



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SISTEMA DE ADQUISICIÓN DE DATOS Y CONTROL SUPERVISADO DE MICRO-REDES EN CÓDIGO ABIERTO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

ANDRÉS ALEJANDRO GUERRA CAREAGA

PROFESOR GUÍA:
PATRICIO MENDOZA ARAYA

MIEMBROS DE LA COMISIÓN:
ALBERTO CASTRO ROJAS
FRANCISCO RIVERA SERRANO

SANTIAGO DE CHILE
2023

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: ANDRÉS ALEJANDRO GUERRA CAREAGA
FECHA: 2023
PROF. GUÍA: PATRICIO MENDOZA ARAYA

SISTEMA DE ADQUISICIÓN DE DATOS Y CONTROL SUPERVISADO DE MICRO-REDES EN CÓDIGO ABIERTO

En la última década, la generación distribuida de energía ha aumentado considerablemente debido a la reducción en los costos de inversión de paneles solares, lo cual, ha movilizó tanto a clientes residenciales como empresas pequeñas a invertir en este tipo de tecnología para ver reducciones en el costo de la energía eléctrica. Este avance tecnológico ha dado lugar a nuevas redes pequeñas en la que se concentran consumidores como generadores y derribó el paradigma de que los centros de consumos se encuentran alejados de los centros de generación.

Bajo este contexto, se presentan las micro-redes como pequeñas redes en donde existe un operador local que controla el balance de generación-carga dentro de esta, optimizando la operación para reducir costos, asegurando el abastecimiento de potencia y gestionando los recursos disponibles de la red. Otra de las capacidades de las micro-redes es operar en modo isla y conectadas a la red. Para llevar a cabo estas obligaciones, el operador debe tener acceso al estado de la red en todo momento y ser capaz de operar remotamente los distintos dispositivos que la componen.

Para llevar a cabo dicho cometido, es necesario utilizar una herramienta que se ha utilizado desde hace muchos años: los sistemas SCADA. Se ha identificado a través de una revisión de mercado, que los sistemas que se encuentran disponibles en el mercado no son capaces de satisfacer todas las necesidades de estas nuevas micro-redes o comúnmente conocidas como *Smart Grids*. Al adquirir un sistema en el mercado, se debe escoger en tener una buena visualización del estado de la red o que posea todas las capacidades de control.

En esta memoria se desarrolla un sistema SCADA escrito en el lenguaje de programación *Python*, que es capaz de monitorear en tiempo real las principales variables eléctricas de la micro-red en una interfaz simple e intuitiva para su configuración y operación. Este sistema tiene las aplicaciones de Historizador, capacidad de almacenar información histórica de la red para futuras referencias, y Gestor de Alarmas para llevar un monitoreo activo de las variables más críticas del sistema.

El sistema SCADA, implementando el protocolo de comunicación Modbus, es validado en un entorno de simulación, en el cual monitorea seis dispositivos simultáneamente. Se detecta que a pesar de obtener las mediciones correctamente, existe un retraso promedio de 7.64 [ms] en cada intervalo de medición, cuya causa radica en la programación misma. Posteriormente, es validado en el Laboratorio de Energía de la Universidad de Chile, conectándose a dispositivos de medida PowerLogic PM750, almacenando los datos de medición y permitiendo un monitoreo constante de los activos de la red.

Agradecimientos

Quiero iniciar agradeciendo a mi padre y a mi madre, que desde pequeño siempre me dieron todo el apoyo necesario y me inculcaron grandes valores que al día de hoy me acompañan. Gracias por darme la oportunidad de estudiar, todo el cariño que me dieron, por las conversaciones y estar siempre en mi vida. Gracias a mi hermano por ser un apoyo y siempre estar. Muchas gracias a ti, Lucy.

Luego agradecer a todos los amigos que hice en la Universidad, al Pollo, Diego, Oscar, Sergio, gracias a todas las horas y días que pasamos estudiando, todos los buenos momentos, todas las risas e historias que pasamos durante estos años. Sin ustedes, definitivamente que sacar esta carrera hubiera sido tres veces más difícil.

Agradecer también a mi novia, Bárbara, por las miles de veces que me brindo su apoyo y paciencia, por aprender de ti y tu de mi, por hacerme persistir y no rendirme por lo que si vale la pena.

Quiero agradecer también al profesor Patricio Mendoza, quien me guió en este proceso y que incontables veces me ayudo a despejar dudas y problemáticas que ocurrieron en el camino. Hasta el ultimo momento fue un gran apoyo.

Finalmente, gracias a todas las personas que conocí durante los años de estudios en la Universidad de Chile, que de alguna u otra manera me aportaron conocimientos y nuevas perspectivas.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Alcances	3
1.4. Estructura del Trabajo	3
2. Marco Teórico	4
2.1. Micro-red	4
2.1.1. Ventajas de una Micro-red	4
2.1.2. Dificultades en la Implementación	5
2.1.3. Arquitectura de Control	5
2.1.4. Control Jerárquico en Micro-redes	6
2.2. Protocolos de Comunicación	8
2.2.1. Modbus	8
2.2.1.1. Query-Response Cycle	8
2.2.1.2. Modbus en Puertos Seriales	9
2.2.1.3. Modbus con comunicación TCP/IP	10
2.2.2. IEC 60870-5	12
2.2.3. DNP 3	13
2.2.4. IEC 61850	14
2.2.5. Gateway	17
2.3. Sistemas SCADA	18
2.3.1. SIMATIC WinCC V7	18
2.3.2. eSCADA	20
2.3.3. EcoStruxure™ Power Operation	22
2.3.4. VTScada	23
2.3.5. Comparación Técnica	25
3. Arquitectura y Caso de Estudio	27
3.1. Arquitectura del Sistema	27
3.1.1. Aplicaciones del Sistema	28
3.1.2. Caso de Estudio	28
3.1.2.1. Topología de la Micro-red	28
3.1.2.2. Elementos de Red	29
3.1.2.3. Canales de Comunicación	29

3.1.2.4.	Dispositivos de Campo	30
4.	Desarrollo y resultados	31
4.1.	Librerías Utilizadas	31
4.2.	Interfaz Hombre-Máquina	31
4.2.1.	Objetos Gráficos	32
4.2.1.1.	FrameObject	32
4.2.1.2.	LineObject	33
4.2.2.	Objetos de Datos	33
4.2.2.1.	Paneles Fotovoltaicos	34
4.2.2.2.	Linea de Transmisión	34
4.2.2.3.	Subestaciones	36
4.2.2.4.	Transformadores	36
4.2.2.5.	Consumo	37
4.2.3.	Otros Objetos	37
4.2.3.1.	Alarma	37
4.2.3.2.	Alarm Manager	37
4.2.3.3.	Historical Data	38
4.2.4.	Base de Datos	38
4.2.4.1.	Datos de Red	38
4.2.4.2.	Datos de Medición	38
4.2.4.3.	Algoritmo de Adquisición de Datos	39
4.3.	Implementación de la Micro-red del Laboratorio	40
4.3.1.	Validación en un Entorno Simulado	44
4.3.2.	Validación Experimental en el Laboratorio	50
4.3.2.1.	Perfil de Tensión de la Micro-red	51
4.3.2.2.	Conexión de Carga Trifásica	52
5.	Conclusiones	55
5.1.	Trabajo Futuro	56
	Bibliografía	56
	Anexo	59

Índice de Tablas

2.1.	Evaluación de los productos SCADA	25
4.1.	Métodos de la clase <i>FrameObject</i>	32
4.2.	Métodos de la clase <i>LineObject</i>	33
4.3.	Métodos de la clase <i>Photovoltaic</i>	34
4.4.	Métodos de la clase <i>LineSegment</i>	35
4.5.	Métodos de la clase <i>Terminal</i>	36
4.6.	Métodos de la clase <i>Trafo</i>	36
4.7.	Métodos de la clase <i>Load</i>	37
4.8.	Métodos de la clase <i>Alarma</i>	37
4.9.	Métodos de la clase <i>Alarm Manager</i>	38
4.10.	Métodos de la clase <i>Historical Data</i>	38

Índice de Ilustraciones

2.1.	Control Jerárquico. Adaptada de [4]	7
2.2.	Ciclo de solicitud y respuesta del protocolo Modbus. Fuente: [6]	9
2.3.	Unidad de información del protocolo. Fuente: [7]	10
2.4.	Unidad de información en comunicación serial. Fuente: [7]	10
2.5.	Unidad de información en comunicación TCP/IP. Fuente: [8]	11
2.6.	Arquitectura de comunicación de una red Modbus TCP/IP. Fuente: [8]	11
2.7.	Estándares del protocolo IEC 60870-5 (imagen adaptada de [9]).	13
2.8.	Estándares del protocolo IEC 61850 (imagen adaptada de [10]).	15
2.9.	Imagen referencial de SCADA Data Gateway. Fuente: [11]	18
2.10.	Supervisión y control de proceso en WinCC de Siemens	20
2.11.	Supervisión y control de una micro-red en Microgrid Control de Siemens	20
2.12.	Supervisión y control de una micro-red con Electrical Scada de ETAP	21
2.13.	Supervisión y control de una micro-red con EcoStruxure Power Operation de SE	22
2.14.	Análisis fasorial de una falla en EcoStruxure Power Operation de SE	23
2.15.	Ejemplo de HMI en VTScada para una central generadora. Fuente: VTScada	24
3.1.	Topología de la micro-red en Universidad de Chile	29
4.1.	Diagrama UML - Clase FrameObject	32
4.2.	Diagrama UML - Clase LineObject	33
4.3.	Diagrama UML - Paneles Fotovoltaicos	34
4.4.	Diagrama UML - Segmento de Línea	35
4.5.	Estructura de datos almacenados en archivo .csv	39
4.6.	Diagrama de flujo del algoritmo de adquisición de datos	40
4.7.	Ventana de inicio	40
4.8.	Ventana principal	41
4.9.	Formulario para crear nuevo elemento	42
4.10.	Elemento en la ventana principal	43
4.11.	Micro-red del laboratorio implementada en el software	44
4.12.	Configuración de un esclavo Modbus simulado para la barra 1	45
4.13.	Monitoreando activamente la micro-red (simulación)	46
4.14.	Deteniendo el monitoreo de la micro-red (simulación)	47
4.15.	Potencia activa obtenida de los esclavos Modbus simulados	47
4.16.	Potencia reactiva obtenida de los esclavos Modbus simulados	48
4.17.	Diferencia entre intervalos de medición con respecto a 1000 [ms]	48
4.18.	Monitoreo de variables en tiempo real mediante gráfico	49
4.19.	Ventana del gestor de alarmas	49
4.20.	Activación de la alarma cuando la variable supera el umbral	50
4.21.	Perfil de tensión F-F de la micro-red	51
4.22.	Perfil de tensión F-N de la micro-red	51

4.23.	Potencia Activa, Potencia Reactiva y Tensión F-F en la Barra 4	52
4.24.	Potencia Activa, Potencia Reactiva y Frecuencia en la Barra 4	52
4.25.	Potencia Activa, Potencia Reactiva y Corriente Media en la Barra 4	53
4.26.	Potencia Activa, Potencia Reactiva y Factor de Potencia en la Barra 4	53
4.27.	Perfil de Tensión en Barra 1 y 4	54

Capítulo 1

Introducción

1.1. Motivación

En los últimos años, la instalación de centrales en base a energías renovables no convencionales (ERNC), como la generación eólica y solar, ha aumentado considerablemente debido a razones medioambientales, pero principalmente también por los bajos costos de producción que impactan en la inversión inicial de estas centrales. Durante la última década, las centrales eólicas y fotovoltaicas han aumentado anualmente en promedio en un 19 % y 49 %, respectivamente. [14].

En particular, las centrales solares tienen un mayor aumento en capacidad instalada que las centrales eólicas debido a que presentan un menor costo nivelado de la energía, el cual es un indicador que equivale al precio de venta de la energía de la central en donde el VAN de la evaluación del proyecto es igual a cero, pero también hay otros factores importantes de los cuales se destacan: la flexibilidad de instalación, pueden ser instalados en un techo de una casa o en la azotea de un edificio; la rapidez de construcción, existen casos en el mundo en el que se han alcanzado a construir 1 MW por día, mientras que las centrales eólicas pueden demorar hasta 24 meses; el recurso solar puede ser aprovechado en la mayoría del territorio nacional, mientras que existe un territorio más acotado en donde una instalación eólica es factible; y, por último, los estudios a realizar en una central eólica requieren más tiempo que de una central solar.

No solamente la gran industria ha aprovechado la caída de costos de los paneles solares, sino que también ha sido un factor habilitante para que medianas, pequeñas empresas y hasta clientes residenciales que están conectados al sistema de distribución, instalen paneles en sus propiedades y así reducir los costos de su consumo de electricidad. Esto también les permite inyectar los excedentes de energía al Sistema Eléctrico Nacional, como es el caso de centrales PMGD en Chile. Lo anterior, genera un cambio en el paradigma que se tenía una década atrás, en donde los centros de consumos se encontraban a largas distancias de las fuentes de generación.

En este contexto, se define y nacen las micro-redes, pequeñas redes que pueden estar conectadas a la red o aisladas de esta, y que poseen recursos energéticos distribuidos para abastecer localmente los consumos de la zona. Esto posee ventajas para el sistema, ya que aumenta la eficiencia energética al reducir las pérdidas por las líneas de transmisión (entre

otros beneficios), pero también pueden tener como objetivo abastecer zonas rurales que no tienen acceso al Sistema Eléctrico Nacional (SEN).

Para la correcta operación de una micro-red y los componentes de ella, es de vital importancia mantener un canal de comunicación en todo momento con los activos de la red, realizando un monitoreo constante y, si las condiciones lo ameritan, controlando de manera manual o automáticamente consignas de potencia, activación de interruptores o encendido de un grupo generador para mantener el abastecimiento de energía en situaciones de emergencia.

Para lograr lo anterior, es importante que la micro-red posea un sistema de adquisición de datos y control supervisado, comúnmente conocido como *Supervisory Control and Data Acquisition*, o por sus siglas en inglés, SCADA. Un sistema SCADA en el ámbito de Sistemas de Potencia se asocia al sistema que permite visualizar las variables eléctricas de la red, identificando zonas con perfiles de tensión fuera de rango, monitoreo de la frecuencia de la red eléctrica, entre otros. Este es un aspecto clave en cualquier operación de un SEP.

Existen productos SCADA en el mercado para la industria eléctrica, pero estos no son técnica ni económicamente viables a ser implementados en una micro-red de pequeña escala.

Por lo anterior, se propone realizar el diseño e implementación de un sistema SCADA que cumpla con los requisitos de una micro-red.

1.2. Objetivos

1.2.1. Objetivo General

Esta memoria busca abordar las necesidades de una micro-red en el ámbito del monitoreo, control y adquisición de datos y desarrollar un sistema que satisfaga dichas necesidades.

El objetivo general de la memoria es el desarrollo de un programa en un lenguaje de código abierto que pueda realizar la adquisición de datos, el monitoreo y control de los elementos de la red.

1.2.2. Objetivos Específicos

1. Identificar las funcionalidades y limitaciones de los sistemas SCADA que se utilizan globalmente en la industria.
2. Identificar los distintos protocolos de comunicación utilizados para el intercambio de información de equipos automatizados.
3. Establecer el intercambio de información entre dispositivos de una micro-red, utilizando protocolos estándares en la industria.
4. Diseñar un programa SCADA que permita la visualización y control en tiempo real de las variables eléctricas de los elementos de una micro-red.
5. Comparar el desempeño del programa con el sistema SCADA que se encuentra implementado en el laboratorio.

1.3. Alcances

El trabajo en esta memoria considera la realización del sistema SCADA de una micro-red, incluyendo la implementación de los protocolos de comunicación entre dispositivos. No se desarrollará un sistema de gestión de energía, aunque el programa si estará diseñado para permitir la posterior integración de uno. Los aspectos de ciberseguridad de la información no serán abordados en el programa.

1.4. Estructura del Trabajo

Este trabajo se dividirá en capítulos como se describe a continuación:

En el Capítulo 2 se presenta el marco teórico que concierne a esta memoria, en donde se expone las necesidades y cualidades de una micro-red, qué son los sistemas SCADA y qué se puede encontrar en el mercado, además de una revisión de los protocolos de comunicación comúnmente utilizados en la industria.

En el Capítulo 3 se presenta la propuesta metodológica, cómo se abordó el desarrollo y creación de un sistema SCADA que satisface las necesidades de las micro-redes.

En el Capítulo 4 se muestran los resultados obtenidos y el análisis de estos, al ser comparados con un sistema SCADA implementado en el laboratorio de la Universidad de Chile.

Por ultimo, en el Capítulo 5, se presentan las conclusiones de la memoria y lo que queda como trabajo futuro.

Capítulo 2

Marco Teórico

2.1. Micro-red

Una micro-red es una red eléctrica de baja tensión que, generalmente posee recursos energéticos distribuidos (DER) como paneles solares, centro de baterías y otros equipamientos de pequeña escala que logran abastecer el consumo de potencia de la zona que abarca de manera parcial o completa, pudiendo operar conectada o desconectada a la red. Por otro lado, existen varias definiciones de una micro-red. Por ejemplo, el Departamento de Energía de los Estados Unidos [1] lo define como *“un grupo interconectado de cargas y recursos distribuidos entre límites eléctricos bien definidos que actúan como una sola entidad controlada con respecto a la red. Una micro-red puede conectar y desconectarse desde la red para operar conectado a la red o en modo isla”*, definición que es bastante similar al que se encuentra en el estándar IEEE 2030.7.

No obstante, el New York State Energy Research and Development Authority (NYSER-DA) menciona [2] *“las micro-redes son redes locales de energía que son capaces de separarse de la red eléctrica principal durante condiciones climáticas extremas o durante emergencias, proveyendo potencia a clientes individuales y servicios públicos cruciales como hospitales, primeros auxilios, primeras intervenciones y centrales de tratamiento de aguas”*. Entre ambas definiciones no hay muchas diferencias, y coinciden en que una micro-red:

- puede operar de forma aislada de la red, entregando potencia a consumos locales y cruciales, de manera que aumenta la resiliencia y la confiabilidad;
- las unidades de generación distribuida dentro de ella actúan coordinadamente, de manera que aguas arriba se observa como una sola entidad controlada.

De esta manera, una micro-red puede ser desde una pequeña comunidad en baja tensión como un pueblo o área urbanizada, hasta consumos más específicos como hospitales, centros comerciales por nombrar algunos, que tienen múltiples unidades pequeñas de generación conectadas en ellas.

2.1.1. Ventajas de una Micro-red

El desarrollo de una micro-red trae ventajas tanto para el sistema eléctrico global como para los consumidores locales, cuyos beneficios dependerán de los métodos de control y el tipo de operación pero, en general, la micro-red provee:

- **Reducción de los costos de energía:** abasteciendo localmente los consumos de la micro-red, haciendo uso de los sistemas de batería, autos eléctricos y demandas controlables. Por ejemplo, haciendo uso de la información de los precios de energía de la red de utilidad pública, se genera una política de operación para las baterías, de modo que se cargan cuando los precios están bajos y se utilizan para abastecer la demanda cuando son mas altos. De la misma manera, aprovechar de generar y vender excesos de energía a la red frente a una alza en los precios.
- **Resiliencia:** actúa como central de respaldo frente a situaciones de emergencia para restaurar el servicio de energía a una infraestructura critica [5].
- **Sustentabilidad:** promueve la integración de las energías renovables y reduce las pérdidas por transmisión.
- **Optimización de las redes de distribución y transmisión:** mejora la infraestructura existente, retrasando obras para abastecer a la creciente demanda y mejorando la confiabilidad.

Adicionalmente, mejoran el perfil de tensión de la zona y la calidad del suministro, inclusive con la posibilidad de proveer servicios complementarios como, por mencionar algunos, la regulación de frecuencia y partidas en negro.

Dada la posibilidad de que las micro-redes sean instaladas en zonas rurales sin acceso a la red eléctrica, su implementación involucra un acercamiento y trabajo en conjunto con los locatarios.

2.1.2. Dificultades en la Implementación

Las desventajas o dificultades que se presentan en la implementación de una micro-red son las siguientes:

- **Operación:** debido a la aparición de distintas fuentes de generación de energía, resulta imperativo realizar la coordinación la operación dentro de la micro-red. Adicionalmente, puede significar un rediseño de las protecciones por potenciales inversiones de flujo y cambio en las corrientes de cortocircuito, originados por las inyecciones de potencia de los recursos distribuidos. Por último, el factor de potencia visto aguas arriba de la micro-red puede disminuir debido al aumento individual de la inyección de potencia activa dentro de esta, disminuyendo así la robustez en el punto de conexión común de la micro-red con la red eléctrica pública.
- **Comunicaciones:** para llevar a cabo la operación de la micro-red se necesita instalar infraestructura adicional que permita el intercambio de información entre los dispositivos de campo y un centro de control, que consolida la información y evalúa el estado de la red.

