systems*.
- Hernández, L. (2022). *Programafacil*. Recuperado el 19 de Octubre de 2022, de <https://programarfacil.com/blog/arduino-blog/sct-013-consumo-electrico-arduino/>
- Holger, M., Alonso, P., & Hernan, R. (2017). Monitoreo del Consumo de Energía Eléctrica Doméstica con Arduino. 5.
- IEEE 115. (2009). *IEEE Guide for Test Procedures for Synchronous Machines*. New York: IEEE.
- IEEE 43. (2013). *IEEE Recommended Practice for Testing Insulation Resistance of Electric Machinery*. Nueva York: IEEE.
- Inmaculada Fernández, D., & Robles Díaz, A. R. (2017). *opencourseware*. Obtenido de opencourseware: <https://ocw.unican.es/pluginfile.php/1160/course/section/1407/bloque-energia-II.pdf>
- Izquierdo Franco, J. (2002). Estudio de flujos de potencia y análisis de fallas en sistemas eléctricos de distribución radial. *Estudio de flujos de potencia y análisis de fallas en sistemas eléctricos de distribución radial*. Universidad Autónoma de Nuevo León, San Nicolás de los Garza, Mexico.
- Jorge, V., Lina, M., Flavio, F., Julio, D., Faustino, C., & Jose, G. (2017). Entorno de aprendizaje para la enseñanza de programación en Arduino mediado por una mano robótica didáctica. *Espacios*, 23.
- Landy, W. (2015). En U. P. Salesiana, *Desarrollo de un modelo de sistema de control y monitoreo de parámetros operacionales de un transformador de potencia, para el diagnóstico y detección temprana de fallas* (pág. 173). Cuenca.

- Landy, W. (2015). Desarrollo de un modelo de sistema de control y monitoreo de parámetros operacionales de un transformador de potencia, para el diagnóstico y detección temprana de fallas. En U. P. Saieciana. Ibarra.
- León, M. M.-F.-H. (2019). *Impresión 3D con materiales elásticos*.
- Leroy Somer, Nidec. (2018). *Wind Turbine Generators*.
- Llamas, L. (16 de octubre de 2016). *Ingeniería, Informática y diseño*. Obtenido de <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>
- Llamas, L. (17 de Noviembre de 2017). *Ingeniería, informática y diseño*. Recuperado el 18 de Mayo de 2020, de <https://www.luisllamas.es/modelos-de-raspberry-pi/>
- López-Rey García-Rojas, Á. (2016). Influencia del marco regulatorio en la implantación de la generación distribuida. *Programa de Doctorado en Tecnologías Industriales*. Universidad Nacional de Educación a Distancia , España.
- Meier, L. (3 de Marzo de 2020). *Pixhawk*. Recuperado el 18 de Mayo de 2020, de <https://pixhawk.org/products/>
- Mera, A., & Ruano, D. (2019). *Implementación de un robot de navegación autónoma terrestre para evasión de obstáculos usando el dispositivo PIXHAWK*. Tesis, Universidad de las Américas, Quito. Recuperado el 1 de Diciembre de 2019
- Naylamp. (2020). *Mechatronics*. Recuperado el 21 de Enero de 2023, de [https://naylampmechatronics.com/blog/37\\_comunicacion-rs485-con-arduino.html](https://naylampmechatronics.com/blog/37_comunicacion-rs485-con-arduino.html)
- Naylamp. (2021). *Mechatronics*. Recuperado el 19 de Septiembre de 2022, de [https://naylampmechatronics.com/blog/51\\_tutorial-sensor-de-corriente-ac-no-invasivo-sct-013.html](https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-sct-013.html)
- Ninazunta Anaguano, J. A. (2019). Estimación de estado dinámico de un generador sincrónico aplicando la técnica del filtro Kalman extendido. *Proyecto previo a la obtención del título de ingeniero en electrónica y control*. Escuela Politécnica Nacional, Quito.
- Oleas, A. (2019). Control de voltaje en sistemas eléctricos de potencia usando transformadores con taps. 35.

- Orjuela Tirano, L. F., & Cortés Bolívar, J. D. (2020). Estimación de los parámetros del generador síncrono del módulo TNA a partir de oscilogramas eléctricos. *Trabajo de grado para optar por el título profesional de Ingeniero Electricista*. Universidad La Salle, Bogotá.
- Ortiz Boada, A. F., & Viteri Navarrete, F. A. (2014). Estudio, diseño e implementación de un nuevo sistema de protección eléctrica para el motor síncrono de 3Mw-4.14Kv del molino de bolas de crudo de la fabrica de cemento Holcim Ecuador- Planta Guayaquil. *Tesis previa a la obtencion del título de Ingeniero Electrico*. Universidad Politecnica Salesiana, Guayaquil.
- Patiño, F., & Limas, G. (2018). *Metodologías para el análisis de estabilidad de Tensión en estado estacionario*. Tesis, Universidad Tecnológica de pereira. Recuperado el 12 de Agosto de 2021, de <https://core.ac.uk/download/pdf/71396041.pdf>
- PDAControl. (12 de Julio de 2018). *Control*. Recuperado el 20 de Septiembre de 2022, de <http://pdacontroles.com/medidor-pzem-004t-con-arduino-esp32-esp8266-python-raspberry-pi/>
- Pelaesz, E., & Jimenez, P. (2018). Diseño de un Sistema de Medición y Monitoreo del Consumo de Energía por Circuitos en el Hogar, Mediante Tecnología de Comunicación por Línea de Potencia. 92.
- Pilatásig, J. (2020). Resiliencia de Sistemas Eléctricos de Potencia mediante la Conmutación de Líneas de Transmisión. *Revista Id tecnologico*, 11.
- Postes, J., & Rendon, A. (2019). Análisis de flujo de corriente continua aplicado a sistemas eléctricos de potencia. 101.
- Ramírez Medina, H. J., & Sánchez Barroso, R. G. (2012). Diseño y Contrucción de un tablero de tranferencia y sincronización de generdpres de emergencia con PLC y pantalla táctil. *Tesis de grado previa a la obtención del título de Ingeniero en Mantenimiento*. Escuela Sueprior Plitécnica de Chimborazo, Riobamba.
- REGLAMENTO DE REGIMEN ACADEMICO CONSEJO. (2017, 22 de marzo). *Resolución del Consejo de Educación Superior 51*. Ecuador: Registro Oficial Edición Especial. Obtenido de <https://www.ces.gob.ec/lotaip/Anexos%20Generales/a3/Anexo-lit-a3-R-R%C3%A9g-Acad%C3%A9mico.pdf>

- Roda, G. (18 de Mayo de 2020). *Arduino, Genuino, Raspberry Pi. Noticias y proyectos*. Recuperado el 19 de Mayo de 2020, de <https://descubrearduino.com/comparamos-varias-de-las-arduino-mas-utilizadas/>
- Rodríguez Chicaiza, D. A. (2008). Graficación en tiempo real de curvas de capacidad de generadores sincrónicos en sistemas de potencia. *Proyecto previo a la obtención del título de ingeniero Eléctrico*. Escuela Politécnica Nacional, Quito.
- Rodríguez Pozuela, M. A. (2015). Máquinas Síncrona. *Departamento de Ingeniería Eléctrica y Energética*. Universidad de Cantabria, Cantabria.
- Rodríguez Pozueta, M. (2012). Transformadores. *Departamento de Ingeniería Eléctrica y Energética*. Universidad de Cantabria, Cantabria, España.
- Rodríguez Pozueta, M. A. (2014). Máquina síncrona funcionamiento en red aislada en potencia infinita y en paralelo con otras. Motor Síncrono. *Departamento de Ingeniería Eléctrica y Energética*. Universidad de Cantabria, Cantabria.
- Roman, L. (10 de Febrero de 2021). *Educación 3.0*. Obtenido de <https://www.educaciontrespuntocero.com/recursos/arduino-primaria/>
- Romero Guavita, J., & Núñez Eguis, M. (2017). Tecnología en Electricidad. *Generador Síncrono*. Universidad Distrital Francisco José de Cladas, Caldas, Colombia.
- Sierra, D. (2017). Propuesta de nuevas prácticas de la materia microcontroladores utilizando laplaca arduino mega 2560 como complemento y mejora del curso actual. 196.
- Stevenson, W., & Grainger, J. (2011). *Análisis de sistemas de potencia*. Mexico, Mexico. Recuperado el 10 de Febrero de 2023, de <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiRvaGnkJb9AhUzjLAFHd5WDw0QFnoECBAQAQ&url=https%3A%2F%2Fcedras.facet.unt.edu.ar%2Fsep%2Fwp-content%2Fuploads%2Fsites%2F20%2F2020%2F03%2FAn%25C3%25A1lisis-de-Sistemas>
- Tapia, E. (2015). Monitoreo en tiempo real de oscilaciones de potencia en el corredor oriental molino – totoras - santa rosa - pomasqui utilizando mediciones sincrofásicas de las subestaciones pomasqui y molino a nivel de 230 kv. 156.