2.1.3. Arquitectura de Control

Los sistemas de control en una micro-red son parte importante en la operación al encargarse y asegurar que el suministro de la energía sea entregada de manera segura, de buena calidad y al menor costo posible. Lo anterior se logra: manteniendo el balance entre carga y

generación de manera interna de la micro-red; regulando la tensión en una banda admisible según la regulación vigente del país; gestionando reservas de potencia; entre otros. En general, la arquitectura de control de una red eléctrica es un componente esencial para monitorear, gestionar y optimizar la operación y los flujos de potencia.

Para llevar a cabo las funciones descritas con anterioridad, se han desarrollado tres métodos de control: el centralizado, el descentralizado y el distribuido [3]:

- Centralizado: la principal característica de esta estructura de control es que consiste en un único controlador que gestiona y opera los elementos de la red, entregando consignas de control a estos en función de la información recibida y las salidas del problema de optimización que resuelve. El control centralizado obtiene soluciones óptimas globales, a cambio del aumento de procesamiento de datos, la necesidad de contar con una infraestructura de comunicaciones eficaz, de lo contrario podría afectar la estabilidad por el retraso de la información, y reduce la escalabilidad.
- Descentralizado: es aquella estructura en la cual existen diversos equipos de control que no intercambian información entre ellos. Contrario al método de control centralizado, reduce el esfuerzo computacional, incrementa la escalabilidad y resiliencia, y no requiere de sistemas comunicacionales, pero obtiene soluciones sub-óptimas.
- Distribuido: esta estructura inicia de la base de un control descentralizado, pero en la cual si hay un grado de intercambio de información, aumentando así el desempeño de los controladores en las acciones de control. Esta estructura de control es un trade-off entre los dos métodos mencionados anteriormente. Se pueden obtener mejor resultados que en un control descentralizado, pero requiere mayor procesamiento computacional, infraestructura de comunicaciones, etc.

2.1.4. Control Jerárquico en Micro-redes

El control jerárquico se encuentra inspirado a partir de los sistemas de potencia convencionales, el cual esta compuesto por tres capas de control: el control primario, el control secundario y el control terciario. Cada una de las capas se diferencia por el tipo de control que realiza y los intervalos de respuesta que tienen.

El control primario es el que opera en la escala de tiempo de los milisegundos. Estos controladores, instalados generalmente en cada central generadora, buscan controlar la frecuencia y tensión en el punto de conexión a la micro-red, utilizando únicamente mediciones locales. Este trabajo lo realiza comúnmente un control descentralizado tipo *droop*, cuyo comportamiento obedece la siguiente expresión:

$$w = w^* - m(P - P^*)$$

$$V = V^* - n(Q - Q^*)$$

En donde w y w^* corresponden a la frecuencia medida y la frecuencia nominal, respectivamente; m la constante *droop* de frecuencia; P y P^* a la potencia activa medida y la consigna de potencia activa, respectivamente. En la segunda expresión, V y V^* corresponden a la tensión medida y la tensión nominal, respectivamente; n la constante *droop* de tensión;

Q y Q^* a la potencia reactiva medida y la consigna de potencia reactiva, respectivamente. Las expresiones anteriores son validas en redes inductivas (alta tensión), en donde la frecuencia se encuentra acoplada a la potencia activa, y la tensión a la potencia reactiva. En redes más resistivas (baja tensión), el acoplamiento se invierte, es decir, la potencia activa se encuentra más acoplada a la tensión, mientras que la potencia reactiva a la frecuencia.

El control secundario tiene como función llevar al sistema nuevamente a los puntos de operación nominales, cuyas desviaciones son introducidas por los controles primarios, entregando nuevas consignas de potencia a cada central generadora. Este control puede ser distribuido o centralizado. Esta capa de control actúa en una escala de tiempo mas lenta que el control primario.

Por último, el control terciario esta estrechamente relacionado con la coordinación del sistema de potencia, optimizando la operación económica del sistema y sus flujos de potencia. Este control, generalmente realizado de manera centralizada, busca minimizar los costos de operación y las pérdidas, despachando todas las centrales generadoras que participan del sistema, considerando restricciones operativas. En el caso del Sistema Eléctrico Nacional de Chile, este control es realizado a intervalos de una hora por el Coordinador Eléctrico Nacional.

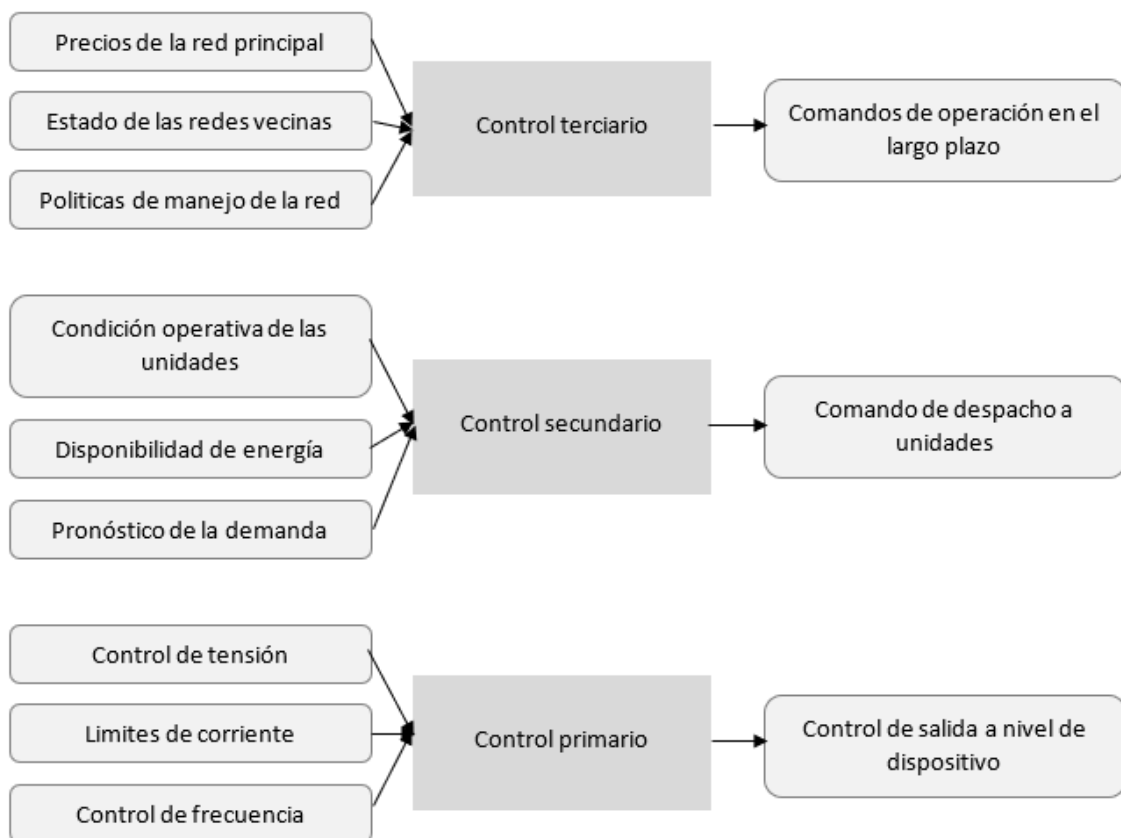


Figura 2.1: Control Jerárquico. Adaptada de [4]

2.2. Protocolos de Comunicación

Un protocolo de comunicación es un conjunto de reglas que conforman un estándar, políticas, procedimientos y formatos que definen el intercambio de paquetes de información para lograr la comunicación entre dos o más dispositivos a través de una red. Estos protocolos especifican una serie de requerimientos para crear una comunicación confiable. Para ello, se define como se forman los paquetes de información, mecanismos de reconocimiento de llegada de la información, entre otros.

En el contexto de las redes eléctricas, se crearon protocolos de comunicación determinados para la automatización de dispositivos utilizados en centrales eléctricas y crear la comunicación con un operador central que se encarga de monitorear y controlar la operación. Uno de los aspectos claves que buscan satisfacer los protocolos creados, es garantizar la suficiente interoperabilidad entre los distintos fabricantes de dispositivos de la industria y, así, permitir integrarlos dentro del mismo sistema, manteniendo una velocidad de información aceptable.

2.2.1. Modbus

Modbus es el protocolo de comunicaciones que se ha extendido por toda la industria de la automatización y el medio más común para conectar dispositivos electrónicos en una red industrial.

Este protocolo fue creado por Modicon (ahora conocido como Schneider Electric) en el año 1979 con el objetivo de establecer una comunicación con los controladores lógicos programables (PLC), permitiendo la operación remota y accediendo a las variables de medición. En esos años se convirtió rápidamente en el estándar y, actualmente, sigue siendo el protocolo más común para automatizar procesos industriales. El protocolo de comunicación Modbus es de libre acceso, por lo que puede ser implementado en todo proceso sin tener que pagar por los derechos de autor.

2.2.1.1. Query-Response Cycle

La primera versión de Modbus utiliza una arquitectura de maestro-esclavo, donde el dispositivo que solicita información se llama maestro Modbus, mientras que los dispositivos que entregan la información son los esclavos Modbus. Esta nomenclatura se utiliza debido a que el dispositivo esclavo no puede ofrecer información ni comunicarse de ninguna manera con el maestro. Así, el esclavo se mantiene en *stand-by* hasta que el maestro solicite información escribiendo en sus registros, el cual a su vez envía la información requerida al maestro, o tomando acciones de control según se instruya [6]. Ejemplo del ciclo de solicitud-respuesta del protocolo, se muestra en la Figura 2.2.

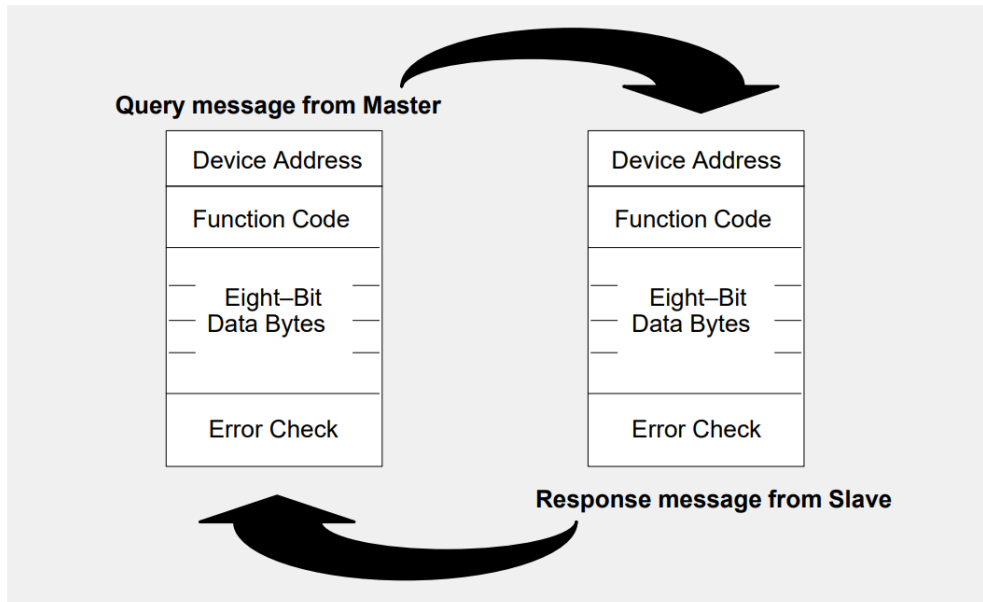


Figura 2.2: Ciclo de solicitud y respuesta del protocolo Modbus. Fuente: [6]

Cada mensaje enviado en el protocolo Modbus se divide en cuatro partes: dirección del dispositivo, código de función, la información propiamente tal como *data-bytes* y el verificador de errores. Cuando el emisor es el maestro, el código de función indica qué acción debe tomar el esclavo con la dirección de dispositivo del encabezado. Los *data-bytes* contienen información adicional que ayudará al esclavo a llevar a cabo la acción requerida, mientras que el verificador de errores le permite al esclavo validar la integridad del mensaje. Si la acción instruida por el maestro no puede ser llevada a cabo por el esclavo, este último enviará un mensaje al primero con la descripción del error en los *data-bytes*.

2.2.1.2. Modbus en Puertos Seriales

En sus inicios el protocolo se implementaba con puertos seriales que adoptaban los estándares RS232/RS485/RS422 (*Recommended Standard*) para establecer la comunicación entre dispositivos. Antiguamente se reconocían dos tipos de comunicación por sobre este medio físico: Modbus RTU (el más utilizado) y Modbus ASCII, el cual posee la misma estructura que RTU pero transmite la información en caracteres ASCII, es decir, en caracteres hexadecimales que contienen 4 bits cada uno. Actualmente, se considera obsoleto y se recomienda no llevarlo a la práctica en nuevas implementaciones.

La unidad básica de información del protocolo es llamado *Protocol Data Unit* (PDU), independientemente de la capa de comunicación que se utilice. La composición del PDU se muestra en la Figura 2.3

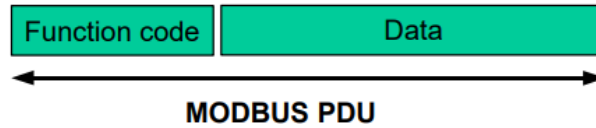


Figura 2.3: Unidad de información del protocolo. Fuente: [7]

El cliente que inicia una transacción en Modbus agrega campos adicionales al PDU para construir una unidad de información apropiada para ser transmitida mediante los puertos seriales, como se muestra en la Figura 2.4.

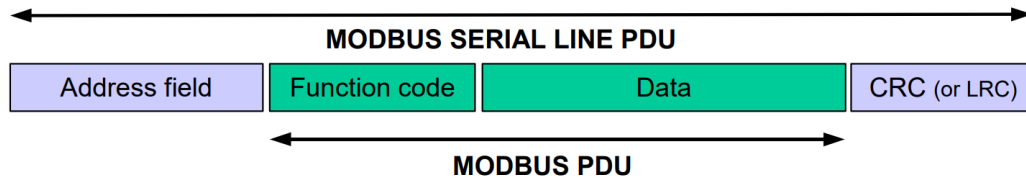


Figura 2.4: Unidad de información en comunicación serial. Fuente: [7]

La comunicación a través de este medio se lleva a cabo como se indica en la sección anterior. El maestro es el único elemento que puede iniciar solicitudes en la red, mientras que los esclavos solo responden a los requerimientos entrantes. El maestro puede emitir mensajes dirigidos a un solo esclavo (cuando se requiere realizar una acción de control) o en modo “broadcast”, de manera que se da una instrucción de acción global a toda la red (para enviar datos en tiempo real).

El verificador de errores es CRC (*Cyclical Redundacy Checking*) cuando se utiliza en modo RTU o LRC (*Longitudinal Redundacy Checking*) cuando se utiliza en modo ASCII. Ambos tienen el mismo principio de funcionamiento: el emisor calcula un valor de 16-bits (8-bits en LRC) y lo adjunta al final del PDU, mientras que el receptor a medida que llega el mensaje calcula el valor del mensaje y lo compara con el campo CRC/LRC recibido; si ambos coinciden el mensaje se recibió correctamente.

2.2.1.3. Modbus con comunicación TCP/IP

En una red TCP/IP deja de ser válida la denominación de maestro/esclavo ya que la comunicación se puede dar entre dispositivos de la misma red y, por ello, se pasa a un modelo de cliente/servidor, donde el maestro se convierte en cliente y el esclavo a servidor. Esta nueva definición tiene la principal ventaja de que puede existir más de un cliente que solicita información al sistema y aumenta el número de elementos que pueden ser monitoreados en tiempo real [8].

La unidad básica en este protocolo es el PDU, como se muestra en la Figura 2.3, y los bloques de este poseen el mismo funcionamiento que en Modbus RTU. Cuando la información va a ser transmitida en una red TCP/IP, el PDU es encapsulado como se muestra en la Figura 2.5.

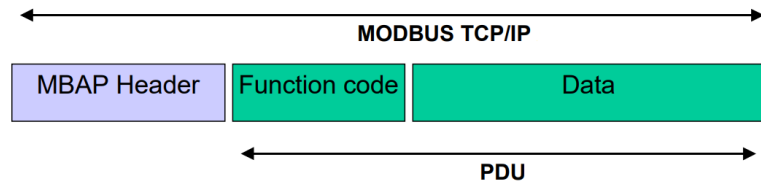


Figura 2.5: Unidad de información en comunicación TCP/IP. Fuente: [8]

El bloque adjuntado al inicio del mensaje llamado *MODBUS Application Protocol Header* (MBAP Header) provee la información necesaria para la comunicación en TCP/IP. Este nuevo bloque posee diferencias con respecto a su contraparte en RTU:

- El “Address Field” utilizado en comunicación serial es reemplazado por un “Unit Identifier”, usado con el objetivo de comunicarse a través de *routers*, *gateways* o *bridges* que tienen una única dirección IP a los múltiples dispositivos Modbus que utilizan comunicación serial.
- La mensajería está diseñada de tal manera que el receptor puede verificar que el mensaje ha finalizado.
- Los mensajes incluyen información adicional en el MBAP Header que permiten al receptor detectar los límites del mensaje, aún cuando este ha sido dividido en múltiples paquetes para su transmisión. La existencia de reglas implícitas y explícitas sobre el largo del mensaje, junto a un verificador de errores CRC-32 en Ethernet, reducen la posibilidad de que un error no sea detectado a valores infinitesimales cercanos a cero.

Una red que opera con el protocolo Modbus no está limitada a operar solo mediante TCP/IP o puertos seriales, sino que la combinación de estos es factible si se dispone de los adaptadores necesarios. La Figura 2.6 muestra un ejemplo de una red que tiene comunicación con puertos seriales y por protocolo TCP/IP.

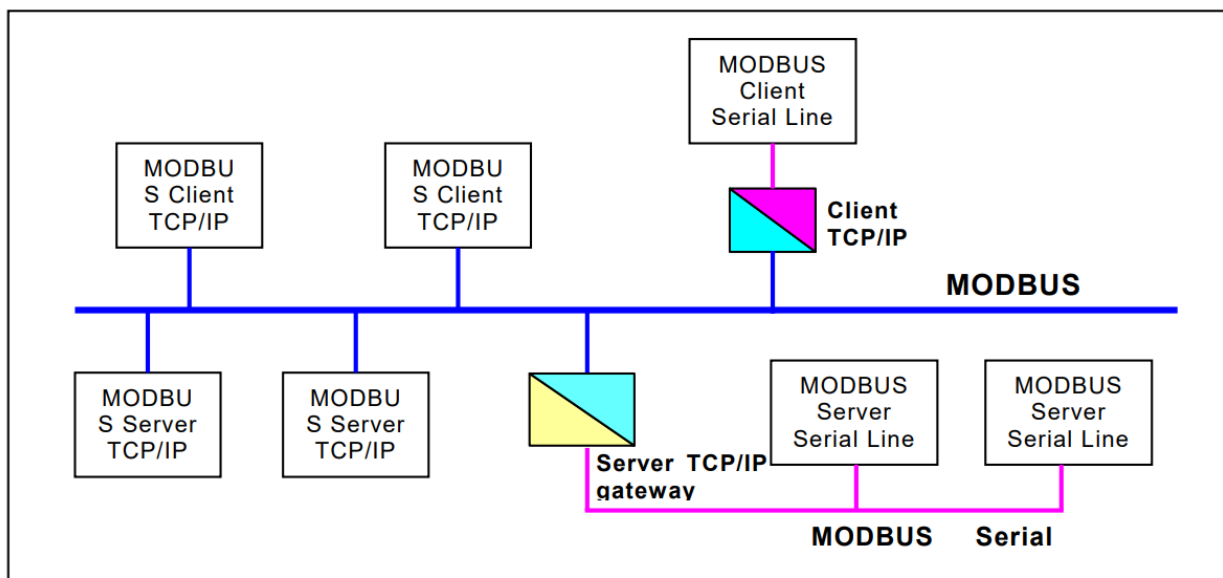


Figura 2.6: Arquitectura de comunicación de una red Modbus TCP/IP. Fuente: [8]

2.2.2. IEC 60870-5

El estándar IEC 60870 esta distribuido en distintas partes y el objetivo de estos es definir los sistemas usados para realizar el control supervisado y la adquisición de datos de sistemas de control en redes eléctricas. En particular, la parte 5 titulada “*Transmission Protocols*” define los protocolos para realizar al transmisión de información en sistemas SCADA. Su desarrollo se inició en 1988, finalizando en el año 2000 con el protocolo de acompañamiento IEC 60870-5-104.

Al mencionar al estándar IEC 60870-5 en el contexto de sistemas SCADA, uno se refiere específicamente al protocolo 101, que fue publicado en el año 1994. Esto debido a que en este documento se definen completamente los protocolos de comunicación para un sistema SCADA, entregando las herramientas de aplicación necesarias para la operación. Sin embargo, este hace referencia a muchas definiciones de las secciones anteriores del paquete.

El protocolo 101 ofrece todas las funcionalidades necesarias para crear conectividad entre dispositivos de un sistema SCADA, pero no esta diseñado para ser implementado con el protocolo TCP/IP. En este contexto el estándar IEC 60870-5-104 cobra vital importancia en entender el actual funcionamiento del protocolo, ya que define el transporte de los mensajes sobre las redes que utilizan IP.

Luego del publicado el estándar 5-104, se añadieron más partes como el 5-6 que incluye una guía para testear el cumplimiento de los estándares de acompañamiento , el 5-7 para extender la ciberseguridad de los protocolos 5-101 y 5-104. Se agregó también casos de prueba para los últimos dos protocolos. En la Figura 2.7 se muestran las partes y secciones de cómo esta compuesto el IEC 60870-5.

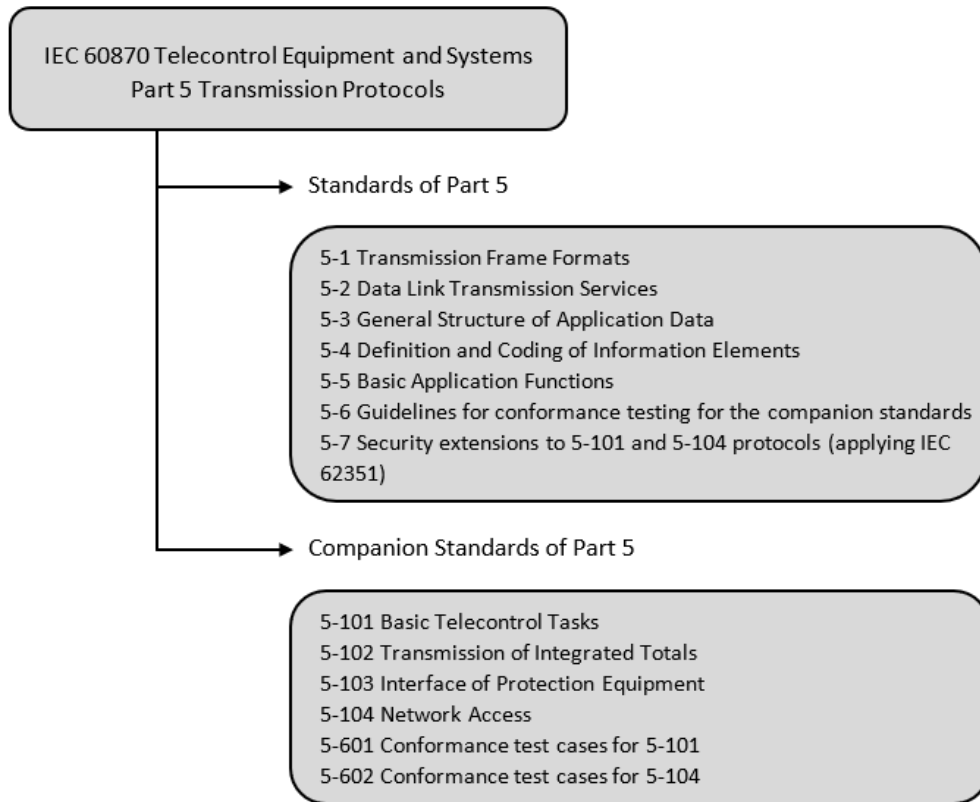


Figura 2.7: Estándares del protocolo IEC 60870-5 (imagen adaptada de [9]).

2.2.3. DNP 3

DNP3, o *Distributed Network Protocol* y también conocido por IEEE Std 1815, es un estándar creado para proveer comunicación entre dispositivos HMI/SCADA, RTU e IED, con el fin de ofrecer un medio ligero de transporte de datos sencillo manteniendo la integridad de los datos.

Cuando el estándar IEC 60870-5 aún estaba bajo desarrollo y no cumplía con todas las necesidades de las compañías eléctricas de Norte América, en específico, lograr la interoperabilidad de distintos proveedores de elementos que componen una red eléctrica de un SCADA, la empresa Harris, *Distributed Automation Products* desarrolla la primera versión de DNP3 en 1993, para luego cederle la responsabilidad de mantener y definir las futuras especificaciones al Grupo de Usuarios de DNP3, un grupo compuesto por usuarios y fabricantes. En un principio fue diseñado para compañías eléctricas, pero en la actualidad se ha expandido ampliamente hacia otras industrias de utilidades como agua, tratamiento de aguas, petróleo y gas. El protocolo era de acceso libre y toda entidad podía acceder a la documentación de cómo implementar el protocolo y las especificaciones que tiene que cumplir. En la actualidad, para acceder a los documentos es necesario pertenecer al Grupo de Usuarios de DNP3, lo cual se logra pagando una tarifa anual. Se pueden adquirir los documentos con fines académicos por un monto de 1 dolar anual [12].

Inicialmente, DNP3 utilizaba puerto seriales al igual que en el IEC 60870-5 y en Modbus,

pero hoy permite comunicaciones a través de los protocolos TCP/IP o UDP/IP posibilitando el uso de redes IP para el transporte de los mensajes.

El protocolo mantiene una arquitectura de maestro-esclavo, sin embargo, la principal característica que lo distingue de su antecesor Modbus es la sincronización de datos y el manejo de estampa de datos, además del hecho de que un esclavo puede transmitir información cuando no se le ha instruido. Dado esto, el protocolo generalmente es estándar para dispositivos como reconvertidores, relés, entre otros para permitir el accionamiento coordinado de los equipos.

En el año 2000 fue referenciado en el IEEE Std 1379-2000, el cual contiene una serie de recomendaciones prácticas en la implementación de enlaces de comunicación entre RTU y HMI/SCADA [13].

2.2.4. IEC 61850

El estándar internacional IEC 61850 define los protocolos de comunicación para intercambiar información con dispositivos inteligentes electrónicos, permitiendo la interoperabilidad entre distintos fabricantes. El enfoque técnico de su desarrollo hace que este protocolo sea flexible y resiliente frente a los nuevos desarrollos tecnológicos.

Inicialmente, el estándar IEC 61850 estaba destinado a incorporar pequeñas mejoras al protocolo ya existente, al protocolo de telecontrol IEC 60870-5. Tal hubiera sido el destino del protocolo, si no fuera por las diferentes ideas proporcionadas por distintos actores de ese tiempo y desde Estados Unidos, quienes trabajan en un nuevo protocolo para abordar las problemáticas de protocolos anteriores. El grupo de usuarios de *Utility Communication Architecture* (UCA) identificó que la integración, el testing y los problemas de mantenimiento causado por los indexadores (que era cómo se accedían a los registros de los equipos), comprendían aproximadamente el 80% de los costos de integración del ciclo de vida de las subestaciones. La *International Electrotechnical Commission* (IEC), en conjunto con *Electric Power Institute* (EPRI), y el grupo *Institute of Electrical and Electronics Engineers* (IEEE) lograron armonizar las nuevas tecnologías, lo cual sentó las bases de lo que hoy en día es el IEC 61850.

La Edición 1 de este estándar fue formalmente completado con las publicaciones IEC 61850-8-1 y IEC 61850-9-2 (*streaming* de información desde TTPP y TTCC) en el año 2004, siendo este protocolo el más reciente reconocido internacionalmente. El proceso del estándar exige que este sea revisado cada 5 años. Cuando llega ese momento, el estándar (partes de este) puede ser reafirmado, eliminado o publicado con cambios. De esta manera, 6 años después de la publicación de la Edición 1, la Edición 2 fue publicada, y se esperaba que para el año 2018 se agregara una adenda a la Edición 2 modificando este último [10]. Esta adenda, que tiene por nombre Edición 2.1, incluye la parte 6 publicada en el año 2018 y, en Febrero 2022 las partes 7-2, 7-3, 7-4, 8-1 y 9-2 (estas publicaciones remplazan a las originales).

Introductory	Requirements	Objects and Base Types	Functions	Execution	Engineering / Configuration	Testing
Standard (61850-1)	General, Electrical and Environmental (61850-3)	Objects and Models (61850-7-1)	Common Functions (61850-7-4)	Mapping to MMS and TCP/IP (61850-8-1)	Configuration (61850-6)	Abstract Test Cases (61850-10)
Glossary (61850-2)		System Engineering (61850-4)	Abstract Communication (61850-7-2)	Hydro-Electric (61850-7-410)	Web Technology (61850-8-2)	Modeling for Automation (61850-600)
	Information Exchange (61850-5)	Common Data Classes (61850-7-3)	Distributed Energy (61850-7-420)	Sampled Values (61850-9-2)	Role Based Access Control (TR 61850-90-19)	UCA Conformance Testing
	Redundancy for IEDs and Applications (TR 61850-90-20)		Wind Power (61400-25-2)	PTP Time Sync (61850-9-3)	HMI (TR 61850-600x)	
	Networking (TR 61850-90-4) and (TR 61850-90-12)		Logic Modeling (61850-90-11)	Communication between Substations (TR 61850-90-1)		
			Alarm Handling (61850-90-18)	Communication between Substations and Control Centers (TR 61850-90-2)		
				Gateway Technology: 61850 to 870-5: TR IEC 61850-80-1 61850 to Modbus: TR 61850-80-5 61850 to DNP: IEEE 1815.1		

Figura 2.8: Estándares del protocolo IEC 61850 (imagen adaptada de [10]).

La Figura 2.8 muestra diferentes categorías en las cuales se pueden dividir las publicaciones. Estos son solo una pequeña parte del total de ellos. A principios del 2020 el total de publicaciones relacionados con el estándar IEC 61850 ascendía a aproximadamente 134 documentos, dentro de los que se incluyen reportes técnicos, especificaciones técnicas y otros relacionados (todos de pago).

El conjunto de documentos que pertenecen a la categoría “Introductory” entregan información introductoria, definiciones y una visión general de la estructura del estándar. En “Requirements” se estudian y definen los requerimientos que deben tener los procesos en la implementación de un sistema IEC 61850 en sus variables ambientales (temperatura, humedad, entre otros) y los requerimientos del protocolo de comunicación como lo es la sincronización de tiempo y la duración de un intercambio de información. La siguiente categoría fija cuál será la estructura del estándar. En “Functions” se estandariza las funcionalidades necesarias para proveer la comunicación entre dispositivos cumpliendo según los requerimientos de cada aplicación. Como se puede ver en la Figura 2.8, existen documentos que se desarrollaron específicamente para cumplir con las aplicaciones individuales del sistema, tales como una planta eólica y el manejo y gestión de un sistema de alarmas. Dentro de la terminología usada en el estándar IEC 61850, estas funciones se definen como Nodo Lógico (*Logical Node*). En la categoría “Execution” se agrupan los documentos que detallan cómo implementar concretamente el sistema de comunicación para cumplir con los requerimientos anteriormente definidos. Por ejemplo, se detalla cómo el intercambio de información se lleva a cabo en

una red TCP/IP, qué protocolos y opciones se pueden utilizar para cumplir con los requerimientos de velocidad que propone el estándar y otros procedimientos. Adicionalmente, se incluyen especificaciones para configurar e implementar *gateways* cuya función es mapear la información (transformación de datos) desde IEC 61850 a DNP, Modbus e IEC 60870-5. El apartado de “Engineering/Configuration” contiene documentos que especifican metodologías concretas y tecnologías para configurar un sistema IEC 61850. Esto abarca temas de seguridad, automatización e interfaz de usuario-máquina. Por último, todo lo anterior debe ser probado en casos de estudio, por lo que en la categoría “Testing” se encuentran todos los documentos que entregan casos de estudio para verificar el desempeño del sistema.

Los elementos diferenciadores de este protocolo con sus precedentes son los siguientes:

- A diferencia de sus predecesores, que para acceder a la información del equipos es necesario conocer el número del registro al cual se accederá y asignarle un nombre, el protocolo IEC 61850 estandariza, entre otras cosas, los nombres de estos registros, creando un sistema jerárquico y estructurado. Así, si el número del registro cambia, el nombre sigue siendo el mismo y no es necesario re-configurar.
- La implementación de este protocolo se realiza mediante la programación en base a objetos. Como este tipo de programación enfatiza la data y no el procedimiento, es posible modificar partes del protocolo para que el sistema se adapte a las necesidades del usuario.
- El protocolo IEC 61850 esta diseñado para operar específicamente a través de Ethernet o con el protocolo TCP/IP, mientras que los protocolos anteriores fueron concebidos para utilizar puertos seriales. La desventaja que tenia este medio de comunicación por sobre los seriales era la velocidad. Mientras que a través de cables las señales se podían transmitir entre 8 a 20 milisegundos; la tecnología Ethernet no era capaz de competir con tales tiempos. Para superar esta desventaja se creó la tecnología “Publish and Suscribe”, la cual sentó las bases de los protocolos usados actualmente: llámese *Generic Object-Oriented Substation Event* (GOOSE) y *Generic Substation Status Event* (GSSE).
- Con el documento IEC 61850-6, se definen los archivos SCL (*Substation Configuration description Language*), los cuales estandarizan el formato de la data que contienen información de los dispositivos de una subestación, facilitando la configuración de dichos equipos. Esto aumenta aún más la interoperabilidad y las capacidades de *Plug and Play*.

Por último, el protocolo IEC 61850 posee distintos tipos de comunicación:

1. **Manufacturing Messaging Specification (MMS):** utilizado para la comunicación entre IED y el sistema SCADA con objetivos de aplicación, configuración y monitoreo del intercambio de información.
2. **Generic Object-Oriented Substations Events (GOOSE):** usado para enviar mensajes independientemente de los estados de las IED. Utilizado generalmente en aplicaciones de teleprotección por su velocidad de actuación.
3. **Sampled Measured Values (SMA):** provee una comunicación rápida y confiable de los valores de medición, protección y control de los transformadores de corriente y tensión.

2.2.5. Gateway

Anteriormente, se revisaron distintos protocolos de comunicación que son ampliamente utilizados en la industria, aunque, en general, no se utilizan todos simultáneamente en una misma planta. A pesar de que uno de estos puede presentar ventajas claras por sobre otros, no existe un consenso global en cual estándar se debe utilizar. Por ejemplo, en Estados Unidos los protocolos mas utilizados siguen siendo Modbus y DNP3.0 en sus versiones seriales y TCP/IP, mientras que en Europa migraron a IEC 60870-5 y luego a IEC 61850 una vez que fue finalizado [10].

Sin embargo, existen otras opciones para crear una conexión de comunicación entre los dispositivos en planta y un centro de control, que son los *Gateways*. Estas piezas de software funcionan como un traductor de protocolos, ya que reciben la información proveniente de un esclavo/cliente que utiliza, por ejemplo, el protocolo Modbus, y transforman la información al protocolo IEC 61850 que es utilizado en el centro de control SCADA.

Uno de los productos que se encuentran en el mercado es *SCADA Data Gateway*, desarrollado por *Triangle MicroWorks, Inc* [11]. Este traductor tiene soporte para los protocolos Modbus, DNP3, IEC 60870-5, IEC 61850, OPC, entre otros.

La ventaja de un software como este, es externalizar la conexión de los dispositivos de planta y, por lo tanto, la adquisición de datos y el control de estos, logrando interoperabilidad, flexibilidad y facilidad en la implementación de un sistema SCADA. En la Figura 2.9 se observa el nivel en el cual opera un *Gateway*, y en este caso específicamente, los lenguajes a los cuales entrega soporte el programa *SCADA Data Gateway*.

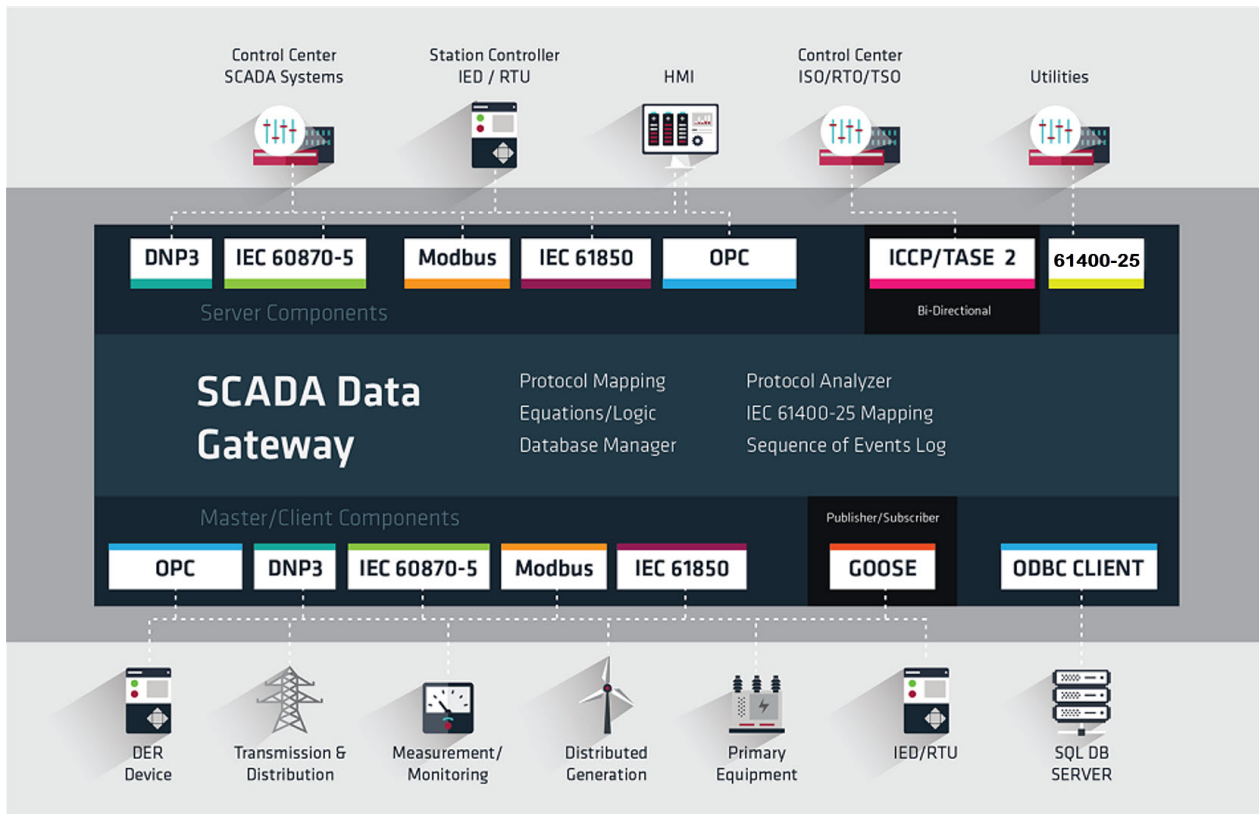


Figura 2.9: Imagen referencial de SCADA Data Gateway. Fuente: [11]

2.3. Sistemas SCADA

Un sistema SCADA (por sus siglas en inglés, *Supervisory Control and Data Acquisition*) es una herramienta de automatización y control utilizada en varios procesos industriales y productivos que puede controlar, supervisar, recopilar datos, analizar datos y generar informes a distancia.

En la actualidad los sistemas SCADA son elementos fundamentales en los procesos industriales, ya que ayudan a mantener los procesos operativos, manteniendo la eficiencia, comunicando rápidamente información crítica del sistema controlado, procesan los datos históricos de la instalación y permite conocer a completitud el funcionamiento de un proceso.

Adicionalmente, el software SCADA puede también poseer funciones como: grabación secuencial de eventos o acontecimientos en una base de datos que se producen en el proceso; gestión de alarmas frente a mediciones fuera de rango.

2.3.1. SIMATIC WinCC V7

Siemens es una empresa reconocida con una alta trayectoria en la venta de productos eléctricos. Dentro de sus productos ofrece soluciones integrales para diversas industrias, controladores lógicos programables o PLC, por sus siglas en inglés, motores, drivers, sistemas de automatización, entre otros.