- Tebiania, H. (2014). Dynamic State Estimation in Power Systems. *Master of engineering Faculty of Engineering and Applied Science*. Memorial University, Newfoundland.
- Tenelema, E. (2019). *Simulación de flujos de potencia en sistemas eléctricos de potencia usando métodos completos, desacoplados y leanizados*. Universidad politécnica salesiana sede quito, Quito. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/17625/1/UPS%20-%20ST004217.pdf>
- Tiberio, V., Jaime, A., Luis, C., & Victor, C. (2017). Análisis comparativo del estudio de flujos de potencia en series de tiempo por el cambio de paso en el muestreo de perfiles de carga y generación de sistemas fotovoltaicos interconectados. *Revista de Aplicación Científica y Técnica*, 19.
- Tonato Muñoz, J. P. (2020). Manual operación y mátenimiento eléctrico del generador síncrono de una unidad de generación del la Central Hidroeléctrica Paute Sopladora. *Proyecto técnico con enfoque general*. Universidad poitécnica Saleciana sede Cuenca, Cuenca, Ecuador.
- Torres, H. (7 de Diciembre de 2017). *Hetpro*. Recuperado el 28 de Diciembre de 2019, de <https://hetpro-store.com/TUTORIALES/arduino-vs-raspberry-pi/>
- Universidad José Matias Delgado. (2015). *webquery.ujmd.edu.sv*. Obtenido de <https://webquery.ujmd.edu.sv/siab/bvirtual/Fulltext/ADLD0000526/Capitulo%202.pdf>
- Universidad Técnica del Norte. (2019, 11 de enero). *Resolución Nro. 001-073-CEAACES-2013-13*. Slider legislacion UTN. Obtenido de <http://www.utn.edu.ec/legislacion/?cat=2>
- Valenzuela Santillán, A. (2013). Sincronoscopio virtual para un sistema mulltimáquina del laboratotio de sistemas electros de potencia. *Proyecto previo a la obtencion del título de ingeniero eléctrico*. Escuela Politécnica Nacional, Quito, Ecuador.
- Ventura, V. (15 de Feberero de 2015). *Polaridad*. Obtenido de <https://polaridad.es/conexion-de-una-tarjeta-sd-por-spi-a-arduino/>
- Viscaino, R. (2015). Desarrollo de un modelo de sistema de control y monitoreo de parámetros operacionales de un transformador de potencia, para el diagnostico y detección temprana de fallas. *Universidad Politecnica Salesiana* , 7-12.
- Walter Brokering, R. P. (2008). *Los Sistemas Electricos de Potencia*. Mexico: Prentice Hall.

Watt, M. (20 de Abril de 2020). *vrelectrificacione*. Obtenido de <https://vrelectrificaciones.com/estructura-general-de-un-sistema-electrico-de-potencia/>

## ANEXOS

### Anexo A: Programación de Pzem-016 para la adquisición de datos

```
//En esta seccion se llama a las librerias del sensor PZEM
#include <PZEM004Tv30.h> //Incluir la libreria de PZEM014
#include <Arduino.h>
#include <String.h>

PZEM004Tv30 pzem(11, 12); //Software Serial pin 11 (RX) & 12 (TX)

#include <Wire.h> //Incluimos la libreria de WIRE que permite la comunicaci3n con dispositivos
de comunicaci3n I2C

#include <LiquidCrystal_I2C.h> //Incluimos la libreria LiquidCrystal para usar la pantalla LCD

LiquidCrystal_I2C lcd(0x27, 20, 4); // seteamos la direcci3n de la LCD a 0x27 para 20
caracteres y 4 lineas de pantalla

String valores = "";

float phi;

//Inicializar el serial con 115200 baudios

void setup() {

  Serial.begin(115200);

  lcd.init();// inicializamos el LCD
```

```
lcd.backlight(); //Encendemos la luz de fondo
```

```
lcd.setCursor(1, 0); //Esta linea pone el cursor en la posicion columna 1 y fila 0
```

```
lcd.print("Lectura del sensor"); //Imprimimos el mensaje
```

```
lcd.setCursor(6, 2); //Esta linea pone el cursor en la posicion columna 6 y fila 2
```

```
lcd.print("PZEM-016"); //Imprimimos el mensaje
```

```
delay(2000);
```

```
lcd.clear(); //Esta linea limpia la pantalla
```

```
lcd.setCursor(2, 0); //Esta linea pone el cursor en la posicion columna 2 y fila 0
```

```
lcd.print("Valores sensados"); //Imprimimos el mensaje
```

```
//Valores izquierda(V,P,F)
```

```
lcd.setCursor(0, 1); //Esta linea pone el cursor en la posicion columna 0 y fila 1
```

```
lcd.print("V:"); //Imprimimos el parametro V
```

```
lcd.setCursor(2, 1); //Esta linea pone el cursor en la posicion columna 2 y fila 1
```

```
lcd.print("****");//Imprimimos una cadena de caracteres donde se imprimirá el valor de la variable
```

```
lcd.setCursor(0, 2); //Esta linea pone el cursor en la posicion columna 0 y fila 2
```

```
lcd.print("P:");//Imprimimos el parametro P
```

```
lcd.setCursor(2, 2); //Esta linea pone el cursor en la posicion columna 2 y fila 2
```

```
lcd.print("****");//Imprimimos una cadena de caracteres donde se imprimirá el valor de la variable
```

```
lcd.setCursor(0, 3); //Esta linea pone el cursor en la posicion columna 0 y fila 3  
lcd.print("F:"); //Imprimimos el parametro F  
lcd.setCursor(2, 3); //Esta linea pone el cursor en la posicion columna 2 y fila 3  
lcd.print("****"); //Imprimimos una cadena de caracteres donde se imprimirá el valor de la  
variable
```

```
//Valores Derecha (I,E,FP)
```

```
lcd.setCursor(11, 1); //Esta linea pone el cursor en la posicion columna 11 y fila 1  
lcd.print("I:"); //Imprimimos el parametro I  
lcd.setCursor(13, 1); //Esta linea pone el cursor en la posicion columna 13 y fila 1  
lcd.print("****"); //Imprimimos una cadena de caracteres donde se imprimirá el valor de la  
variable
```

```
lcd.setCursor(11, 2); //Esta linea pone el cursor en la posicion columna 11 y fila 2  
lcd.print("E:"); //Imprimimos el parametro E  
lcd.setCursor(13, 2); //Esta linea pone el cursor en la posicion columna 13 y fila 2  
lcd.print("****"); //Imprimimos una cadena de caracteres donde se imprimirá el valor de la  
variable
```

```
lcd.setCursor(11, 3); //Esta linea pone el cursor en la posicion columna 11 y fila 3

lcd.print("FP:");//Imprimimos el parametro FP

lcd.setCursor(14, 3); //Esta linea pone el cursor en la posicion columna 14 y fila 3

lcd.print("****");//Imprimimos una cadena de caracteres donde se imprimirá el valor de la
variable

}
```

```
//Lecturas de cada parametro del sensor
```

```
void loop() {
```

```
float voltage = pzem.voltage();//Creamos una variable del tipo flotante para leer la variable del
voltaje
```

```
if (voltage != NAN) {
```

```
    //Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("V:");
```

```
} else {
```

```
    Serial.println("Error reading voltage");
```

```
}
```

```
//Reemplazar respuesta nan por 0
```

```
if (isnan(voltage)) {
```

```
voltage = 0;  
}
```

float current = pzem.current();//Creamos una variable del tipo flotante para leer la variable de la corriente

```
if (current != NAN) {  
    //Serial.print("Current: "); Serial.print(current); Serial.println("A");  
} else {  
    Serial.println("Error reading current");  
}
```

float power = pzem.power();//Creamos una variable del tipo flotante para leer la variable potencia

```
if (current != NAN) {  
    //Serial.print("Power: "); Serial.print(power); Serial.println("W");  
} else {  
    Serial.println("Error reading power");  
}
```

float energy = pzem.energy();//Creamos una variable del tipo flotante para leer la variable Energia

```
if (energy != NAN) {  
    //Serial.print("Energy: "); Serial.print(energy,3); Serial.println("kWh");
```

```
} else {  
    Serial.println("Error reading energy");  
}
```

```
float frequency = pzem.frequency();//Creamos una variable del tipo flotante para leer la  
variable frecuencia
```

```
if (current != NAN) {  
    //Serial.print("Frequency: "); Serial.print(frequency, 1); Serial.println("Hz");  
} else {  
    Serial.println("Error reading frequency");  
}
```

```
float pf = pzem.pf();//Creamos una variable del tipo flotante para leer la variable factor de  
potencia
```

```
if (current != NAN) {  
    //Serial.print("PF: "); Serial.println(pf);  
} else {  
    Serial.println("Error reading frequency");  
}
```

```
Serial.println();
```

```
delay(2500);
```

```
//Muestra de datos en LCD
```

```
lcd.setCursor(2, 0);

lcd.print("Valores sensados");

//Valores izquierda(V,P,F)

lcd.setCursor(0, 1); //Esta linea pone el cursor en la posicion columna 0 y fila 1
lcd.print("V:"); //Imprimimos el parametro V
lcd.setCursor(2, 1); //Esta linea pone el cursor en la posicion columna 2 y fila 1
lcd.print(voltage, 1); //Imprimimos el valor de la variable "voltage"
lcd.print("V "); //Imprimimos la unidad de medida

lcd.setCursor(0, 2); //Esta linea pone el cursor en la posicion columna 0 y fila 2
lcd.print("P:"); //Imprimimos el parametro P
lcd.setCursor(2, 2); //Esta linea pone el cursor en la posicion columna 2 y fila 2
lcd.print(power); //Imprimimos el valor de la variable "power"
lcd.print("W "); //Imprimimos la unidad de medida

lcd.setCursor(0, 3); //Esta linea pone el cursor en la posicion columna 0 y fila 3
lcd.print("F:"); //Imprimimos el parametro F
lcd.setCursor(2, 3); //Esta linea pone el cursor en la posicion columna 2 y fila 3
lcd.print(frequency); //Imprimimos el valor de la variable "frequency"
lcd.print("Hz "); //Imprimimos la unidad de medida
```

```
//Valores Derecha (I,E,FP)
```

```
lcd.setCursor(11, 1); //Esta linea pone el cursor en la posicion columna 11 y fila 1
```

```
lcd.print("I:"); //Imprimimos el parametro I
```

```
lcd.setCursor(13, 1); //Esta linea pone el cursor en la posicion columna 13 y fila 1
```

```
lcd.print(current); //Imprimimos el valor de la variable "current"
```

```
lcd.print("A "); //Imprimimos la unidad de medida
```

```
lcd.setCursor(11, 2); //Esta linea pone el cursor en la posicion columna 11 y fila 2
```

```
lcd.print("E:"); //Imprimimos el parametro E
```

```
lcd.setCursor(13, 2); //Esta linea pone el cursor en la posicion columna 13 y fila 2
```

```
lcd.print(energy); //Imprimimos el valor de la variable "energy"
```

```
lcd.print("kWh "); //Imprimimos la unidad de medida
```

```
lcd.setCursor(11, 3); //Esta linea pone el cursor en la posicion columna 11 y fila 3
```

```
lcd.print("FP:"); //Imprimimos el parametro FP
```

```
lcd.setCursor(14, 3); //Esta linea pone el cursor en la posicion columna 14 y fila 3
```

```
lcd.print(pf); //Imprimimos el valor de la variable "pf"
```

```
lcd.print(" "); //Imprimimos la unidad de medida
```

```
//Calculo de phi
```

```
phi = acos(power/(voltage*current));
```

```
if (isnan(phi)) {  
    phi = 0;  
}  
  
//Serial.println(phi);  
  
//Envio de datos por serial  
  
valores.concat(String(voltage));  
valores.concat(",");  
valores.concat(String(current));  
valores.concat(",");  
valores.concat(String(power));  
valores.concat(",");  
valores.concat(String(energy));  
valores.concat(",");  
valores.concat(String(frequency));  
valores.concat(",");  
valores.concat(String(pf));  
valores.concat(",");  
valores.concat(String(phi));  
  
// Mostrar la cadena por pantalla
```

```

Serial.println(valores);

valores = "";

delay(250);

}

```

## Anexo B: Programación para HMI

```

classdef Scada_Control_Barras_Potencia < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure matlab.ui.Figure
    SENSOR1Panel matlab.ui.container.Panel
    Lamp matlab.ui.control.Lamp
    CALabel matlab.ui.control.Label
    B1_C matlab.ui.control.NumericEditField
    Lamp_2 matlab.ui.control.Lamp
    VVLabel matlab.ui.control.Label
    B1_V matlab.ui.control.NumericEditField
    Lamp_3 matlab.ui.control.Lamp
    PWLabel_7 matlab.ui.control.Label
    B1_P matlab.ui.control.NumericEditField
    Lamp_15 matlab.ui.control.Lamp
    Lamp_16 matlab.ui.control.Lamp
    Lamp_17 matlab.ui.control.Lamp
    fHzLabel matlab.ui.control.Label
    B1_F matlab.ui.control.NumericEditField
    EWhLabel_7 matlab.ui.control.Label
    B1_E matlab.ui.control.NumericEditField
    FPLabel matlab.ui.control.Label
    B1_FP matlab.ui.control.NumericEditField
    Lamp_30 matlab.ui.control.Lamp
    Label matlab.ui.control.Label
    B1_ANG matlab.ui.control.NumericEditField
    UNIVERSIDADTCNICADELNORTELabel matlab.ui.control.Label
    INGENIERAENMANTENIMIENTOELCTRICOLabel matlab.ui.control.Label
    IniciarComunicacinPanel matlab.ui.container.Panel
    ConectarButton matlab.ui.control.StateButton
    Lamp_13 matlab.ui.control.Lamp

```

```
DetenerComunicacinPanel matlab.ui.container.Panel
DesconectarButton matlab.ui.control.StateButton
Lamp_14 matlab.ui.control.Lamp
SENSOR2Panel matlab.ui.container.Panel
Lamp_18 matlab.ui.control.Lamp
CALabel_2 matlab.ui.control.Label
B2_C matlab.ui.control.NumericEditField
Lamp_19 matlab.ui.control.Lamp
VLabel_2 matlab.ui.control.Label
B2_V matlab.ui.control.NumericEditField
Lamp_20 matlab.ui.control.Lamp
PWLabel_8 matlab.ui.control.Label
B2_P matlab.ui.control.NumericEditField
Lamp_21 matlab.ui.control.Lamp
Lamp_22 matlab.ui.control.Lamp
Lamp_23 matlab.ui.control.Lamp
fHzLabel_2 matlab.ui.control.Label
B2_F matlab.ui.control.NumericEditField
EWhLabel_8 matlab.ui.control.Label
B2_E matlab.ui.control.NumericEditField
FPLabel_2 matlab.ui.control.Label
B2_FP matlab.ui.control.NumericEditField
Lamp_31 matlab.ui.control.Lamp
Label_2 matlab.ui.control.Label
B2_ANG matlab.ui.control.NumericEditField
SENSOR3Panel matlab.ui.container.Panel
Lamp_24 matlab.ui.control.Lamp
CALabel_3 matlab.ui.control.Label
B3_C matlab.ui.control.NumericEditField
Lamp_25 matlab.ui.control.Lamp
VLabel_3 matlab.ui.control.Label
B3_V matlab.ui.control.NumericEditField
Lamp_26 matlab.ui.control.Lamp
PWLabel_9 matlab.ui.control.Label
B3_P matlab.ui.control.NumericEditField
Lamp_27 matlab.ui.control.Lamp
Lamp_28 matlab.ui.control.Lamp
Lamp_29 matlab.ui.control.Lamp
fHzLabel_3 matlab.ui.control.Label
B3_F matlab.ui.control.NumericEditField
EWhLabel_9 matlab.ui.control.Label
B3_E matlab.ui.control.NumericEditField
FPLabel_3 matlab.ui.control.Label
B3_FP matlab.ui.control.NumericEditField
Lamp_32 matlab.ui.control.Lamp
Label_3 matlab.ui.control.Label
B3_ANG matlab.ui.control.NumericEditField
end
```

```

properties (Access = private)
com; % creamos una propiedad para asignar los datos del puerto serie
estado;
end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
app.com = serial('COM5','BaudRate',115200,'Terminator','CR/LF');% Iniciamos la
comunicacion serial
warning('on','MATLAB:serial:fscanf:unsuccessfulRead')%Aviso si no se puede
inicial la comunicacion Serial
app.Lamp_13.Color='red'; % inicializamos los leds en rojo
app.Lamp_14.Color='green';
end

% Value changed function: B1_C
function B1_CValueChanged(app, event)
end

% Value changed function: ConectarButton
function ConectarButtonValueChanged(app, event)
app.Lamp_13.Color='green'; % inicializamos los leds en rojo
app.Lamp_14.Color='red';
fopen(app.com);
app.estado=1;
count = 1;
DATA =[10,21];
%%pause(4)
while(true)
if(app.estado==1)
datos = fscanff(app.com);
values = strsplit(datos,',' );
%% BARRA 1
%voltaje
val=cell2mat(values(1));
app.B1_V.Value =str2double(val);
if app.B1_V.Value == 0
app.Lamp_2.Color='red';
else