El programa base de *Siemens* que tiene aplicaciones de telemedida y adquisición de datos, lleva por nombre SIMATIC WinCC V7 [15]. El software está desarrollado de manera modular, lo que quiere decir que el programa base contiene las funcionalidades básicas para controlar y monitorear los procesos de la planta en cuestión, y puede ser mejorado o agregar nuevas capacidades a través de módulos que se adquieren de manera independiente. Las funciones que se incluyen en el programa base incluye:

- Visualizar el proceso y configurar la interfaz gráfica.
- Operar y monitorear el proceso a través de la interfaz.

El programa no está especialmente dedicado a la operación de sistemas eléctricos sino que se encuentra enfocado para ser implementado en todo tipo de industria, incluyendo la eléctrica.

Para aumentar las capacidades del programa, se pueden adquirir módulos que mejoran la escalabilidad, la disponibilidad, el análisis de datos, entre otros. Dentro de todos los módulos disponibles se mencionan:

- **Server:** permite la interoperabilidad del sistema SCADA a varios usuarios, facultando a distintos usuarios utilizar el sistema, simultáneamente.
- **WebUX:** permite visualizar y operar el sistema en dispositivos móviles.
- **TeleControl:** expande el sistema para operar con otros protocolos de comunicación (SIANUT ST7, DNP3, IEC 60870-5-101/-104).
- **WebNavigator:** permite visualizar y operar el sistema a través de la web.
- **Process Historian:** habilita el almacenamiento de la información que se recopila en el SCADA al historiador (base de datos).

El último producto lleva por nombre **Microgrid Control - a SICAM application**. Esta aplicación está desarrollada para ser implementada junto a la serie de productos SICAM A8000, las cuales, son una unidad de control remota pero con la capacidad de automatizar la operación. De esta manera, este producto funciona como un nexo para otros dispositivos controlables. *Microgrid Control* por su parte posee las funciones de: detección de *blackout*, partida en negro, modos de operación (isla y conectado a la red) automático, partida automática de generadores de respaldo, optimización de la operación, monitoreo de la reserva disponible, recorte de punta, gestión del estado de carga de baterías, entre otros [16].

En las Figuras 2.10 y 2.11 se muestran imágenes referenciales de un proceso siendo supervisado mediante el software WinCC y del monitoreo de una micro-red en *Microgrid Control*.

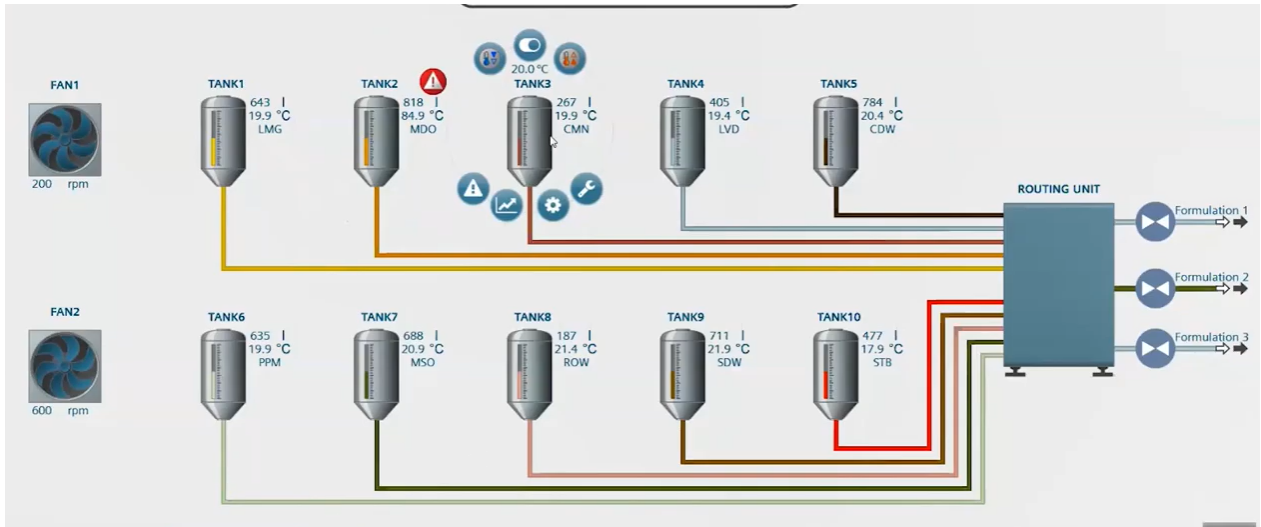


Figura 2.10: Supervisión y control de proceso en WinCC de Siemens

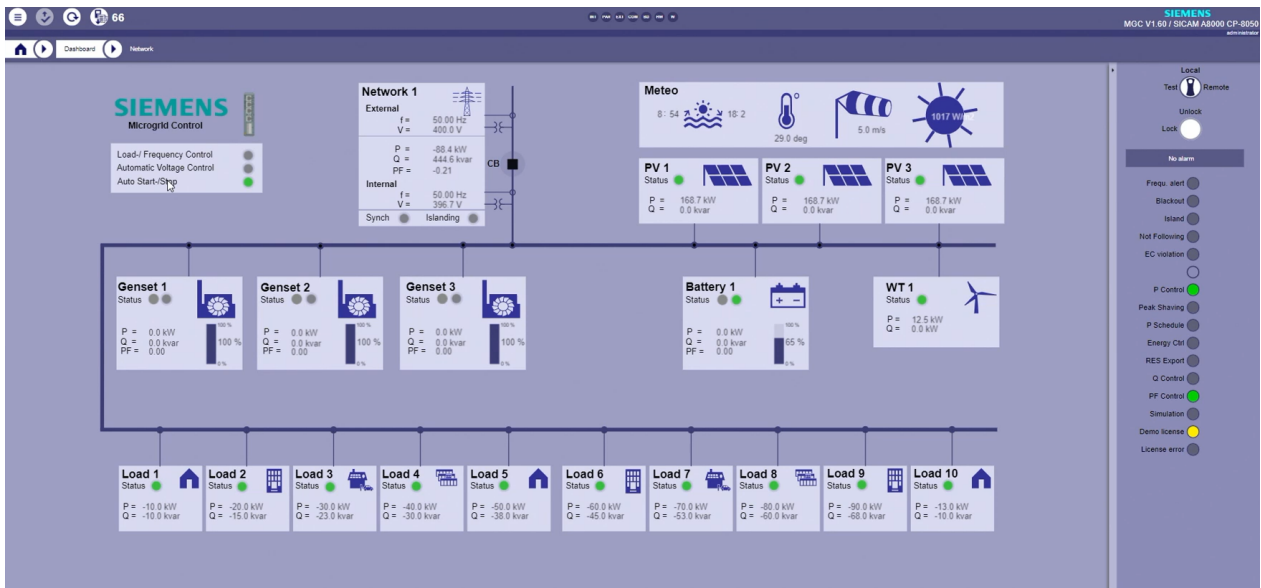


Figura 2.11: Supervisión y control de una micro-red en Microgrid Control de Siemens

2.3.2. eSCADA

Electrical Scada (eSCADA) es el sistema de monitoreo y adquisición de datos de ETAP, empresa internacional que se especializa en ofrecer soluciones de ingeniería con respecto al monitoreo, operación, diseño, simulación, optimización y automatización de sistemas eléctricos [17].

Dentro de las funcionalidades y servicios que ofrece el software de esta empresa, se encuentra la comunicación confiable a IED, RTU y al SCADA Server, cumpliendo con estándares y protocolos internacionales de comunicación, de los cuales, vale mencionar: DNP3, Modbus, OPC, IEC 61850, entre otros. Por lo tanto, cumple con proveer conectividad con equipos que

utilicen variados sistemas de comunicación.

Otras funciones que ofrece son: monitoreo en tiempo real de la red con una interfaz intuitiva alojada en una pagina web, lo que permite la visualización del SCADA a todo dispositivo que tenga acceso a internet; sistema de gestión de alarmas y notificaciones, simulaciones predictivas del sistema en respuesta a acciones realizadas por el operador y eventos utilizando información histórica y real del sistema; sistema 'playback' del sistema, lo que permite identificar causas y efectos de eventos pasados e investigar distintos cursos de acción.

Una de las funciones relevantes de este programa es el control supervisado y la optimización de la operación. Este incluye herramientas de validación al operar acciones de conmutación, inhibidores de elementos de la red, debido a, por ejemplo, mantenencias en alguna línea de transmisión o equipos de maniobras y un sistema variable de optimizador económico variable, ya que permite al operador escoger sobre cuales factores optimizar la operación y asignarle pesos a cada una de ellas. Entre las variables a disposición, es posible minimizar las pérdidas activas y/o reactivas de la red, minimizar el uso de combustible, optimizar según bandas de tensión admisibles y flujos por las líneas permitidos, mantener una tensión plana, entre otros. También incluye un simulador de los elementos de control del sistema (excitadores, governors, etc), lo que ayuda a mejorar al eficiencia y robustez del sistema. En la Figura 2.12 se observa la operación de una micro-red siendo monitoreada con eSCADA.

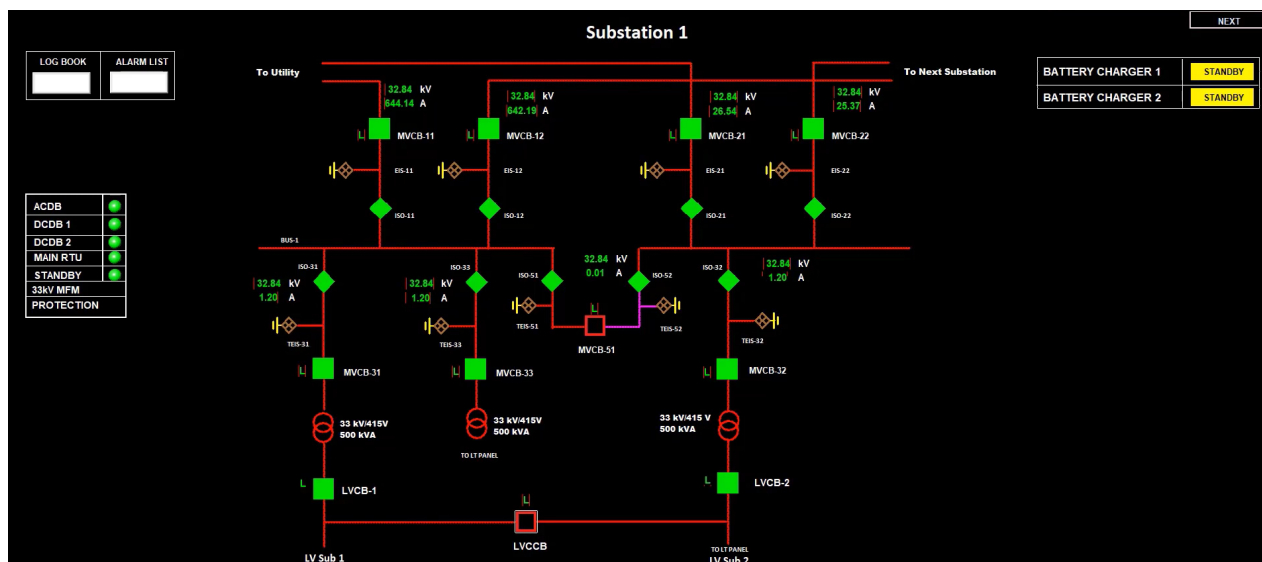


Figura 2.12: Supervisión y control de una micro-red con Electrical Scada de ETAP

En resumen, ETAP propone una solución robusta para el monitoreo y control del sistema eléctrico, es versátil, y cumple con un gran espectro de protocolos de comunicación, lo cual lo hace compatible con una gran variedad de equipos de medición; ofrece distintas herramientas que permiten mejorar la eficiencia del sistema; permite el estudio y análisis con poderosas simulaciones RMS y EMT; y la sintonización de los parámetros de control de las centrales generadoras. Sin embargo, de todas las soluciones que ofrece, no todas serán aprovechadas a su máximo potencial, debido a que el operador de la micro-red no será necesariamente un ingeniero eléctrico o con conocimientos avanzados en sistemas eléctricos. Por otro lado, no

posee la flexibilidad de control, que una micro-red requiere, y se tendría que adquirir otro producto para satisfacer esta necesidad (como, por ejemplo, el EMS que tiene disponible ETAP).

2.3.3. EcoStruxure™ Power Operation

Schneider Electric (SE) ofrece su producto comercial *EcoStruxure™ Power Operation* (anteriormente era llamado EcoStruxure Power SCADA Operation) para la operación de sistemas eléctricos. EcoStruxure de por sí, es el nombre que lleva las soluciones de SE orientadas al internet de las cosas. Dicho esto, sus productos son independientes de los otros (en la mayoría de los casos) en donde cada uno de ellos tiene distintos clientes objetivos, dependiendo de la industria a la cual esta diseñado el programa.

En el contexto de operación y control de sistemas eléctricos, se puede encontrar una amplia variedad de programas con distintos tipos de uso. El programa más relevante para este trabajo es el EcoStruxure™ Power Operation, el SCADA de SE [18]. Este producto incluye las siguientes características: modelado del sistema eléctrico en media y baja tensión con vinculo a las RTU (certificado por un tercero bajo el estandar IEC 61850), una interfaz hombre-máquina moderna la cual permite visualizar el estado de la red desde cualquier dispositivo con HTML5. Permite realizar análisis de potencia de eventos que ocurren en la red (siempre y cuando la rapidez y calidad de la adquisición de datos lo permita), gracias a los algoritmos de *machine learning* que posee. Adicionalmente, el programa cumple con el estandar de ciberseguridad IEC 62443 4-2 al nivel S2. Una de las desventajas de este módulo es no esta desarrollado para una micro-red, por lo tanto, no posee sistemas de optimización propios que una micro-red si requiere. Además, no tiene un acceso nativo a toda la información histórica, ya que esta se almacena como una tendencia dentro del mismo sistema. Para utilizar una base de datos (en este caso SQL) y para poder almacenar y acceder a los datos brutos, se debe adquirir el modulo “Advanced Reporting and Dashboards”, por un precio de 16.542 USD [19]. La operación de una micro-red en EcoStruxure se muestra en la Figura 2.13 junto con un análisis fasorial en el momento que ocurre una falla en la Figura 2.14.

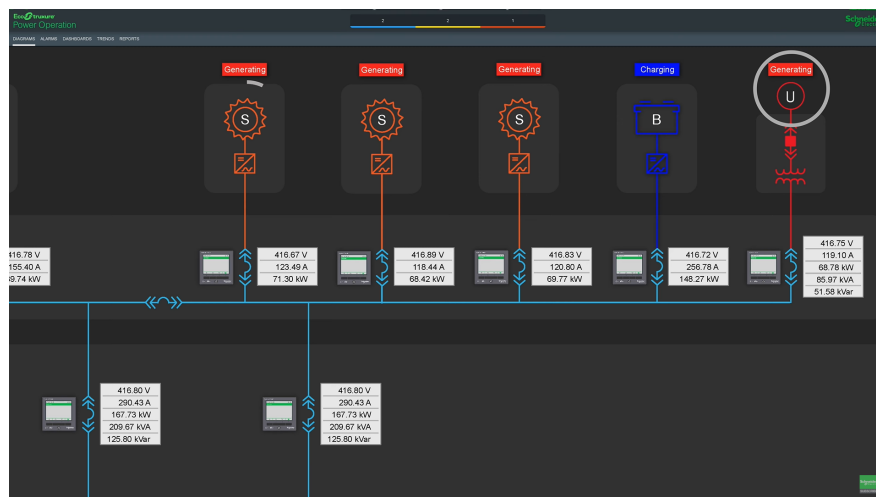


Figura 2.13: Supervisión y control de una micro-red con EcoStruxure Power Operation de SE

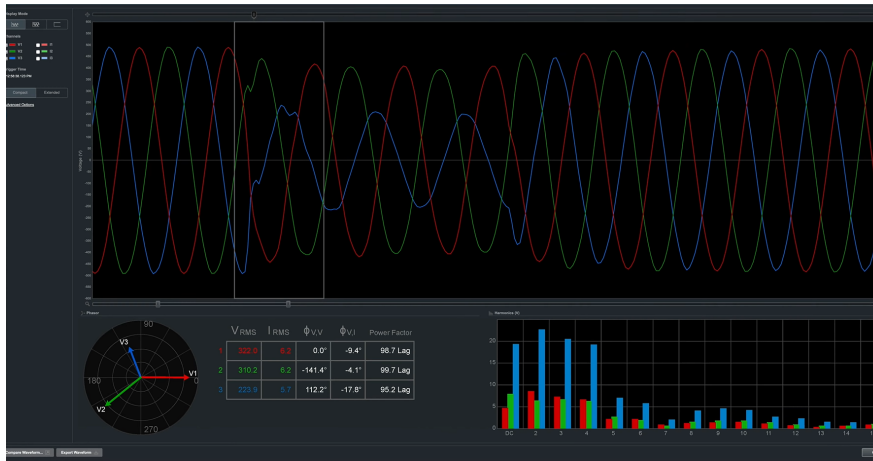


Figura 2.14: Análisis fasorial de una falla en EcoStruxure Power Operation de SE

Por otro lado, SE ofrece otros programas que son parte de la familia EcoStruxure, de los cuales se mencionan:

- **Microgrid Advisor (EMA):** esta aplicación se asemeja principalmente a un *Energy Management System* (EMS), el cual permite maximizar las utilidades de una micro-red dependiendo de las condiciones actuales y futuras mediante un sistema de pronósticos. Por ejemplo, puede determinar el momento óptimo para cargar, descargar o mantener en stand-by un banco de baterías, maximizando el resultado económico de su operación. La tarifa de inyección a la red, la demanda en hora punta y picos de demanda pueden influenciar en la toma de decisiones que determina el EMA. Este programa también puede ser accedido mediante un dispositivo que posea HTML5, donde se pueden consultar datos históricos de las DER y otros datos relevantes para el operador.
- **Microgrid Operation (EMO):** esta aplicación provee la capacidad a la micro-red de operar de manera aislada de la red. Para ello, tiene la facultad de accionar *breakers* de la red de suministro local, encender/apagar generadores síncronos locales y modificar la producción de energía solar. De esta manera, puede hacer la transición de operación *on-grid* a isla y viceversa automáticamente.

Las dos últimas aplicaciones, en conjunto con el centro de control de energía (ECC, por sus siglas en inglés) del mismo fabricante, proporciona una solución robusta para el control y monitoreo de las DER, permitiendo la operación en isla y conectada a la red [20]. Sin embargo, esta solución puede ser levemente insuficiente cuando la micro-red en la que se quiere implementar no es una instalación compacta, es decir, posee una red de distribución y abarca tramos más largos que, por ejemplo, un edificio o universidad, debido a que no es posible evaluar el estado de la red de una manera visual y rápida.

2.3.4. VTScada

El SCADA desarrollado por Trihedral es una solución utilizada ampliamente por muchas industrias en Estados Unidos y en el mundo [21]. Su popularidad se debe a dos razones: su librería de *drivers* que permite crear un canal de comunicación con más de 30 protocolos distintos de diversas industrias, desde plantas de purificación de agua, hasta complejos sistemas

de producción de alimentos; permite integrar de manera fácil y sencilla cualquier proceso a monitorear en el HMI, con una interfaz totalmente personalizable e intuitiva. Igualmente, el programa incluye el historiador, visualizadores de tendencia, generación de reportes, notificación de alarmas y un sistema de respaldo. Sin embargo, para la implementación de una micro-red se pueden observar varias desventajas. En primer lugar, no tiene una integración nativa de un EMS si es que se quisiera implementar en la micro-red, por lo que el control de esta tendría que llevarse a cabo manualmente. Adicionalmente, como no es un software dedicado a la industria eléctrica, presenta elementos gráficos genéricos que sirven para representar todo tipo de proceso, pero no eléctrico. Dicho esto, se pueden lograr resultados similares a los que presenta la solución para micro-redes de *Schneider Electric*, permitiendo visualizar corrientes, tensiones y potencias de distintos consumos o de las DER, y en general, toda variable que pueda ser medida por sensores de campo (velocidad del viento, irradiación solar, etc).

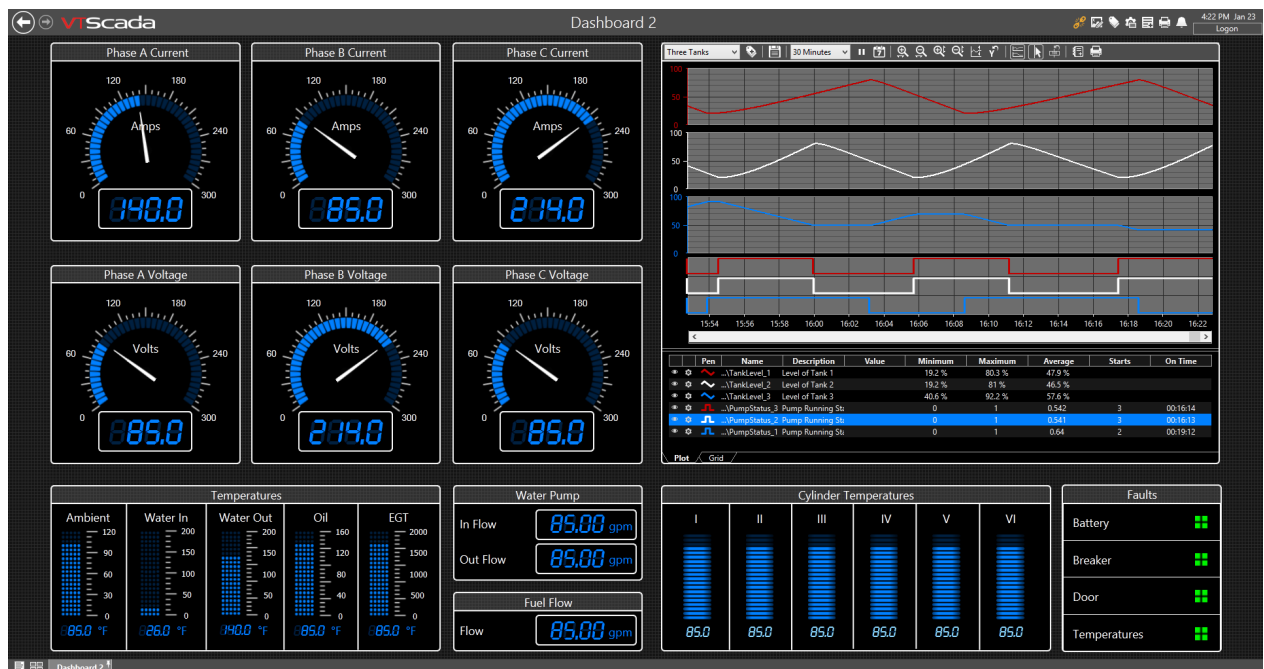


Figura 2.15: Ejemplo de HMI en VTScada para una central generadora.
Fuente: VTScada

VTScada tiene a disposición una versión gratuita para todo público, la cual tiene limitaciones contra la versión de pago. Las notificaciones de alarma vía SMS, e-mail y a través de llamadas de celular, no se encuentran disponibles. Otra limitación es que solo permite el uso de hasta 50 para ser monitoreados. Esta última de las restricciones es la más importante, ya que en solo una unidad de generación existen más de 10 variables (y el programa necesita otros *tags* adicionales para configurar la conexión).

El precio de la licencia de VTScada puede rondar entre 5.490 USD hasta los 71.385 USD, con el potencial de ser aún mayor, dependiendo de los productos adquiridos y la cantidad de *tags* solicitados [22]. La licencia mas básica incluye: acceso a la utilización de 1.000 *tags*, el servidor primario y un solo cliente de acceso concurrente.

2.3.5. Comparación Técnica

En lo que sigue, se discute la comparación técnica de los distintos productos SCADA revisados con anterioridad. La escala utilizada para evaluar las características de los sistemas sera de 1 a 5, donde una nota 1 indica que lo posee pero no satisface todas las necesidades de una micro-red, y un 5 indica que lo posee y que sí satisface todas las necesidades de una micro-red. Se evaluará como ‘vacío’ cuando no posea la característica. Los aspectos en los que serán evaluado cada uno de ellos son: la modularidad del programa, si todo esta incluido en un solo producto o este se desarrolla en torno a un programa base al cual se añaden más funcionalidades; HMI, si es que posee, qué tan bien se adecua para la operación de una micro-red eléctrica; *Multiserver* para determinar la capacidad de acceso concurrente a la plataforma de monitoreo; la interoperabilidad según cuántos protocolos de comunicación soporta para conectarse a dispositivos; Web en base a la capacidad para visualizar el sistema SCADA mediante dispositivos que tengan acceso a internet y a HTML; Historizador para almacenar los datos recibidos en una base de datos; Alarmas para la configuración de notificaciones automáticas cuando una variable se encuentra fuera del rango permitido; EMS para evaluar el sistema de gestión de energía del producto, lo cual incluye optimización de los recursos disponibles, gestión de baterías, recorte de hora punta, desprendimiento de carga para regulación de frecuencia, entre otros; y, por último, la capacidad nativa de operar en modo conectado a la red y en modo insular. Los resultados de la evaluación han sido tabulados en la Tabla 2.1.

Tabla 2.1: Evaluación de los productos SCADA

	Schneider	Siemens	Siemens	ETAP	Schneider	Trihedral
	MG Solution	MG Control	WinCC	eSCADA	EcoStruxure	VTScada
Modular	5	2	5		1	
HMI	2	2	3	4	5	3
Multiserver			3	3	3	3
Interoperabilidad	2	5	4	5	5	5
Web	3	3	3	3	3	
Historizador	3	3	3	3	3	3
Alarmas	3	3	3	3	3	3
EMS	4	5		2		
Op. Isla/Red	3	4				

El aspecto modular confiere, a quien diseña y controla la micro-red la opción de adquirir lo esencial para cumplir los requerimientos propios. Si bien es positivo que el programa sea modular y que sea flexible en cuanto a su configuración, que sea tan granular también significa que su implementación, sin pérdida de generalidad, consume más tiempo de lo necesario. En cuanto a la interfaz HMI, el programa EcoStruxure posee las funciones necesarias para una micro-red, elementos nativos para redes eléctricas y multinodal. Por otra parte, las soluciones para micro-redes son uninodales y descuidan la red de distribución. En el apartado Multiserver, existen solo dos opciones: o tiene o no tiene, pero no es comparable el desempeño de uno con el otro. En este caso, solo las solución de micro-redes no poseen acceso concurrente, según sus especificaciones. Sin embargo, como se accede a estas mediante página web, sí permite el acceso de dos o más usuarios en el mismo instante.

La interoperabilidad es uno de los aspectos más importantes en este análisis. La capacidad del sistema de adquisición de datos para comunicarse con todos los dispositivos de la micro-red, mediante distintos protocolos de comunicación, reduce los costos asociados a la configuración de la micro-red y futuras modificaciones, al mismo tiempo que habilita la adquisición de equipos a mayores fabricantes. EcoStruxure y VTScada son equiparables en el ámbito eléctrico porque tienen soporte para los protocolos presentados en este marco teórico. Fuera del ámbito eléctrico, VTScada provee la mayor conectividad en distintas industrias, incluso los protocolos específicos utilizados por los fabricantes de RTU y PLC como *Siemens* y *Schneider Electric*. WinCC se comunica nativamente con OPC (*OLE for Process Control*), para poder acceder a la interoperabilidad del programa, se vuelve necesario debe adquirir la expansión Telecontrol, la cual ofrece soporte a la mayoría de los protocolos, excepto con IEC 61850. Por otra parte, entre las soluciones para micro-redes, solo Siemens ofrece real interoperabilidad, mientras que SE, como indican sus especificaciones, utiliza OpenADR2.0 para intercambiar información; incluyendo también Modbus RTU, TCP/IP, BACnet MSTP, IP/LonWorks y HTTP/JSON.

Solo VTScada no posee el monitoreo y control del proceso alojado en la red, mientras que el resto de productos lo soporta. Con respecto al Historizador y las Alarmas, es una funcionalidad estándar en los productos encontrados en el mercado; no hay ninguno que sobresalga con respecto a sus competidores. En general, todos cumplen con entregar información histórica, análisis de tendencias, generación de reportes y gestión de alarmas.

El sistema de gestión de energía es principalmente una cualidad única de las soluciones para micro-redes. Sin embargo, el sistema eSCADA posee un sistema de optimización de los recursos de la red que se asemeja a un EMS, pero que deja muchas funciones sin cumplir que requiere una micro-red, por ejemplo, control de desprendimiento de carga. Los EMS de Schneider y Siemens están preparados para todo tipo de evento que pueda ocurrir en la micro-red, así como el control absoluto de las cargas del sistema, DER, generadores de respaldo, entre otros. Dentro de las funciones destacadas que cumplen estos sistemas son: detección de apagones, partidas en negro, reconexión automática a la red, políticas de transición entre modo insular y conectado a la red, partida automática de generadores de respaldo, balance entre carga generación, minimización de costos de combustibles, gestión de reservas, gestión de baterías, recorte de punta, pronóstico de demanda y generación renovable. Dentro de ambas soluciones para micro-redes se destaca Siemens principalmente por el mayor detalle de las funciones que posee.

En conclusión, los sistemas SCADA que no están diseñados para micro-redes tienden a no poseer la flexibilidad operativa que requieren las micro-redes, siendo necesario complementar estos sistemas. En cambio, las soluciones para micro-redes son completas y cubren las necesidades de estas de buena manera, aunque en términos de visualización no representan la red eléctrica como tal, no tiene opciones para monitorear estado de reconectores, flujos de potencia ni análisis de fallas, demostrando ser insuficiente si ocurren problemas dentro de la micro-red.

Capítulo 3

Arquitectura y Caso de Estudio

En este Capítulo se aborda la arquitectura del sistema SCADA desarrollado. Este se realiza teniendo en consideración los elementos que se encuentran disponibles en el laboratorio de energía del Departamento de Energía Eléctrica de la Universidad de Chile.

3.1. Arquitectura del Sistema

En base a lo expuesto en el Capítulo 2, se propone desarrollar un sistema SCADA que logre cumplir con los requerimientos de una micro-red. Los aspectos clave que debe cumplir el sistema es mantener un control supervisado y habilitar el accionamiento remoto de los recursos de la red en tiempo real.

El primer paso de diseño del sistema es identificar los requerimientos de comunicación que tendrá la micro-red, en base a los elementos que la componen, debido a que los dispositivos no son compatibles con todos los protocolos.

El segundo paso es definir y desarrollar los protocolos que serán implementados para lograr el intercambio de información. Estos podrán ser un subconjunto de todos los protocolos disponibles según las capacidades de los equipos.

El tercer paso, es desarrollar el HMI (*Human-Machine Interface*) que dará a lugar al SCADA. En este paso es necesario definir las capacidades que se requieren pensando como usuario. En este caso, se concibe que el sistema SCADA posea capacidades de *drag and drop*, es decir, que la posición y re-configuración espacial de la red sea simple. Es recomendable que las interfaces gráficas y en interfaces de usuario, el diseño siga el principio de la mínima sorpresa. Esto quiere decir, que la interfaz debiera ser programada de tal manera que funcione como la mayoría de la población espera que se comporte. A modo de ejemplo, la tecla <Escape> debiera ser utilizada para cerrar una ventana, y no para copiar un texto. El programa SCADA se desarrolló teniendo en cuenta estas reglas y requerimientos.

El trabajo esta enfocado en el desarrollo del sistema de visualización y de control, de manera que su uso requiera el mínimo conocimiento especializado de los protocolos de comunicación asociados para la configuración de estos. Esto tiene consecuencias importantes para su integración en micro-redes rurales.

3.1.1. Aplicaciones del Sistema

Las aplicaciones que se consideran dentro del desarrollo del sistema SCADA, incluyendo HMI y la comunicación, son:

- Historizador: conexión a una base de datos para almacenar continuamente los datos obtenidos desde los dispositivos de medición.
- Alarmas: un sistema de gestión que permita crear y eliminar alarmas con distintos tipos de prioridad, definido por el usuario.

Un sistema EMS se escapa de los límites temporales en el cual se desarrolla este trabajo, al igual que implementar toda la política de operación insular y conectada a la red.

3.1.2. Caso de Estudio

El sistema SCADA se implementó en el Laboratorio de Energía ubicado en el cuarto piso del Departamento de Ingeniería Civil Eléctrica de la Universidad de Chile. Los elementos que se pueden encontrar dentro del laboratorio son: equipos remotos de medición en distintos puntos de la red, acceso al flujo de potencia por las líneas, estado de los reconectores, módulos fotovoltaicos, baterías y un generador diésel de respaldo.

3.1.2.1. Topología de la Micro-red

La topología de la micro-red implementada se muestra en la Figura 3.1. El sistema está compuesto por 6 barras enumeradas desde aguas arriba (punto de conexión a la red principal) hacia aguas abajo (puntos de consumo y generación). En la barra 2 se encuentra un grupo electrógeno que físicamente se encuentra en el subterráneo de la facultad y, en la barra 6, se encuentra instalado una planta fotovoltaica.

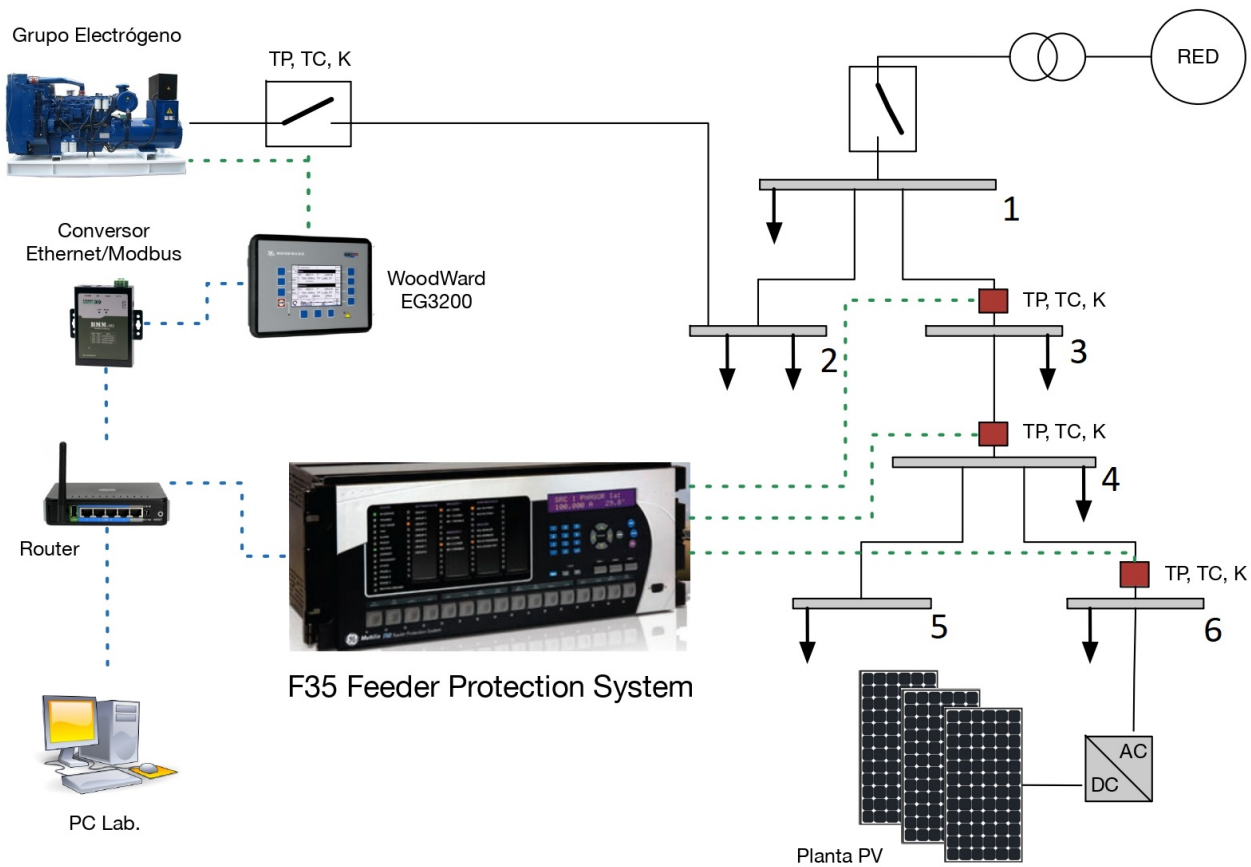


Figura 3.1: Topología de la micro-red en Universidad de Chile

3.1.2.2. Elementos de Red

Se identifican los siguientes elementos en la micro-red:

- **Switch Ethernet:** dispositivo que se encarga de expandir la red *Ethernet* a través de la interconexión de enlaces. Este dispositivo cuenta con hasta 16 puertos Ethernet.
- **Conversor Ethernet/Modbus:** este dispositivo permite hacer la conversión entre protocolo Modbus TCP y Modbus RTU. Esto tiene como principal consecuencia que se pueden conectar distintos dispositivos de medida al dispositivo a través de un bus serial y, posteriormente, convertir los datos y transmitirlos a través de la red *Ethernet*.

3.1.2.3. Canales de Comunicación

Los canales de comunicación en la micro-red son los siguientes:

- **Canal Serial:** este canal está compuesto principalmente por un bus RS-485, conectado serialmente a través de los dispositivos de medición de la micro-red.
- **Canal Ethernet:** el enlace Ethernet se encuentra implementado en todo el laboratorio y es el principal canal de comunicación entre los dispositivos de campo y el computador del laboratorio.

3.1.2.4. Dispositivos de Campo

- **Controlador grupo electrógeno:** corresponde a un controlador marca Woodward y modelo EasyGen 3200, el cual permite operar el grupo diésel en sincronismo con la red principal o en isla. Posee cuatro interfaces de comunicación, de las cuales dos corresponden a comunicación serial con RS-232 y una con RS-485, mientras que las dos restantes son del tipo CAN Bus. En la configuración de la micro-red, se conecta mediante interfaz RS-485 utilizando el protocolo Modbus RTU con el conversor Ethernet/Modbus.
- **F35 Feeder Protection System:** relé de protección que es capaz de proteger hasta 5 barras, simultáneamente. Se conecta a la red Ethernet para compartir información con el computador del laboratorio. No es utilizado en esta memoria.
- **Controlador Inversor PV:** realiza la conversión desde tensión continua a sinusoidal, mediante electrónica de potencia. Actualmente, se encuentra inoperativo.
- **PowerLogic PM750/810:** dispositivos de campo que realizan mediciones en la red eléctrica. En ellos se implementa el protocolo Modbus para el intercambio de información al sistema SCADA.

Capítulo 4

Desarrollo y resultados

En el presente Capítulo se aborda el desarrollo del programa SCADA, mencionando brevemente las librerías utilizadas en el código que se encuentra en el Anexo A.

Posteriormente, se detallan las secciones principales de código que le dan la estructura al programa. El lenguaje de programación utilizado es *Python*, en donde se alienta fuertemente a la programación orientada a objetos. De esta manera, gran parte del código desarrollado se encuentra dentro de clases y objetos.

En la parte final de esta sección, se presentan los resultados de los experimentos realizados para validar el correcto funcionamiento del SCADA.

4.1. Librerías Utilizadas

El código del programa desarrollado hace uso de variadas librerías para su funcionamiento, desde las más comunes como lo son: *numpy*, *matplotlib*, *pandas*, hasta otras menos conocidas como *wuid*, *win32*, *logging*, entre otros. Sin embargo, las librerías más importantes para el desarrollo del SCADA son:

- **pymodbus:** librería que permite transmitir información implementando el protocolo Modbus, en su modo ASCII, RTU, TCP/IP y UDP/IP.
- **tkinter:** librería nativa de python que implementa *widgets* tales como botones, *frames*, *labels*, ventanas, etc.

4.2. Interfaz Hombre-Máquina

En la siguiente sección se describen las clases, objetos y métodos creados para el funcionamiento de la interfaz hombre-máquina, que permite la creación y personalización de la micro-red a monitorear y controlar. El diseño está compuesto por el *back-end*, concepto que comprende la gestión interna de la base de datos del sistema eléctrico y cómo obtiene mediciones en tiempo real; y por el *front-end*, concepto que se relaciona con la interfaz misma y el diseño que observa el usuario. A pesar de esta definición, *back-end* y *front-end* se encuentran estrechamente ligados. A través de la programación orientada a objetos, se separan las funcionalidades de cada elemento entre su parte gráfica y la de datos. Por un lado, maneja

cómo el objeto se muestra en la pantalla y cómo se comunica con otros objetos relacionados, y por el otro, almacena el tipo de elemento, nombre, código identificador, etc.

4.2.1. Objetos Gráficos

Con el objetivo de graficar los distintos elementos que componen una micro-red en la HMI, se desarrollan las siguientes clases que funcionan como bloques individuales y que se asocian a un solo elemento de la red para representarlo y dibujarlo.