```

```

app.Lamp_2.Color='green'
end
%corriente
val=cell2mat(values(2));
app.B1_C.Value =str2double(val);
if app.B1_C.Value == 0
app.Lamp.Color='red';
else
app.Lamp.Color='green'
end
%potencia
val=cell2mat(values(3));
app.B1_P.Value =str2double(val);
if app.B1_P.Value == 0
app.Lamp_3.Color='red';
else
app.Lamp_3.Color='green'
end
%energia
val=cell2mat(values(4));
app.B1_E.Value =str2double(val);
if app.B1_E.Value == 0
app.Lamp_16.Color='red';
else
app.Lamp_16.Color='green'
end
%frecuencia
val=cell2mat(values(5));
app.B1_F.Value =str2double(val);
if app.B1_F.Value == 0
app.Lamp_15.Color='red';
else
app.Lamp_15.Color='green'
end
%factor de potencia
val=cell2mat(values(6));
app.B1_FP.Value =str2double(val);
if app.B1_FP.Value == 0
app.Lamp_17.Color='red';
else
app.Lamp_17.Color='green'
end
%angulo
val=cell2mat(values(7));
app.B1_ANG.Value =str2double(val);
if app.B1_ANG.Value == 0
app.Lamp_30.Color='red';
else

```

```

app.Lamp_30.Color='green'
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BARRA 2
%voltaje
val=cell2mat(values(8));
app.B2_V.Value =str2double(val);
if app.B2_V.Value == 0
app.Lamp_19.Color='red';
else
app.Lamp_19.Color='green'
end
%corriente
val=cell2mat(values(9));
app.B2_C.Value =str2double(val);
if app.B2_C.Value == 0
app.Lamp_18.Color='red';
else
app.Lamp_18.Color='green'
end
%potencia
val=cell2mat(values(10));
app.B2_P.Value =str2double(val);
if app.B2_P.Value == 0
app.Lamp_20.Color='red';
else
app.Lamp_20.Color='green'
end
%energia
val=cell2mat(values(11));
app.B2_E.Value =str2double(val);
if app.B2_E.Value == 0
app.Lamp_22.Color='red';
else
app.Lamp_22.Color='green'
end
%frecuencia
val=cell2mat(values(12));
app.B2_F.Value =str2double(val);
if app.B2_F.Value == 0
app.Lamp_21.Color='red';
else
app.Lamp_21.Color='green'
end
%factor de potencia
val=cell2mat(values(13));
app.B2_FP.Value =str2double(val);
if app.B2_FP.Value == 0

```

```

app.Lamp_23.Color='red';
else
app.Lamp_23.Color='green'
end
%angulo
val=cell2mat(values(14));
app.B2_ANG.Value =str2double(val);
if app.B2_ANG.Value == 0
app.Lamp_31.Color='red';
else
app.Lamp_31.Color='green'
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BARRA 3
%voltaje
val=cell2mat(values(15));
app.B3_V.Value =str2double(val);
if app.B3_V.Value == 0
app.Lamp_25.Color='red';
else
app.Lamp_25.Color='green'
end
%corriente
val=cell2mat(values(16));
app.B3_C.Value =str2double(val);
if app.B3_C.Value == 0
app.Lamp_24.Color='red';
else
app.Lamp_24.Color='green'
end
%potencia
val=cell2mat(values(17));
app.B3_P.Value =str2double(val);
if app.B3_P.Value == 0
app.Lamp_26.Color='red';
else
app.Lamp_26.Color='green'
end
%energia
val=cell2mat(values(18));
app.B3_E.Value =str2double(val);
if app.B3_E.Value == 0
app.Lamp_28.Color='red';
else
app.Lamp_28.Color='green'
end
%frecuencia
val=cell2mat(values(19));

```

```

app.B3_F.Value =str2double(val);
if app.B3_F.Value == 0
app.Lamp_27.Color='red';
else
app.Lamp_27.Color='green'
end
%factor de potencia
val=cell2mat(values(20));
app.B3_FP.Value =str2double(val);
if app.B3_FP.Value == 0
app.Lamp_29.Color='red';
else
app.Lamp_29.Color='green'
end
%angulo
val=cell2mat(values(21));
app.B3_ANG.Value =str2double(val);
if app.B3_ANG.Value == 0
app.Lamp_32.Color='red';
else
app.Lamp_32.Color='green'
end
%%*****
*****
%%*****
*****
if (count>100)
folder = "C:\Users\Toshiba\Documents\Tesis datos";
fecha = convertCharsToStrings(datestr(now, 'dd-mm-yyyy_HH_MM_SS'))
filename = sprintf("datos_%s.xls", fecha)
filepath = fullfile(folder, filename)
%tabla = array2table(DATA);
tabla =
array2table(DATA, 'VariableNames', {'Voltaje_1', 'Corriente_1', 'Potencia_1', 'Energia
_1', 'Frecuencia_1', 'Fp_1', 'Angulo_1', 'Voltaje_2', 'Corriente_2', 'Potencia_2', 'Ener
gia_2', 'Frecuencia_2', 'Fp_2', 'Angulo_2', 'Voltaje_3', 'Corriente_3', 'Potencia_3', 'E
nergia_3', 'Frecuencia_3', 'Fp_3', 'Angulo_3'});
writetable(tabla,filepath);
count=1;
DATA =[];
else
%% BARRA 1
%voltaje
val=cell2mat(values(1));
DATA(count,1)=str2double(val);
%corriente
val=cell2mat(values(2));
DATA(count,2) =str2double(val);

```

```

%energia
val=cell2mat(values(3));
DATA(count,3) =str2double(val);
%potencia
val=cell2mat(values(4));
DATA(count,4) =str2double(val);
%frecuencia
val=cell2mat(values(5));
DATA(count,5) =str2double(val);
%factor de potencia
val=cell2mat(values(6));
DATA(count,6) =str2double(val);
%angulo
val=cell2mat(values(7));
DATA(count,7)=str2double(val);
%% BARRA 2
%voltaje
val=cell2mat(values(8));
DATA(count,8) =str2double(val);
%corriente
val=cell2mat(values(9));
DATA(count,9) =str2double(val);
%energia
val=cell2mat(values(10));
DATA(count,10) =str2double(val);
%potencia
val=cell2mat(values(11));
DATA(count,11) =str2double(val);
%frecuencia
val=cell2mat(values(12));
DATA(count,12) =str2double(val);
%factor de potencia
val=cell2mat(values(13));
DATA(count,13) =str2double(val);
%angulo
val=cell2mat(values(14));
DATA(count,14) =str2double(val);
%% BARRA 3
%voltaje
val=cell2mat(values(15));
DATA(count,15) =str2double(val);
%corriente
val=cell2mat(values(16));
DATA(count,16) =str2double(val);
%energia
val=cell2mat(values(17));
DATA(count,17) =str2double(val);
%potencia