4.2.1.1. FrameObject

La clase *FrameObject*, mostrada en la Figura 4.1, es utilizada para graficar elementos estáticos de la red, vale decir, transformadores, paneles solares, turbina eólicas, generadores de respaldo, etc. Está compuesto por un objeto llamado *Frame* que encapsula los distintos elementos utilizados para mostrar información del elemento como, el flujo/inyección de potencia activa/reactiva, el nombre, la tensión, entre otros.

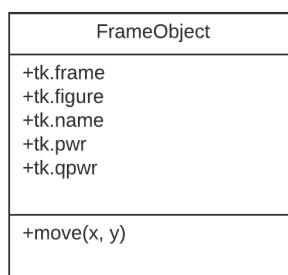


Figura 4.1: Diagrama UML - Clase *FrameObject*

Esta clase posee los métodos necesarios para que el usuario pueda personalizar libremente la posición de este y manteniendo la coherencia con los otros elementos de la red, a través de métodos para consultar posiciones cada vez que hay un cambio en el mismo. De esta manera, cuando un elemento cualquiera es modificado, todos los que guardan relación con él, también son modificados. Los métodos de esta clase se muestran en la Tabla 4.1.

Tabla 4.1: Métodos de la clase *FrameObject*

Métodos	Descripción
.rotate	Rota el elemento en -90° e invoca <code>.draw_lines()</code> .
.draw_lines()	Re-dibuja las líneas conectadas, consultando las posiciones relativas de los terminales de las líneas.
.getX(<i>terminal</i> , <i>line</i>)	Obtiene la posición en el eje x del punto de conexión del elemento, según otro terminal y la línea que los conecta.
.getY(<i>terminal</i> , <i>line</i>)	Obtiene la posición en el eje y del punto de conexión del elemento, según otro terminal y la línea que los conecta.
.clear()	Elimina todos los elementos dentro del <i>Frame</i> .
.update_labels()	Actualiza las variables eléctricas que se muestran en pantalla principal.

4.2.1.2. LineObject

La clase *LineObject*, que se muestra en la Figura 4.2, es utilizada para graficar elementos de línea de la red, los cuales pueden ser de simple conexión como, por ejemplo, conectar un elemento de la red a su respectiva barra, o para representar una línea de transmisión con flujos de potencia. Se compone de un objeto *canvas* que, al igual que un *Frame*, es utilizado para encapsular los elementos dentro de esta, pero con la diferencia que en un *canvas* es posible trazar líneas y otras figuras. Posee los atributos *line1*, *line2* y *line3*, que dibujan líneas rectas una tras otra para conectar elementos que se encuentran posicionados diagonalmente, y diversos atributos que en el diagrama se encuentran contenidos como ‘coordenadas’ que almacenan las coordenadas de los distintos puntos necesarios para dibujar las líneas.

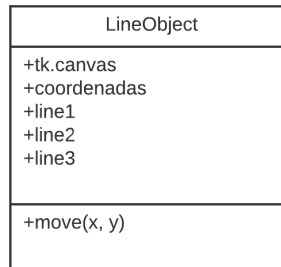


Figura 4.2: Diagrama UML - Clase LineObject

Este objeto, al igual que el anterior, sirve de entorno gráfico para la clase de líneas de transmisión que se presentará posteriormente. Dicho objeto es clave para conectar los distintos elementos creados en la interfaz gráfica. Utilizando la posición relativa que se obtienen de los objetos *FrameObject*, actualiza su dibujo en tiempo real cuando otro objeto es desplazado. Los métodos de la clase se encuentran en la Tabla 4.2.

Tabla 4.2: Métodos de la clase *LineObject*

Métodos	Descripción
.transparent(widget)	Convierte el fondo del <i>widget</i> en transparente.
.replace(x1, y1, x2, y2)	Traslada el objeto <i>canvas</i> según las coordenadas provistas.
.move(x1, y1, x2, y2)	Traslada la línea dibujada entre las terminales según las coordenadas provistas.
.clear()	Elimina todos los elementos dentro del <i>canvas</i>
.check_hand_enter()	Cambia la imagen del cursor cuando entra en el <i>canvas</i> .
.check_hand_leave()	Restablece la imagen del cursor cuando entra en el <i>canvas</i>
.update_labels()	Función 'dummy' para otro llamado.

4.2.2. Objetos de Datos

Las clases y por ende los elementos disponibles a utilizar en la interfaz son los que se describen a continuación.

4.2.2.1. Paneles Fotovoltaicos

Para modelar paneles fotovoltaicos se crea una clase particular para este fin. En la Figura 4.3 se muestra el diagrama UML que indica las relaciones de la clase y sus atributos.

La clase *Photovoltaic* debe ser necesariamente inicializada con un objeto de tipo *FrameObject* para que sea dibujado en la GUI. A su vez, la clase *Photovoltaic* contiene la información básica de los paneles como la potencia nominal, tensión nominal, junto al nombre utilizado para identificarlo y un atributo *oid* que almacena un identificador único generado con el estándar RFC 4122 [23] UUID (*Universally Unique Identifier*) en su versión 4.

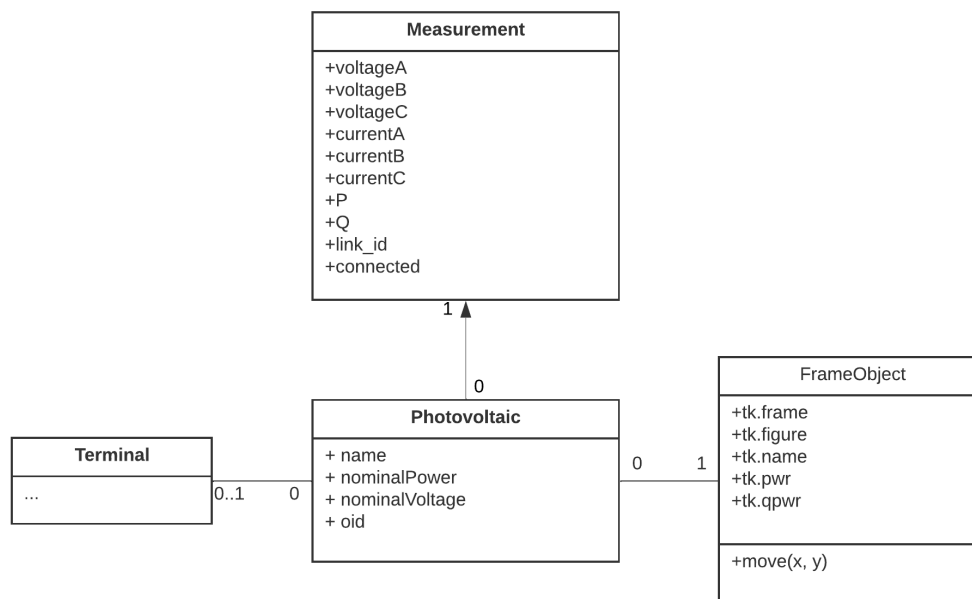


Figura 4.3: Diagrama UML - Paneles Fotovoltaicos

Los métodos de esta clase están relacionados a la gestión de sus mismos datos y de aquellos elementos que se encuentran referenciados en él, como, por ejemplo, lo pueden ser alarmas, líneas de transmisión, entre otros. Los métodos de la clase se encuentran en la Tabla 4.3.

Tabla 4.3: Métodos de la clase *Photovoltaic*

Métodos	Descripción
.delete()	Elimina el objeto de la base de datos y del directorio de objetos. Además, borra las líneas conectadas, alarmas, y el objeto gráfico.
.actualizar()	Actualiza cambios realizados a partir del formulario de modificación.
.delete_alarms()	Elimina las alarmas del objeto.

4.2.2.2. Línea de Transmisión

Para modelar líneas de transmisión se crea la clase *LineSegment*. En la Figura 4.4 se muestra el diagrama UML que indica las relaciones de la clase y sus atributos. Esta clase debe ser inicializada en conjunto con un objeto de tipo *LineObject* y los dos objetos que se quieren conectar.

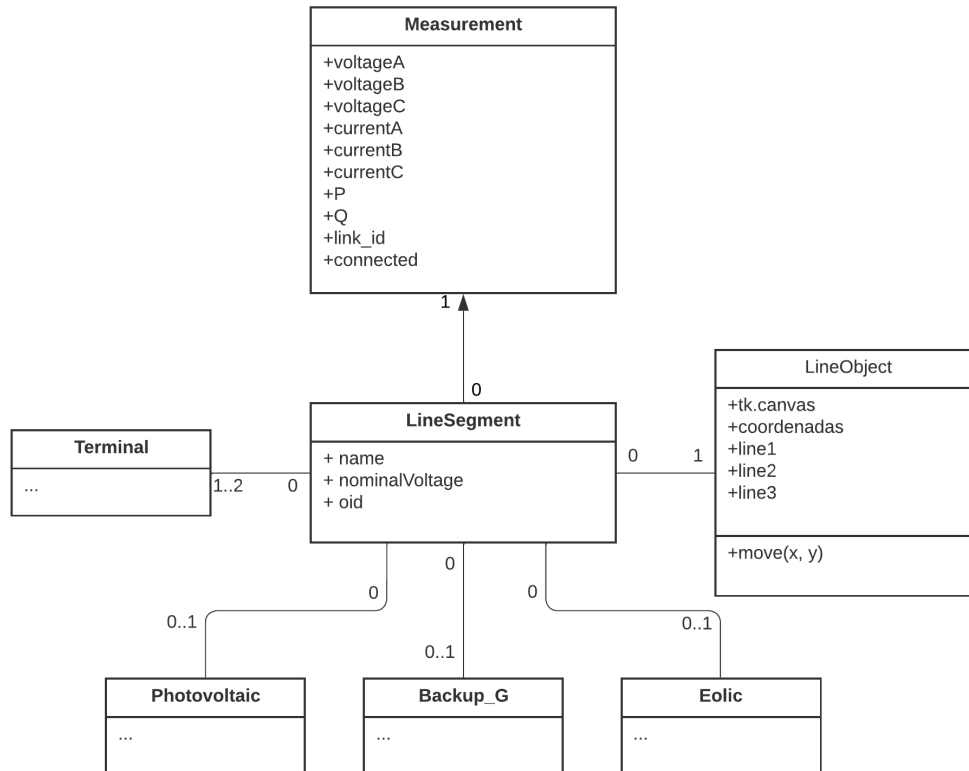


Figura 4.4: Diagrama UML - Segmento de Linea

Esta clase, al igual que las anteriores, debe ser inicializado en conjunto con un objeto *FrameObject* y, en relación a sus métodos, posee los mismos que sus predecesores. Sin embargo, tiene dos métodos que son utilizados para obtener las posiciones en las que se conectan las líneas de transmisión, que se conectan a esta subestación, en función de cuántas líneas se encuentran a la izquierda y a la derecha de esta. Los métodos de esta clase se encuentran en la Tabla 4.4

Tabla 4.4: Métodos de la clase *LineSegment*

Métodos	Descripción
.delete()	Elimina el objeto de la base de datos y del directorio de objetos.
.reset()	Método auxiliar para cuando se modifican los terminales de la línea.
.get_other_terminal(terminal)	Retorna el terminal contrario a <i>terminal</i> de la línea.
.actualizar(terminal)	Actualiza los <i>labels</i> del objeto FrontEnd asociado a esta línea.
.delete_alarms()	Elimina las alarmas del objeto.

4.2.2.3. Subestaciones

Para modelar subestaciones se crea la clase *Terminal*.

La característica de este objeto es la implementación de métodos que permiten obtener una lista de los elementos a los cuales se encuentra conectado mediante una línea de transmisión y separarlos según la dirección en la que se encuentran. De esta manera, las líneas que llegan al objeto se pueden distribuir equitativamente en el terminal. Los métodos de esta clase se encuentran en la Tabla 4.5.

Tabla 4.5: Métodos de la clase *Terminal*

Métodos	Descripción
<code>.delete()</code>	Elimina el objeto de la base de datos y del directorio de objetos. Además, borra las líneas conectadas, alarmas y el objeto gráfico.
<code>.pos_lines_x()</code>	Retorna dos listas con los elementos que se encuentran a la izquierda y a la derecha del terminal, ordenados de mayor a menor por posición en el eje y.
<code>.pos_lines_y()</code>	Retorna dos listas con los elementos que se encuentran en la parte superior e inferior del terminal, ordenados de mayor a menor por posición en el eje x.
<code>.actualizar(terminal)</code>	Actualiza los labels del objeto FrontEnd asociado a este terminal.
<code>.delete_alarms()</code>	Elimina las alarmas del objeto.

4.2.2.4. Transformadores

Para modelar transformadores de poder se crea la clase *Trafo*.

La clase posee los mismos métodos que las clases anteriores, aunque con la particularidad de que posee dos atributos para definir la tensión nominal en baja tensión y en alta tensión. Los métodos de esta clase se encuentran en la Tabla 4.6.

Tabla 4.6: Métodos de la clase *Trafo*

Métodos	Descripción
<code>.delete()</code>	Elimina el objeto de la base de datos y del directorio de objetos. Además, borra las líneas conectadas, alarmas y el objeto gráfico.
<code>.pos_lines_x()</code>	Retorna dos listas con los elementos que se encuentran a la izquierda y a la derecha del transformador ordenados de mayor a menor por posición en el eje y.
<code>.pos_lines_y()</code>	Retorna dos listas con los elementos que se encuentran en la parte superior e inferior del transformador, ordenados de mayor a menor por posición en el eje x.
<code>.actualizar(terminal)</code>	Actualiza los labels del objeto FrontEnd asociado a este terminal.
<code>.delete_alarms()</code>	Elimina las alarmas del objeto.

4.2.2.5. Consumo

Para modelar consumos se crea la clase *Load*.

Los métodos de esta clase se encuentran en la Tabla 4.7.

Tabla 4.7: Métodos de la clase *Load*

Métodos	Descripción
<code>.delete()</code>	Borra el objeto <i>Load</i> de la base de datos y del directorio de objetos. Además, borra las líneas conectadas, alarmas y el objeto gráfico.
<code>.actualizar()</code>	Actualiza cambios realizados a partir del formulario de modificación.

4.2.3. Otros Objetos

Dentro de esta sección, se detallan otras clases creadas pero que no corresponden a objetos gráficos o de datos.

4.2.3.1. Alarma

Para definir alarmas a cualquier elemento de la red, se crea la clase *Alarm*.

Dentro de sus atributos, este objeto almacena: el nombre de la alarma, un código identificador único, el elemento al cual se le asigna la alarma, la variable a monitorear, la condición que cumplir para la activación de la alarma, el umbral de comparación y la prioridad de la alarma. Una alarma puede ser creada directamente haciendo click derecho en cualquier elemento gráfico de la red o a través del *Alarm Manager*.

Con todos los atributos mencionados anteriormente, un *loop* central puede iterar sobre estos objetos y procesar si las variables superan un valor de umbral y dar el correspondiente aviso en la consola del sistema. La Tabla 4.8 muestra los métodos de esta clase.

Tabla 4.8: Métodos de la clase *Alarma*

Métodos	Descripción
<code>.delete()</code>	Borra el objeto <i>Alarm</i> de la base de datos y del directorio de objetos.

4.2.3.2. Alarm Manager

El entorno de gestión de alarmas es llamado *Alarm Manager*. A partir de este se pueden visualizar, eliminar y crear nuevas alarmas. La Tabla 4.9 muestra los métodos de esta clase.

Tabla 4.9: Métodos de la clase *Alarm Manager*

Métodos	Descripción
<code>.borrar_alarma()</code>	Borra el objeto <i>Alarm</i> seleccionado de la tabla contenida en el <i>AlarmManager</i>
<code>.update()</code>	Actualiza la tabla del <i>AlarmManager</i> cuando se crea o elimina un objeto <i>Alarm</i>

4.2.3.3. Historical Data

Para visualizar información histórica de un elemento de la red mientras el proceso de monitoreo se encuentra activo, se implementa la clase *History Data*. El objeto debe ser inicializado con un objeto de datos, para que posteriormente pueda buscar la información histórica en la base de datos.

Tabla 4.10: Métodos de la clase *Historical Data*

Métodos	Descripción
<code>.plot_data()</code>	Consulta la base de datos por el nombre y variable del objeto buscado; gráfica los resultados. Inicializa <code>.update_data()</code> .
<code>.update_data()</code>	Consulta la base de datos cada ciertos intervalos configurado por el usuario y actualiza con valores recientes.
<code>.clear_frame()</code>	Detiene la ejecución de <code>.update_data()</code> y limpia el gráfico.
<code>.on_closing()</code>	Detiene los procesos pendientes del objeto y ordena el cierre de la ventana.

4.2.4. Base de Datos

En la siguiente sección, se detalla la estructura de las bases de datos del programa. En esta base se almacenan los datos que son procesados para que se puedan importar redes ya creadas, o exportar una red recién configurada y guardarla para ser utilizada en otro momento. La estructura de los datos recibidos desde los dispositivos de medición, también será tratado en esta sección y cómo se permite el acceso concurrencido a esta, alternando entre consultas de información y de almacenamiento.

4.2.4.1. Datos de Red

Los datos principales de los elementos de la red son almacenados en tablas que, a su vez, estas son parte de un archivo HDF5. Las tablas creadas son cuatro: alarmas, elementos estáticos, elementos de segmentación y medidores. Cada una de las tablas posee una estructura apropiada para almacenar la información necesaria para reconstruir la red.

4.2.4.2. Datos de Medición

Los datos recibidos desde los dispositivos de medida son almacenados en una archivo `.csv` (comma separated values). Los datos que se almacenan son: código identificador del elemento, nombre del elemento, la estampa de tiempo de la medición, la variable medida y, por último,

el valor medido. En la Figura 4.5 se muestran los datos almacenados de una simulación que se realizó al final de esta sección.

Cabe mencionar, que la estampa de tiempo es calculada utilizando como suposición que el tiempo que demora en llegar la solicitud de medición al esclavo Modbus, es igual al tiempo que demora en llegar el paquete con la información desde el esclavo Modbus.

uuid	name	datetime	variable	value
0de4f174-105c-4aab-8003-22ce8c507021	Barra 5	2022-12-19 20:19:27.989561	Potencia Activa 3P	238
53d00fe2-4a19-4b4a-9ead-1b2f79ff6a7b	Barra 6	2022-12-19 20:19:28.049036	Potencia Activa 3P	233
40ff2740-dcc8-45b2-acc1-7cd1ea422615	Barra 1	2022-12-19 20:19:28.754336	Potencia Activa 3P	261
7e43668e-8b87-4f9a-b894-2c587e8758b5	Barra 2	2022-12-19 20:19:28.812703	Potencia Activa 3P	254
f37d95ab-c40e-4d34-a6d9-1000a335a08c	Barra 3	2022-12-19 20:19:28.873104	Potencia Activa 3P	249
a098871a-3e16-4815-aa73-24fbbdf79f87	Barra 4	2022-12-19 20:19:28.936267	Potencia Activa 3P	244
0de4f174-105c-4aab-8003-22ce8c507021	Barra 5	2022-12-19 20:19:28.995144	Potencia Activa 3P	239
53d00fe2-4a19-4b4a-9ead-1b2f79ff6a7b	Barra 6	2022-12-19 20:19:29.059491	Potencia Activa 3P	234
40ff2740-dcc8-45b2-acc1-7cd1ea422615	Barra 1	2022-12-19 20:19:29.761529	Potencia Activa 3P	262
7e43668e-8b87-4f9a-b894-2c587e8758b5	Barra 2	2022-12-19 20:19:29.817117	Potencia Activa 3P	255
f37d95ab-c40e-4d34-a6d9-1000a335a08c	Barra 3	2022-12-19 20:19:29.879227	Potencia Activa 3P	250
a098871a-3e16-4815-aa73-24fbbdf79f87	Barra 4	2022-12-19 20:19:29.941450	Potencia Activa 3P	245
0de4f174-105c-4aab-8003-22ce8c507021	Barra 5	2022-12-19 20:19:30.003414	Potencia Activa 3P	240

Figura 4.5: Estructura de datos almacenados en archivo .csv

4.2.4.3. Algoritmo de Adquisición de Datos

El algoritmo de adquisición de datos se presenta en la Figura 4.6. El proceso se inicia cuando el botón 'Iniciar' de la ventana principal es presionado. En el inicio, se inicializan las variables necesarias para el resto del proceso, esto es:

- **Lock:** la llave para controlar el acceso a la base en donde se almacenan los datos de medición. Esto permite que no exista un acceso simultáneo entre este proceso y la clase *HistoricalData* al leer la data histórica para graficar.
- **ms:** indica cada cuantos milisegundos se debe solicitar nuevas mediciones.
- **Elemento:** lista de todos los elementos de la red.
- **Alarma:** lista de todas las alarmas de la red.