```

```

val=cell2mat(values(18));
DATA(count,18) =str2double(val);
%frecuencia
val=cell2mat(values(19));
DATA(count,19) =str2double(val);
%factor de potencia
val=cell2mat(values(20));
DATA(count,20) =str2double(val);
%angulo
val=cell2mat(values(21));
DATA(count,21) =str2double(val);
%disp(DATA)
count = count + 1;
end
pause(0.5)
else
break
end
end
end

% Value changed function: DesconectarButton
function DesconectarButtonValueChanged(app, event)
app.estado=0;
fclose(app.com);
app.Lamp_13.Color='red'; % inicializamos los leds en rojo
app.Lamp_14.Color='green';
end

% Close request function: UIFigure
function UIFigureCloseRequest(app, event)
delete(app.com);
delete(app);
end

% Size changed function: SENSOR1Panel
function SENSOR1PanelSizeChanged(app, event)
position = app.SENSOR1Panel.Position;
end

% Size changed function: SENSOR2Panel
function SENSOR2PanelSizeChanged(app, event)
position = app.SENSOR2Panel.Position;
end

```

```
end
```

```
% Component initialization  
methods (Access = private)
```

```
% Create UIFigure and components  
function createComponents(app)
```

```
% Create UIFigure and hide until all components are created
```

```
app.UIFigure = uifigure('Visible', 'off');  
app.UIFigure.Color = [0.8118 0.9804 0.9804];  
app.UIFigure.Colormap = [0.2431 0.149 0.6588;0.251 0.1647 0.7059;0.2588 0.1804  
0.7529;0.2627 0.1961 0.7961;0.2706 0.2157 0.8353;0.2745 0.2353 0.8706;0.2784  
0.2549 0.898;0.2784 0.2784 0.9216;0.2824 0.302 0.9412;0.2824 0.3216 0.9569;0.2784  
0.3451 0.9725;0.2745 0.3686 0.9843;0.2706 0.3882 0.9922;0.2588 0.4118  
0.9961;0.2431 0.4353 1;0.2196 0.4588 0.9961;0.1961 0.4863 0.9882;0.1843 0.5059  
0.9804;0.1804 0.5294 0.9686;0.1765 0.549 0.9529;0.1686 0.5686 0.9373;0.1529  
0.5922 0.9216;0.1451 0.6078 0.9098;0.1373 0.6275 0.898;0.1255 0.6471  
0.8902;0.1098 0.6627 0.8745;0.0941 0.6784 0.8588;0.0706 0.6941 0.8392;0.0314  
0.7098 0.8157;0.0039 0.7216 0.7922;0.0078 0.7294 0.7647;0.0431 0.7412  
0.7412;0.098 0.749 0.7137;0.1412 0.7569 0.6824;0.1725 0.7686 0.6549;0.1922 0.7765  
0.6235;0.2157 0.7843 0.5922;0.2471 0.7922 0.5569;0.2902 0.7961 0.5176;0.3412 0.8  
0.4784;0.3922 0.8039 0.4353;0.4471 0.8039 0.3922;0.5059 0.8 0.349;0.5608 0.7961  
0.3059;0.6157 0.7882 0.2627;0.6706 0.7804 0.2235;0.7255 0.7686 0.1922;0.7725  
0.7608 0.1647;0.8196 0.749 0.1529;0.8627 0.7412 0.1608;0.902 0.7333 0.1765;0.9412  
0.7294 0.2118;0.9725 0.7294 0.2392;0.9961 0.7451 0.2353;0.9961 0.7647  
0.2196;0.9961 0.7882 0.2039;0.9882 0.8118 0.1882;0.9804 0.8392 0.1765;0.9686  
0.8627 0.1647;0.9608 0.8902 0.1529;0.9608 0.9137 0.1412;0.9647 0.9373  
0.1255;0.9686 0.9608 0.1059;0.9765 0.9843 0.0824];  
app.UIFigure.Position = [100 100 851 539];  
app.UIFigure.Name = 'UI Figure';  
app.UIFigure.CloseRequestFcn = createCallbackFcn(app, @UIFigureCloseRequest,  
true);
```

```
% Create SENSOR1Panel
```

```
app.SENSOR1Panel = uipanel(app.UIFigure);  
app.SENSOR1Panel.TitlePosition = 'centertop';  
app.SENSOR1Panel.Title = 'SENSOR 1';  
app.SENSOR1Panel.BackgroundColor = [0.4196 0.7255 0.9294];  
app.SENSOR1Panel.SizeChangedFcn = createCallbackFcn(app,  
@SENSOR1PanelSizeChanged, true);  
app.SENSOR1Panel.FontWeight = 'bold';  
app.SENSOR1Panel.Position = [9 28 247 314];
```

```

% Create Lamp
app.Lamp = uilamp(app.SENSOR1Panel);
app.Lamp.Position = [176 205 27 27];

% Create CALabel
app.CALabel = uilabel(app.SENSOR1Panel);
app.CALabel.HorizontalAlignment = 'right';
app.CALabel.FontWeight = 'bold';
app.CALabel.Position = [16 207 31 22];
app.CALabel.Text = 'C(A)';

% Create B1_C
app.B1_C = uieditfield(app.SENSOR1Panel, 'numeric');
app.B1_C.ValueChangedFcn = createCallbackFcn(app, @B1_CValueChanged, true);
app.B1_C.Editable = 'off';
app.B1_C.FontWeight = 'bold';
app.B1_C.Position = [62 207 62 22];

% Create Lamp_2
app.Lamp_2 = uilamp(app.SENSOR1Panel);
app.Lamp_2.Position = [176 248 27 27];

% Create VVLabel
app.VVLabel = uilabel(app.SENSOR1Panel);
app.VVLabel.HorizontalAlignment = 'right';
app.VVLabel.FontWeight = 'bold';
app.VVLabel.Position = [18 250 29 22];
app.VVLabel.Text = 'V(V)';

% Create B1_V
app.B1_V = uieditfield(app.SENSOR1Panel, 'numeric');
app.B1_V.Editable = 'off';
app.B1_V.FontWeight = 'bold';
app.B1_V.Position = [62 250 62 22];

% Create Lamp_3
app.Lamp_3 = uilamp(app.SENSOR1Panel);
app.Lamp_3.Position = [176 166 27 27];

% Create PWLabel_7
app.PWLabel_7 = uilabel(app.SENSOR1Panel);
app.PWLabel_7.HorizontalAlignment = 'right';

```

```

app.PWLabel_7.FontWeight = 'bold';
app.PWLabel_7.Position = [14 168 33 22];
app.PWLabel_7.Text = 'P(W)';

% Create B1_P
app.B1_P = uieditfield(app.SENSOR1Panel, 'numeric');
app.B1_P.Editable = 'off';
app.B1_P.FontWeight = 'bold';
app.B1_P.Position = [62 168 62 22];

% Create Lamp_15
app.Lamp_15 = uilamp(app.SENSOR1Panel);
app.Lamp_15.Position = [176 80 27 27];

% Create Lamp_16
app.Lamp_16 = uilamp(app.SENSOR1Panel);
app.Lamp_16.Position = [176 122 27 27];

% Create Lamp_17
app.Lamp_17 = uilamp(app.SENSOR1Panel);
app.Lamp_17.Position = [176 42 27 27];

% Create fHzLabel
app.fHzLabel = uilabel(app.SENSOR1Panel);
app.fHzLabel.HorizontalAlignment = 'right';
app.fHzLabel.FontWeight = 'bold';
app.fHzLabel.Position = [15 82 32 22];
app.fHzLabel.Text = 'f(Hz)';

% Create B1_F
app.B1_F = uieditfield(app.SENSOR1Panel, 'numeric');
app.B1_F.Editable = 'off';
app.B1_F.FontWeight = 'bold';
app.B1_F.Position = [62 82 62 22];

% Create EWhLabel_7
app.EWhLabel_7 = uilabel(app.SENSOR1Panel);
app.EWhLabel_7.HorizontalAlignment = 'right';
app.EWhLabel_7.FontWeight = 'bold';
app.EWhLabel_7.Position = [4 124 43 22];
app.EWhLabel_7.Text = 'E(W/h)';