Para cada elemento de la red, se intenta hacer una conexión con el dispositivo de medida asociado. Si la conexión es exitosa, se leen los registros del esclavo Modbus, se actualizan los 'labels' del objeto *FrameObject* asociado al elemento y, utilizando la llave *Lock*, se accede a la base de datos para escribir la nueva medición.

Al finalizar el loop anterior, se ejecutan dos procesos. El proceso de obtención de datos se sigue procesando en el *Thread* principal, mientras que el proceso se inicia en un *Thread* distinto. Se inicia un *loop* con todas las alarmas creadas, verificando que las variables no hayan activado ninguna condición. Si una variable supera la condición y, la alarma no ha

sido activada anteriormente, entonces se imprime en la consola. De lo contrario, sigue iterando con la siguiente alarma, hasta finalizar con la lista.

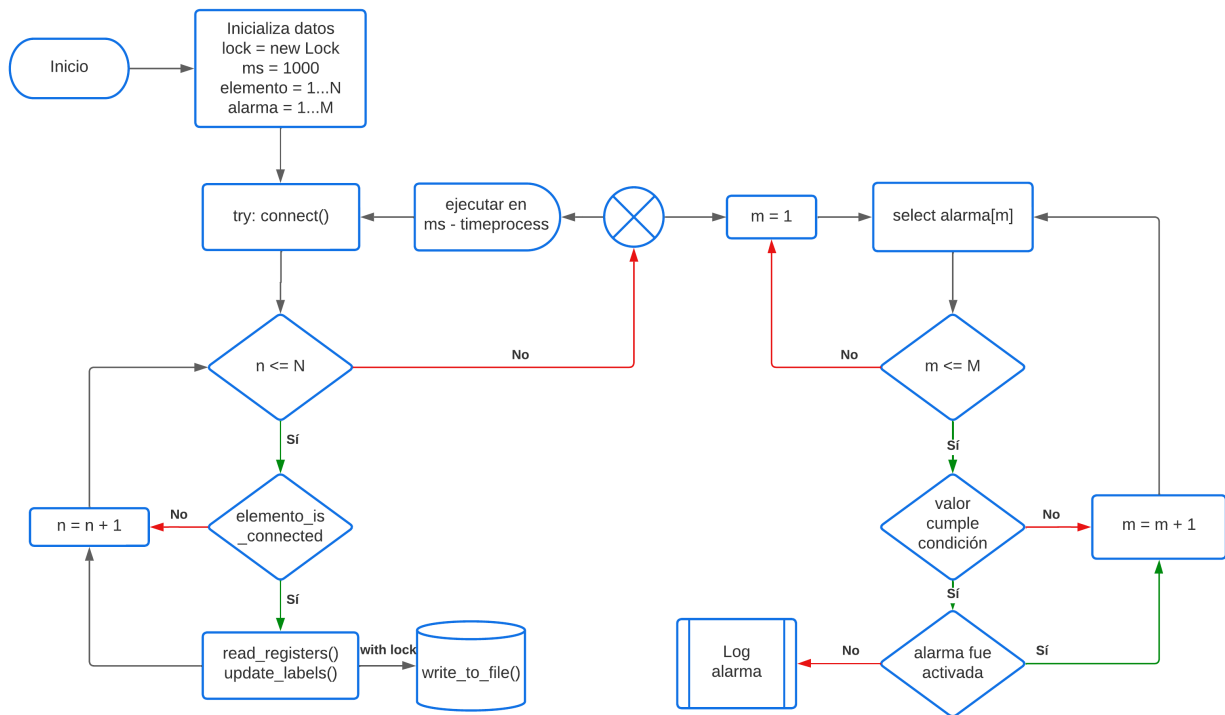


Figura 4.6: Diagrama de flujo del algoritmo de adquisición de datos

4.3. Implementación de la Micro-red del Laboratorio

En esta sección se muestran los pasos para la creación de una micro-red en el programa SCADA, desde la configuración general de los elementos y su distribución en la red.

El programa, en una primera instancia, solicitará al usuario un nombre para crear un archivo de datos HDF5, o alternativamente, el usuario puede importar uno ya creado con el botón '+' como se muestra en la Figura 4.7. Si la creación/importación finaliza sin problemas, se envía un mensaje a la consola que informará el suceso.

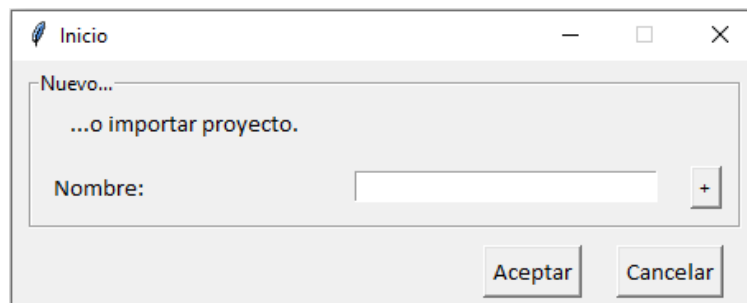


Figura 4.7: Ventana de inicio

El archivo una vez abierto, la atención se dirigirá a la ventana presentada en la Figura 4.8. En esta ventana se pueden distinguir los siguientes elementos:

1. **Panel solar:** abre formulario para configurar y crear paneles solares.
2. **Barra:** abre formulario para configurar y crear una barra.
3. **Línea de transmisión:** abre formulario para configurar y crear una línea de transmisión.
4. **Transformador de poder:** abre formulario para configurar y crear un transformador de poder.
5. **Consumo:** abre formulario para configurar y crear un nuevo consumo.
6. **Iniciar:** inicializa el proceso de monitoreo de la red.
7. **Detener:** detiene el proceso de monitoreo de la red.
8. **Consola:** en la consola se observa información importante relacionado a alarmas, instrucciones, creación y modificación de elementos.



Figura 4.8: Ventana principal

En esta ventana se observan dos menús desplegables. En 'File' se encuentran las opciones para crear, guardar, exportar e importar archivos al sistema. En 'Alarm Manager' se encuentra la opción para abrir el gestor de alarmas.

En la Figura 4.9 se presenta el formulario para crear nuevos elementos. El caso particular que se presenta es para crear paneles solares, en donde solo el recuadro 'Configuración general' es diferente para cada elemento. En este formulario se puede definir el nombre, potencia nominal, tensión nominal, etc. En la 'Configuración IP' se debe escoger la dirección IP del

medidor, el número esclavo de la dirección IP (si corresponde) y las direcciones a consultar del medidor, los cuales dependerán de cada dispositivo de medida. Si no se escoge un número de registro, la variable no será consultada. De igual manera, si no se define una dirección IP para el elemento, se da por entendido que el elemento no tiene dispositivo de medida asociada y no se intentará realizar la conexión.

Un caso especial de formulario es la línea de transmisión. En este caso, se debe escoger el inicio y final de la línea, considerando que cada uno de estos debe ser otro elemento de la red. Una línea de transmisión solo puede existir si sus dos terminales existen. Todas las líneas de transmisión que estén conectadas a un terminal cualquiera serán eliminadas si el terminal en cuestión es eliminado (eliminación en cascada). De la misma manera, se eliminarán todos los elementos que guardan relación al terminal.

Configuración general	
Nombre:	Angamos
Potencia Nominal [W]:	100
Tensión Nominal [V]:	23000
Terminal:	Null

Configuración IP	
Nombre medidor:	
Dirección IP:	
Slave Unit:	0
Reg. de Tensión AN	
Reg. de Tensión BN	
Reg. de Tensión CN	
Reg. de Tensión AB	
Reg. de Tensión BC	
Reg. de Tensión CA	
Reg. de Tensión FF	
Reg. de Tensión FN	
Reg. de Corriente Media	
Reg. de Corriente A	
Reg. de Corriente B	
Reg. de Corriente C	
Reg. de Corriente Neutro	
Reg. de Potencia Activa A	
Reg. de Potencia Activa B	
Reg. de Potencia Activa C	
Reg. de Potencia Activa 3P	
Reg. de Potencia Reactiva A	
Reg. de Potencia Reactiva B	
Reg. de Potencia Reactiva C	
Reg. de Potencia Reactiva 3P	
Reg. de Frecuencia	
Reg. de Factor de Potencia	

Figura 4.9: Formulario para crear nuevo elemento

Al dar click en 'Aceptar', la ventana es cerrada y se visualizará el elemento en la ventana principal, como se muestra en la Figura 4.10. Cada elemento que se despliegue en la ventana principal, tiene la capacidad de ser desplazado, manteniendo presionado click izquierdo sobre él y moviendo el ratón por la pantalla. Al hacer click derecho sobre un elemento, se desplegará

un nuevo menú cómo el que se muestra en la Figura 4.10, con las siguientes opciones:

1. **Historical Data:** despliega una nueva ventana con una lista de variables para graficar del elemento.
2. **Parameters:** despliega un formulario para modificar la configuración del elemento. El formulario es el mismo de la Figura 4.9.
3. **Set Alarm:** abre un nuevo formulario para crear una alarma del elemento seleccionado.
4. **Rotate:** rota el elemento gráfico en -90° .
5. **Delete:** elimina el elemento.

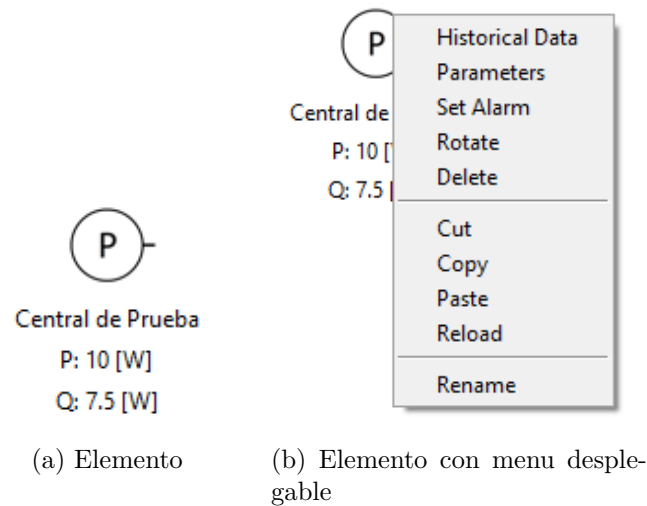


Figura 4.10: Elemento en la ventana principal

Utilizando como referencia la topología de la red mostrada en la Figura 3.1, se configura la micro-red del laboratorio, obteniendo así una representación de esta que se observa en la Figura 4.11.

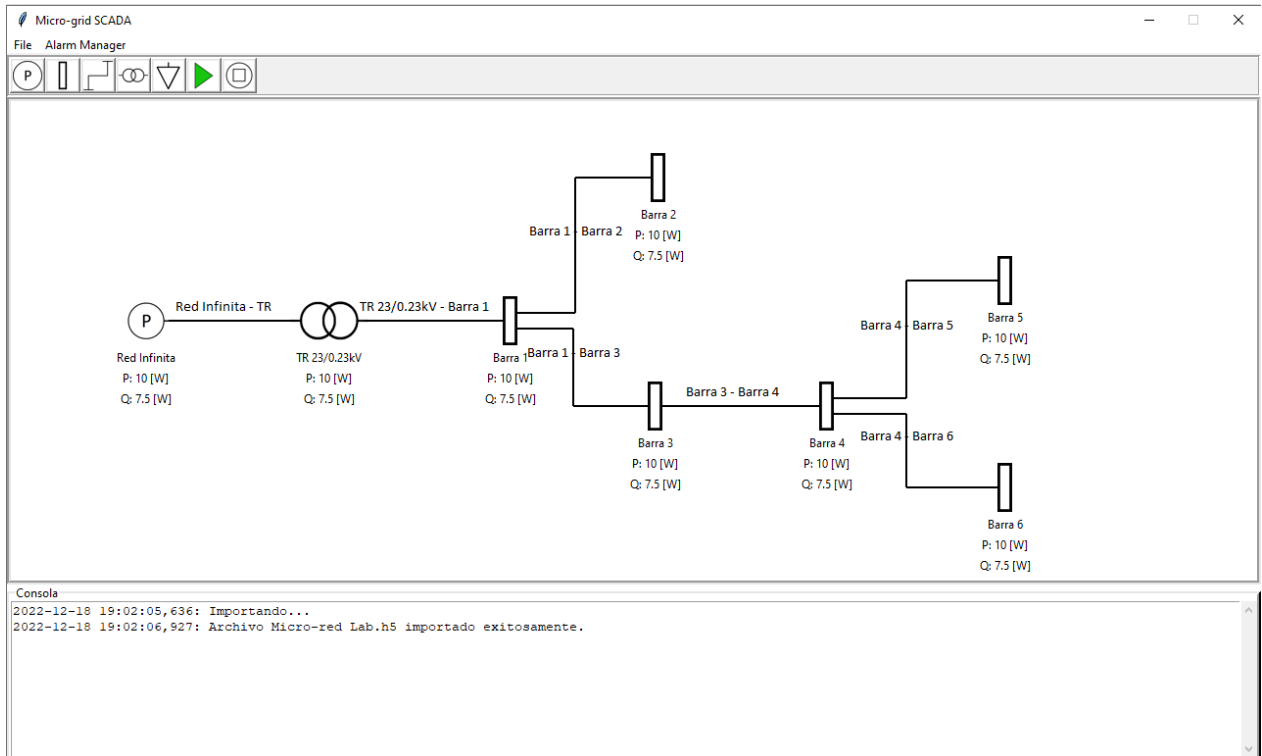


Figura 4.11: Micro-red del laboratorio implementada en el software

4.3.1. Validación en un Entorno Simulado

El programa es validado en un entorno controlado como lo es dentro del mismo computador, utilizando simuladores de esclavos Modbus a los cuales se les asigna la dirección IP del *host* local. En el caso de la micro-red del laboratorio, en cada barra existe un dispositivo de medida, por lo tanto, se configuran seis esclavos Modbus para representar a las barras.

La simulación de los esclavos Modbus se realiza mediante el programa gratuito *diagslave* [24], en el cual solo es necesario especificar la unidad esclavo dentro del localhost (IP: 127.0.0.1). Para probar la funcionalidad del programa, se solicitará la lectura de los registros 40016 y 40020 que corresponden a los registros de solo lectura de los dispositivos Modbus, que en este caso, se asocian a la variable de Potencia Activa Total y Potencia Reactiva Total, respectivamente. En algunos dispositivos Modbus, la dirección del registro debe ser absoluta (4xxxx), mientras que en otros relativa (xxxx). Lo anterior es debido a que al solicitar la lectura de los registros, ya se está haciendo referencia a las direcciones 40000. En la Figura 4.12 se observa la configuración utilizada para conectar el SCADA a los esclavos Modbus simulados.

Configuración IP	
Nombre medidor:	PM750-1
Dirección IP:	127.0.0.1
Slave Unit:	1
Reg. de Tensión AN	
Reg. de Tensión BN	
Reg. de Tensión CN	
Reg. de Tensión AB	
Reg. de Tensión BC	
Reg. de Tensión CA	
Reg. de Tensión FF	
Reg. de Tensión FN	
Reg. de Corriente Media	
Reg. de Corriente A	
Reg. de Corriente B	
Reg. de Corriente C	
Reg. de Corriente Neutro	
Reg. de Potencia Activa A	
Reg. de Potencia Activa B	
Reg. de Potencia Activa C	
Reg. de Potencia Activa 3P	16
Reg. de Potencia Reactiva A	
Reg. de Potencia Reactiva B	
Reg. de Potencia Reactiva C	
Reg. de Potencia Reactiva 3P	20
Reg. de Frecuencia	
Reg. de Factor de Potencia	

Figura 4.12: Configuración de un esclavo Modbus simulado para la barra 1

La experiencia a simular es la siguiente: cada 5 segundos, la potencia activa de una barra se activará, lo que significa que tendrá una rampa de subida de 1 W/s , luego se activarán las potencias reactivas de la misma manera, con una rampa de subida de 1 W/s .

Iniciando el monitoreo con el botón 'Iniciar', se realizan las conexiones a las direcciones IP configuradas en cada uno de los elementos, como se muestra en la Figura 4.13. En la consola se observa que las conexiones fueron exitosas. Si alguna conexión falla por cualquier motivo, en la consola se indicará el momento de fallo. Solo los elementos que fueron configurados para ser monitoreados (las barras), están activamente actualizando sus valores, mientras que los que no son configurados (la barra infinita y el transformador) no cambian sus valores.

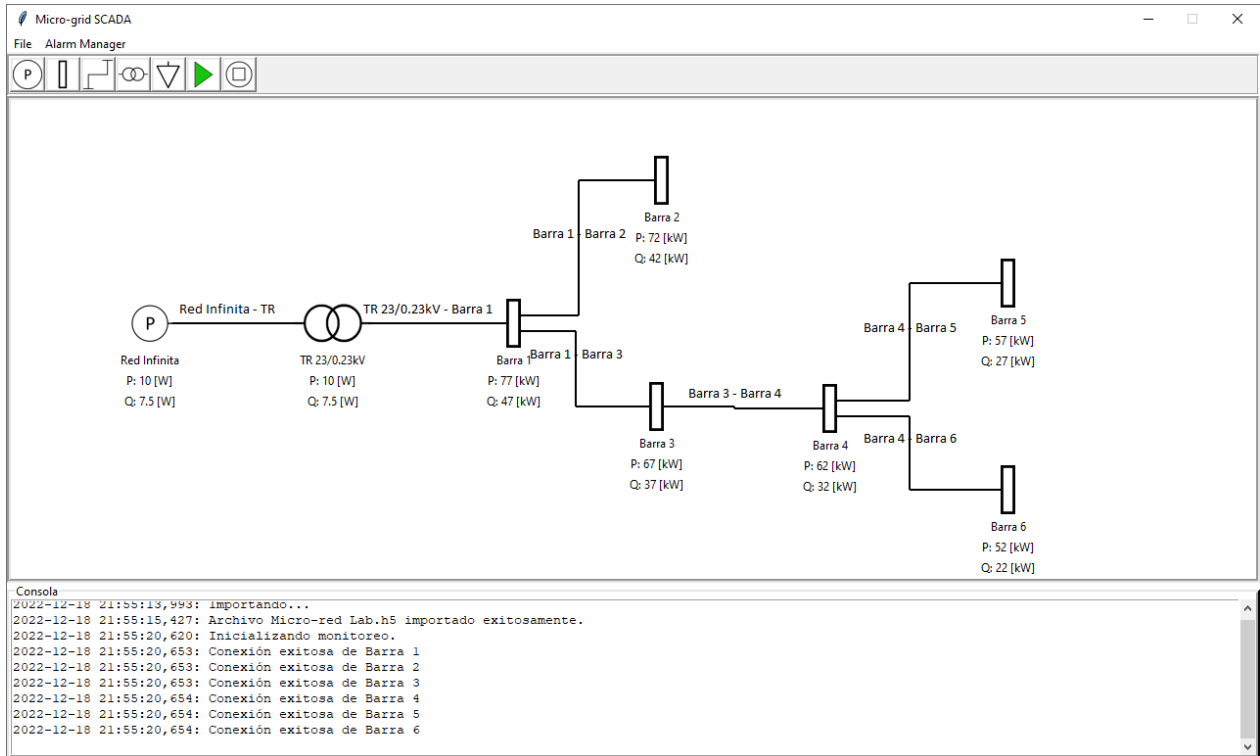


Figura 4.13: Monitoreando activamente la micro-red (simulación)

En la Figura 4.14 se muestra la ventana principal del programa al finalizar el monitoreo con el botón 'Detener'. En la consola se imprime que las conexiones a los dispositivos han sido cerradas y que el algoritmo de adquisición ha sido detenido.

En las Figuras 4.15 y 4.16 se grafican los valores obtenidos de potencia activa y de potencia reactiva, en donde se muestra el comportamiento esperado según la simulación.

Sin embargo, en ambas figuras se puede observar que existe una brecha en ciertos momentos para cada barra. Este comportamiento tiene dos aristas que tienen que ser exploradas. En primer lugar, la brecha existe por consecuencia de la manera en que se grafican los datos. Para que estos fueran comprensibles en un gráfico, solo se consideró hasta el segundo para crear el eje x, cuando en realidad, todas las mediciones tienen precisión en los milisegundos. Pero el problema va mas allá, en que no se reciben mediciones cada 1000 [ms] como fue configurado, sino con un retraso de mayor magnitud, por lo que cada cierto periodo de tiempo, la medición se salta un segundo. Esto se puede corroborar en la Figura 4.17, que se obtiene al obtener las diferencias en [ms] entre cada medición y comparando el resultado con 1000 [ms].