```

```

% Create B1_E
app.B1_E = uicontrolfield(app.SENSOR1Panel, 'numeric');
app.B1_E.Editable = 'off';
app.B1_E.FontWeight = 'bold';
app.B1_E.Position = [62 124 62 22];

% Create FPLabel
app.FPLabel = uicontrol(app.SENSOR1Panel);
app.FPLabel.HorizontalAlignment = 'right';
app.FPLabel.FontWeight = 'bold';
app.FPLabel.Position = [22 44 25 22];
app.FPLabel.Text = 'FP';

% Create B1_FP
app.B1_FP = uicontrolfield(app.SENSOR1Panel, 'numeric');
app.B1_FP.Editable = 'off';
app.B1_FP.FontWeight = 'bold';
app.B1_FP.Position = [62 44 62 22];

% Create Lamp_30
app.Lamp_30 = uicontrol(app.SENSOR1Panel);
app.Lamp_30.Position = [176 8 27 27];

% Create Label
app.Label = uicontrol(app.SENSOR1Panel);
app.Label.HorizontalAlignment = 'right';
app.Label.FontWeight = 'bold';
app.Label.Position = [22 10 25 22];
app.Label.Text = ' $\theta(^{\circ})$ ';

% Create B1_ANG
app.B1_ANG = uicontrolfield(app.SENSOR1Panel, 'numeric');
app.B1_ANG.Editable = 'off';
app.B1_ANG.FontWeight = 'bold';
app.B1_ANG.Position = [62 10 62 22];

% Create UNIVERSIDADTCNICADELNORTELabel
app.UNIVERSIDADTCNICADELNORTELabel = uicontrol(app.UIFigure);
app.UNIVERSIDADTCNICADELNORTELabel.FontSize = 20;
app.UNIVERSIDADTCNICADELNORTELabel.FontWeight = 'bold';
app.UNIVERSIDADTCNICADELNORTELabel.FontColor = [1 0 0];
app.UNIVERSIDADTCNICADELNORTELabel.Position = [213 469 365 24];
app.UNIVERSIDADTCNICADELNORTELabel.Text = 'UNIVERSIDAD TÉCNICA DEL NORTE';

```

```

% Create INGENIERAENMANTENIMIENTOELCTRICOLabel
app.INGENIERAENMANTENIMIENTOELCTRICOLabel = uilabel(app.UIFigure);
app.INGENIERAENMANTENIMIENTOELCTRICOLabel.FontWeight = 'bold';
app.INGENIERAENMANTENIMIENTOELCTRICOLabel.FontColor = [0 0 1];
app.INGENIERAENMANTENIMIENTOELCTRICOLabel.Position = [246 444 276 22];
app.INGENIERAENMANTENIMIENTOELCTRICOLabel.Text = 'INGENIERÍA EN MANTENIMIENTO
ELÉCTRICO';

```

```

% Create IniciarComunicacinPanel
app.IniciarComunicacinPanel = uipanel(app.UIFigure);
app.IniciarComunicacinPanel.TitlePosition = 'centertop';
app.IniciarComunicacinPanel.Title = 'Iniciar Comunicación';
app.IniciarComunicacinPanel.FontWeight = 'bold';
app.IniciarComunicacinPanel.Position = [72 370 140 77];

```

```

% Create ConectarButton
app.ConectarButton = uibutton(app.IniciarComunicacinPanel, 'state');
app.ConectarButton.ValueChangedFcn = createCallbackFcn(app,
@ConectarButtonValueChanged, true);
app.ConectarButton.Text = 'Conectar';
app.ConectarButton.FontName = 'Cooper Black';
app.ConectarButton.FontSize = 14;
app.ConectarButton.FontWeight = 'bold';
app.ConectarButton.FontColor = [0.0549 0.5412 0.0549];
app.ConectarButton.Position = [21 28 100 25];

```

```

% Create Lamp_13
app.Lamp_13 = uilamp(app.IniciarComunicacinPanel);
app.Lamp_13.Position = [60 1 20 20];

```

```

% Create DetenerComunicacinPanel
app.DetenerComunicacinPanel = uipanel(app.UIFigure);
app.DetenerComunicacinPanel.TitlePosition = 'centertop';
app.DetenerComunicacinPanel.Title = 'Detener Comunicación';
app.DetenerComunicacinPanel.FontWeight = 'bold';
app.DetenerComunicacinPanel.Position = [553 371 140 77];

```

```

% Create DesconectarButton
app.DesconectarButton = uibutton(app.DetenerComunicacinPanel, 'state');
app.DesconectarButton.ValueChangedFcn = createCallbackFcn(app,
@DesconectarButtonValueChanged, true);
app.DesconectarButton.Text = 'Desconectar';

```

```

app.DesconectarButton.FontName = 'Cooper Black';
app.DesconectarButton.FontSize = 14;
app.DesconectarButton.FontColor = [0.8196 0.051 0.051];
app.DesconectarButton.Position = [19 26 103 25];

% Create Lamp_14
app.Lamp_14 = uilamp(app.DetenerComunicacinPanel);
app.Lamp_14.Position = [63 1 20 20];

% Create SENSOR2Panel
app.SENSOR2Panel = uipanel(app.UIFigure);
app.SENSOR2Panel.TitlePosition = 'centertop';
app.SENSOR2Panel.Title = 'SENSOR 2';
app.SENSOR2Panel.BackgroundColor = [0.4196 0.7255 0.9294];
app.SENSOR2Panel.SizeChangedFcn = createCallbackFcn(app,
@SENSOR2PanelSizeChanged, true);
app.SENSOR2Panel.FontWeight = 'bold';
app.SENSOR2Panel.Position = [307 28 247 314];

% Create Lamp_18
app.Lamp_18 = uilamp(app.SENSOR2Panel);
app.Lamp_18.Position = [176 205 27 27];

% Create CALabel_2
app.CALabel_2 = uilabel(app.SENSOR2Panel);
app.CALabel_2.HorizontalAlignment = 'right';
app.CALabel_2.FontWeight = 'bold';
app.CALabel_2.Position = [16 207 31 22];
app.CALabel_2.Text = 'C(A)';

% Create B2_C
app.B2_C = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_C.Editable = 'off';
app.B2_C.FontWeight = 'bold';
app.B2_C.Position = [62 207 62 22];

% Create Lamp_19
app.Lamp_19 = uilamp(app.SENSOR2Panel);
app.Lamp_19.Position = [176 248 27 27];

% Create VVLabel_2
app.VVLabel_2 = uilabel(app.SENSOR2Panel);

```

```

app.VVLabel_2.HorizontalAlignment = 'right';
app.VVLabel_2.FontWeight = 'bold';
app.VVLabel_2.Position = [18 250 29 22];
app.VVLabel_2.Text = 'V(V)';

% Create B2_V
app.B2_V = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_V.Editable = 'off';
app.B2_V.FontWeight = 'bold';
app.B2_V.Position = [62 250 62 22];

% Create Lamp_20
app.Lamp_20 = uilamp(app.SENSOR2Panel);
app.Lamp_20.Position = [176 166 27 27];

% Create PWLabel_8
app.PWLabel_8 = uilabel(app.SENSOR2Panel);
app.PWLabel_8.HorizontalAlignment = 'right';
app.PWLabel_8.FontWeight = 'bold';
app.PWLabel_8.Position = [14 168 33 22];
app.PWLabel_8.Text = 'P(W)';

% Create B2_P
app.B2_P = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_P.Editable = 'off';
app.B2_P.FontWeight = 'bold';
app.B2_P.Position = [62 168 62 22];

% Create Lamp_21
app.Lamp_21 = uilamp(app.SENSOR2Panel);
app.Lamp_21.Position = [176 80 27 27];

% Create Lamp_22
app.Lamp_22 = uilamp(app.SENSOR2Panel);
app.Lamp_22.Position = [176 122 27 27];

% Create Lamp_23
app.Lamp_23 = uilamp(app.SENSOR2Panel);
app.Lamp_23.Position = [176 42 27 27];

% Create fHzLabel_2

```

```

app.fHzLabel_2 = uilabel(app.SENSOR2Panel);
app.fHzLabel_2.HorizontalAlignment = 'right';
app.fHzLabel_2.FontWeight = 'bold';
app.fHzLabel_2.Position = [15 82 32 22];
app.fHzLabel_2.Text = 'f(Hz)';

% Create B2_F
app.B2_F = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_F.Editable = 'off';
app.B2_F.FontWeight = 'bold';
app.B2_F.Position = [62 82 62 22];

% Create EWhLabel_8
app.EWhLabel_8 = uilabel(app.SENSOR2Panel);
app.EWhLabel_8.HorizontalAlignment = 'right';
app.EWhLabel_8.FontWeight = 'bold';
app.EWhLabel_8.Position = [11 124 43 22];
app.EWhLabel_8.Text = 'E(W/h)';

% Create B2_E
app.B2_E = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_E.Editable = 'off';
app.B2_E.FontWeight = 'bold';
app.B2_E.Position = [62 124 62 22];

% Create FPLabel_2
app.FPLabel_2 = uilabel(app.SENSOR2Panel);
app.FPLabel_2.HorizontalAlignment = 'right';
app.FPLabel_2.FontWeight = 'bold';
app.FPLabel_2.Position = [22 44 25 22];
app.FPLabel_2.Text = 'FP';

% Create B2_FP
app.B2_FP = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_FP.Editable = 'off';
app.B2_FP.FontWeight = 'bold';
app.B2_FP.Position = [62 44 62 22];

% Create Lamp_31
app.Lamp_31 = uilamp(app.SENSOR2Panel);
app.Lamp_31.Position = [176 8 27 27];

```

```

% Create Label_2
app.Label_2 = uilabel(app.SENSOR2Panel);
app.Label_2.HorizontalAlignment = 'right';
app.Label_2.FontWeight = 'bold';
app.Label_2.Position = [22 10 25 22];
app.Label_2.Text = ' $\theta(^{\circ})$ ';

% Create B2_ANG
app.B2_ANG = uieditfield(app.SENSOR2Panel, 'numeric');
app.B2_ANG.Editable = 'off';
app.B2_ANG.FontWeight = 'bold';
app.B2_ANG.Position = [62 10 62 22];

% Create SENSOR3Panel
app.SENSOR3Panel = uipanel(app.UIFigure);
app.SENSOR3Panel.TitlePosition = 'centertop';
app.SENSOR3Panel.Title = 'SENSOR 3';
app.SENSOR3Panel.BackgroundColor = [0.4196 0.7255 0.9294];
app.SENSOR3Panel.FontWeight = 'bold';
app.SENSOR3Panel.Position = [591 28 247 314];

% Create Lamp_24
app.Lamp_24 = uilamp(app.SENSOR3Panel);
app.Lamp_24.Position = [176 205 27 27];

% Create CALabel_3
app.CALabel_3 = uilabel(app.SENSOR3Panel);
app.CALabel_3.HorizontalAlignment = 'right';
app.CALabel_3.FontWeight = 'bold';
app.CALabel_3.Position = [16 207 31 22];
app.CALabel_3.Text = 'C(A)';

% Create B3_C
app.B3_C = uieditfield(app.SENSOR3Panel, 'numeric');
app.B3_C.Editable = 'off';
app.B3_C.FontWeight = 'bold';
app.B3_C.Position = [62 207 62 22];

% Create Lamp_25
app.Lamp_25 = uilamp(app.SENSOR3Panel);
app.Lamp_25.Position = [176 248 27 27];

```

```

% Create VVLabel_3
app.VVLabel_3 = uilabel(app.SENSOR3Panel);
app.VVLabel_3.HorizontalAlignment = 'right';
app.VVLabel_3.FontWeight = 'bold';
app.VVLabel_3.Position = [18 250 29 22];
app.VVLabel_3.Text = 'V(V)';

% Create B3_V
app.B3_V = uieditfield(app.SENSOR3Panel, 'numeric');
app.B3_V.Editable = 'off';
app.B3_V.FontWeight = 'bold';
app.B3_V.Position = [62 250 62 22];

% Create Lamp_26
app.Lamp_26 = uilamp(app.SENSOR3Panel);
app.Lamp_26.Position = [176 166 27 27];

% Create PWLabel_9
app.PWLabel_9 = uilabel(app.SENSOR3Panel);
app.PWLabel_9.HorizontalAlignment = 'right';
app.PWLabel_9.FontWeight = 'bold';
app.PWLabel_9.Position = [14 168 33 22];
app.PWLabel_9.Text = 'P(W)';

% Create B3_P
app.B3_P = uieditfield(app.SENSOR3Panel, 'numeric');
app.B3_P.Editable = 'off';
app.B3_P.FontWeight = 'bold';
app.B3_P.Position = [62 168 62 22];

% Create Lamp_27
app.Lamp_27 = uilamp(app.SENSOR3Panel);
app.Lamp_27.Position = [176 80 27 27];

% Create Lamp_28
app.Lamp_28 = uilamp(app.SENSOR3Panel);
app.Lamp_28.Position = [176 122 27 27];

% Create Lamp_29
app.Lamp_29 = uilamp(app.SENSOR3Panel);
app.Lamp_29.Position = [176 42 27 27];

```

```

% Create fHzLabel_3
app.fHzLabel_3 = uilabel(app.SENSOR3Panel);
app.fHzLabel_3.HorizontalAlignment = 'right';
app.fHzLabel_3.FontWeight = 'bold';
app.fHzLabel_3.Position = [15 82 32 22];
app.fHzLabel_3.Text = 'f(Hz)';

% Create B3_F
app.B3_F = uieditfield(app.SENSOR3Panel, 'numeric');
app.B3_F.Editable = 'off';
app.B3_F.FontWeight = 'bold';
app.B3_F.Position = [62 82 62 22];

% Create EWhLabel_9
app.EWhLabel_9 = uilabel(app.SENSOR3Panel);
app.EWhLabel_9.HorizontalAlignment = 'right';
app.EWhLabel_9.FontWeight = 'bold';
app.EWhLabel_9.Position = [4 124 43 22];
app.EWhLabel_9.Text = 'E(W/h)';

% Create B3_E
app.B3_E = uieditfield(app.SENSOR3Panel, 'numeric');
app.B3_E.Editable = 'off';
app.B3_E.FontWeight = 'bold';
app.B3_E.Position = [62 124 62 22];

% Create FPLabel_3
app.FPLabel_3 = uilabel(app.SENSOR3Panel);
app.FPLabel_3.HorizontalAlignment = 'right';
app.FPLabel_3.FontWeight = 'bold';
app.FPLabel_3.Position = [22 44 25 22];
app.FPLabel_3.Text = 'FP';

% Create B3_FP
app.B3_FP = uieditfield(app.SENSOR3Panel, 'numeric');
app.B3_FP.Editable = 'off';
app.B3_FP.FontWeight = 'bold';
app.B3_FP.Position = [62 44 62 22];

% Create Lamp_32
app.Lamp_32 = uilamp(app.SENSOR3Panel);
app.Lamp_32.Position = [176 8 27 27];

```

```

% Create Label_3
app.Label_3 = uilabel(app.SENSOR3Panel);
app.Label_3.HorizontalAlignment = 'right';
app.Label_3.FontWeight = 'bold';
app.Label_3.Position = [22 10 25 22];
app.Label_3.Text = ' $\theta(^{\circ})$ ';

% Create B3_ANG
app.B3_ANG = ueditfield(app.SENSOR3Panel, 'numeric');
app.B3_ANG.Editable = 'off';
app.B3_ANG.FontWeight = 'bold';
app.B3_ANG.Position = [62 10 62 22];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = Scada_Control_Barras_Potencia

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)

if nargin == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)

```

```

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end

```

### Anexo C: Programación de cálculo de error

```

%VALORES NOMINALES DEL SISTEMA
M1=120;
M2=118.3;
M3=600;
M4=900;

VN=120; %Voltaje base
IN=5.10; %Corriente base
PN=750; %Potencia base

%LOS VALORES ESTIMADOS INICIALES

Xi(1,1)=0; %Valor inicial del angulo en la barra 1
Xi(2,1)= 1; %Valor inicial del voltaje en la barra 1
Xi(3,1)=1; %Valor inicial voltaje en la barra 2
X=Xi;

Nestimados=size(Xi,1);

%VALORES MEDIDOS EN POR UNIDAD PARA UN SISTEMA DE 110V
%Voltaje PU/Voltaje nominal 116.4/116V y potencia nominal de 598 W y
%corriente nominal 5.10

%Valores medidos a ser colocados en el vector z

V2=M1/VN;          z(1,1)=V2;
V1=M2/VN;          z(2,1)=V1;
P1=M3/PN;          z(3,1)=P1;
P2=M4/PN;          z(4,1)=P2;

Nmediciones=size(z,1);

for i = 1:Nmediciones
    if z(i) < 0
        error('Valor negativo encontrado en la medicion %d', i)
    end
end
end
%LOS VALORES DE LAS VARIANZAS PARA CADA MEDICIÓN R_inv

```

```

VR_S=0.05;

VR(1)=(VR_S^2);
VR(2)=(VR_S^2);
VR(3)=(VR_S^2);
VR(4)=(VR_S^2);

for(i=1:Nmediciones)
    R(i,i)=VR(i);
    Rinv=(inv(R));
end

z(1,1)=V1;
z(1,1)=V2;
%LOS GRADOS DE LIBERTAD K Y EL INTERVALO DE CONFIANZA ALFA
k=Nmediciones-Nestimados;
alfa=0.01;

%SEGŪN LA TABLA 15.3 DEL LIBRO STEVENSON, LA DISTRIBUCIŪN JICUADRADA
X^2(k,ALFA)
X_K_ALFA=6.64;

%LA SUMA DE LOS CUADRADOS DE LOS ERRORES
f=0;
ITERMAX=100;
ERRMAX=1e-5;

for j=1:ITERMAX

%LAS MEDICIONES CALCULADAS INICIALES

h(1,1)=X(2,1);
h(2,1)=X(3,1);
h(3,1)=((6.41*(X(3,1))*(X(2,1))))*sin(-X(1,1));
h(4,1)=((6.41*(X(3,1))*(X(2,1))))*sin(X(1,1));

%LOS ERRORES INICIALES MEDIDOS
err=z-h;
%LA JACOBIANA DE LOS VALORES CALCULADOS INICIALES

Hx=[0                1
0
    0                0
1
    -6.41*X(3,1)*X(2,1)*cos(X(1,1))    -6.41*X(3,1)*sin(X(1,1))
6.41*X(2,1)*sin(X(1,1))
    6.41*X(3,1)*X(2,1)*cos(X(1,1))    6.41*X(3,1)*sin(X(1,1))
6.41*X(2,1)*sin(X(1,1))];

%LA MATRIZ DEGRADACIONES CALCULADA
Gx=Hx'*Rinv*Hx;

```

```

%SE CALCULAN LOS NUEVOS VALORES DE X HASTA LA CONVERGENCIA
X=X+(inv(Gx)*Hx'*Rinv*err);

if(abs(X-Xi)<ERRMAX)
    fprintf('SE REALIZARON %d ITERACIONES PARA LA CONVERGENCIA\n',j);
    %SE CALCULA LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES
for i=1:Nmediciones
    f=f+((err(i)^2)*(Rinv(i,i)));
end
    fprintf('LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES ES: f=%d\n',f);
    fprintf('LA DISTRIBUCIÓN CON:\n K=%d Y ALFA=%d PERMITE UN
f<=X^2(K,ALFA)<=%d\n',k,alfa,X_K_ALFA);
    if(f>X_K_ALFA)
        fprintf('POR LO QUE EXISTEN MEDICIONES ERRONEAS A SER
IDENTIFICADAS\n',f);
        break
    else
        fprintf('POR LO QUE NO EXISTEN MEDICIONES ERRONEAS A SER
IDENTIFICADAS\n',f);
        break
    end
end
Xi=X;
end

%SE CALCULA LA MATRIZ DE COVARIANZA
Rcovarianza=(eye(Nmediciones)-Hx*inv(Gx)*Hx'*Rinv)*R;
Rcovarianza(1,1)=1*V2;
Rcovarianza(2,2)=1*V1;
Rcovarianza(3,3)=1*P2;
Rcovarianza(4,4)=1*P1;

%Calculo del vector de errores ponderados
err(1,1)=1*V2;
err(2,1)=1*V1;
err(3,1)=1*P1;
err(4,1)=1*P2;

%SE CALCULAN LOS e'j CALCULADOS DE LOS ERRORES ESTANDARIZADOS
ej=zeros(Nmediciones,1);

for i=1:Nmediciones
    %Llenar los valores del vector ej para encontrar los min o max del error
    ej(i,1)=err(i,1)/sqrt(Rcovarianza(2,2));
end

%Condicional para encontrar los valores donde existe el mayor error
if min(ej) >= 0.00001 & min(ej)< 0.09

```

```

    fprintf('LUEGO DE CALCULAR LOS ERRORES ESTANDARIZADOS:\n EL ERROR MAS
GRANDE ESTA EN:ej (%d)=%d\n', find(ej==abs(min(ej))), abs(min(ej)));
    elseif max(ej)>0.1
        fprintf('LUEGO DE CALCULAR LOS ERRORES ESTANDARIZADOS:\n EL ERROR MAS
GRANDE ESTA EN:ej (%d)=%d\n', find(ej==abs(max(ej))), abs(max(ej)));
    end

for j=1:ITERMAX

%LAS MEDICIONES CALCULADAS INICIALES
h(1,1)=X(2,1);
h(2,1)=X(3,1);
h(3,1)=(6.41*(X(3,1))*(X(2,1)))*sin(-X(1,1));
h(4,1)=(6.41*(X(3,1))*(X(2,1)))*sin(X(1,1));

%SE VUELVE h CALCULADO = z MEDIDO,, ASÃ? EL ERROR ES CERO PARA ELIMINAR EL
DATO EN LOS CÃ?LCULOS
if min(ej) <= 0.1
    h(find(ej==abs(min(ej))),1)=z(find(ej==abs(min(ej))),1);;
elseif max(ej)>0.1
    h(find(ej==abs(max(ej))),1)=z(find(ej==abs(max(ej))),1);;
end
% h(find(ej==abs(max(ej))),1)=z(find(ej==abs(max(ej))),1);

%LOS ERRORES INICIALES MEDIDOS
err=z-h;
%LA JACOBIANA DE LOS VALORES CALCULADOS INICIALES

Hx2=[0                1
0
0                0
1
-6.41*X(3,1)*X(2,1)*cos(X(1,1))    -6.41*X(3,1)*sin(X(1,1))
6.41*X(2,1)*sin(X(1,1))
6.41*X(3,1)*X(2,1)*cos(X(1,1))    6.41*X(3,1)*sin(X(1,1))
6.41*X(2,1)*sin(X(1,1))            ];

%LA MATRIZ DEGANACIAS CALCULADA
Gx=Hx'*Rinv*Hx;

%SE CALCULAN LOS NUEVOS VALORES DE X HASTA LA CONVERGENCIA
X=X+(inv(Gx)*Hx'*Rinv*err);
%Condional para quitar la medicion donde existe el mayor error
if(abs(X-Xi)<ERRMAX)
    if min(ej) >= 0.00001 & min(ej)< 0.09
        fprintf('\n \n AL RECALCULAR SIN LA MEDICIÃ?N Z%d:\n
', find(ej==abs(min(ej))));
        fprintf('SE REALIZARON %d ITERACIONES PARA LA CONVERGENCIA\n',j);
    elseif max(ej)>0.1
        fprintf('\n \n AL RECALCULAR SIN LA MEDICIÃ?N Z%d:\n
', find(ej==abs(max(ej))));
        fprintf('SE REALIZARON %d ITERACIONES PARA LA CONVERGENCIA\n',j);
    end
end

```

```

%SE CALCULA LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES
f=0;
for i=1:Nmediciones-1

    f=f+((err(i)^2)*(Rinv(i,i)))/10000;

end
fprintf('LA SUMA PONDERADA DEL CUADRADO DE LOS ERRORES ES: f=%d\n',f);
fprintf('LA DISTRIBUCIÃ"N CON:\n K=%d Y ALFA=%d PERMITE UN
f<=X^2(K,ALFA)<=%d\n',k,alfa,X_K_ALFA);
if(f>X_K_ALFA)
    fprintf('POR LO QUE EXISTEN MEDICIONES ERRONEAS A SER
IDENTIFICADAS\n',f);
    break
else
    fprintf('POR LO QUE NO EXISTEN MEDICIONES ERRONEAS A SER
IDENTIFICADAS\n',f);
    break
end
end
Xi=X;
end

```