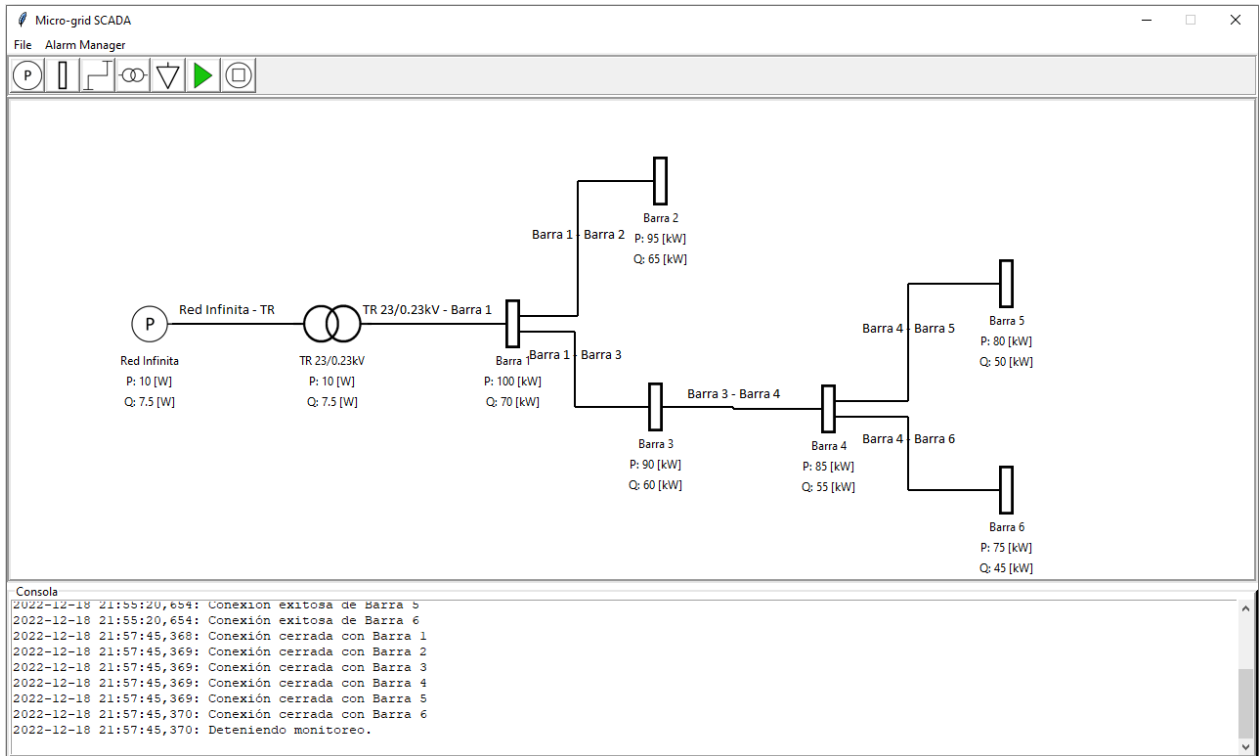


Figura 4.14: Deteniendo el monitoreo de la micro-red (simulación)

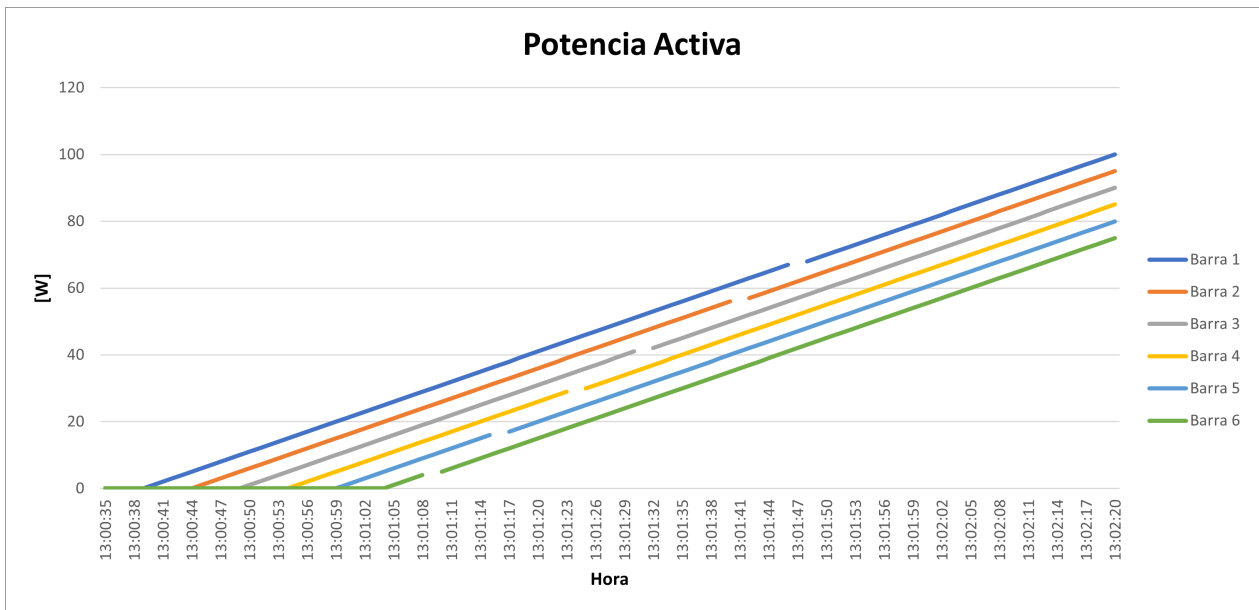


Figura 4.15: Potencia activa obtenida de los esclavos Modbus simulados

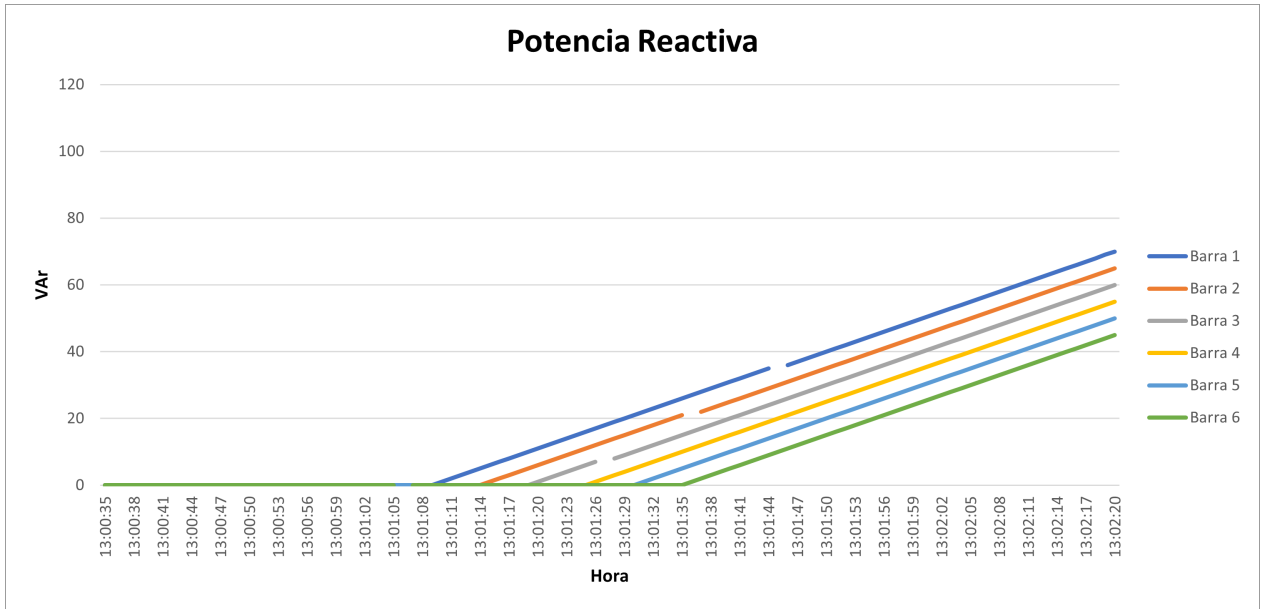


Figura 4.16: Potencia reactiva obtenida de los esclavos Modbus simulados



Figura 4.17: Diferencia entre intervalos de medición con respecto a 1000 [ms]

La diferencia promedio entre el intervalo de medición, con respecto a lo configurado, es de $7,64[ms]$, lo que significa que, en promedio, cada 131 mediciones, la medición en el segundo siguiente no existe.

En la Figura 4.18 se observa la ventana de la clase *HistoricalData* en la cual permite observar el comportamiento histórico de una variable durante los momentos en que el monitoreo se encuentra activo.

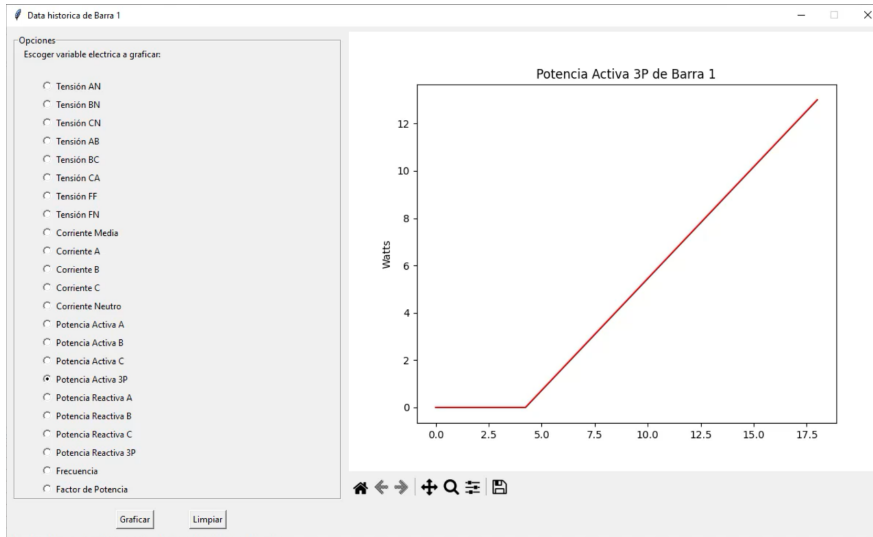


Figura 4.18: Monitoreo de variables en tiempo real mediante gráfico

En la Figura 4.19 se observa la ventana del gestor de alarmas, donde existe una alarma creada para dar un aviso cuando la potencia activa total de la barra 1 supere los 20 [W].

The screenshot shows a window titled "Alarm Manager". It contains a table with the following data:

Alarma	Elemento	Variable	Condicion	Prioridad
Alarma 1	Barra 1	Potencia Activa 3P	>20	Normal

Below the table, there is a section labeled "Comandos" with two buttons: "Crear Alarma" and "Borrar Alarma".

Figura 4.19: Ventana del gestor de alarmas

Finalmente, en la Figura 4.20 se observa el momento en donde la potencia activa de la barra 1 supera los 20 [W], imprimiendo la correspondiente alarma en la consola.

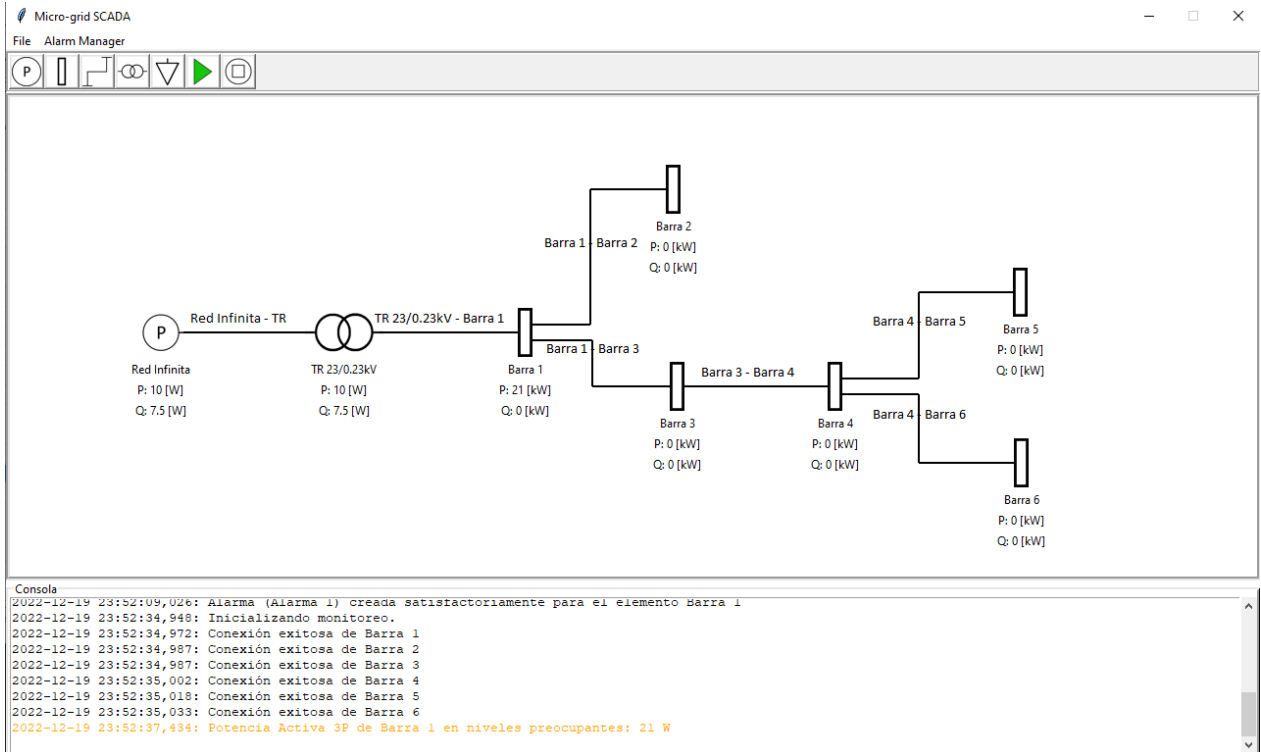


Figura 4.20: Activación de la alarma cuando la variable supera el umbral

4.3.2. Validación Experimental en el Laboratorio

Luego de haber testeado el SCADA en un entorno controlado, se procede a instalarlo en el Laboratorio de Energía de la Universidad de Chile, configurando cada dispositivo de medida en su correspondiente barra. Se realizaran dos experimentos con el Micro-grid SCADA: el primer experimento es monitorear la tensiones F-F y F-N en todos los nudos de la micro-red durante 30 minutos.

El segundo experimento corresponde a monitorear las principales variables eléctricas en una barra, es decir, tensión F-F, tensión F-N, corriente media, potencia activa, potencia reactiva, frecuencia y factor de potencia, durante se realizan las siguientes maniobras:

1. En $t = 0 [s]$ se conecta el Micro-grid SCADA al dispositivo de medida de la barra 4, mientras que no existe consumo en la barra.
2. En $t = 10 [s]$ se conecta una carga trifásica variable en vacío a la barra.
3. En $t = 60 [s]$ empieza a aumentar la carga con una rampa lenta, hasta alcanzar 1 [kW].
4. En $t = 120 [s]$ empieza a aumentar la carga con un rampa rápida, hasta alcanzar una potencia demandada de 3 [kW].
5. En $t = 200 [s]$ se regula la potencia reactiva de la carga trifásica, para alcanzar la tensión nominal de 380 [V] F-F.
6. En $t = 230 [s]$ se desconecta intempestivamente la carga.

4.3.2.1. Perfil de Tensión de la Micro-red

A continuación, se muestran los resultados del primer experimento con el sistema SCADA. En la Figura 4.21 se observa el promedio en cada minuto del perfil de tensión F-F en las barras de la micro-red, mientras que en la Figura 4.22 se observa el promedio en cada minuto del perfil de tensión F-N en las barras de la micro-red.

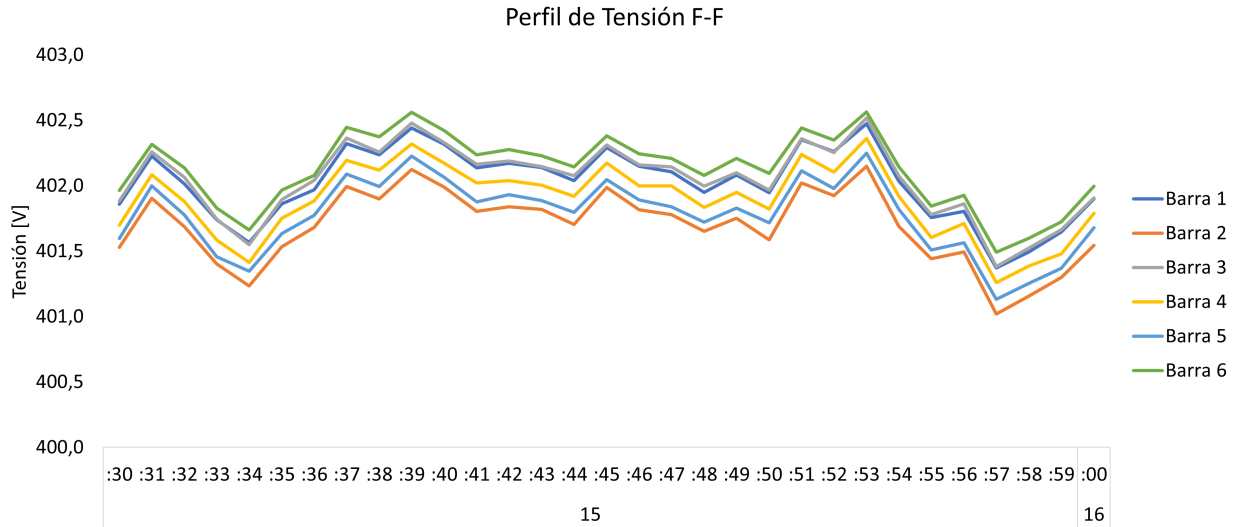


Figura 4.21: Perfil de tensión F-F de la micro-red

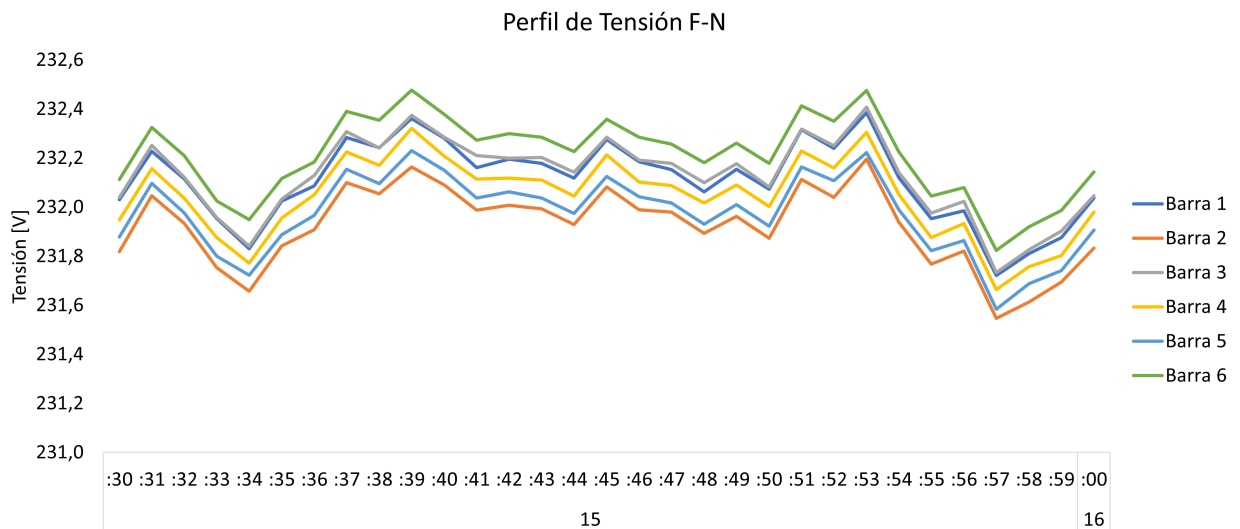


Figura 4.22: Perfil de tensión F-N de la micro-red

Teniendo en consideración que la Barra 1 es la mas cercana a la red, luego la Barra 2 con la Barra 3, y así sucesivamente hasta llegar a la Barra 6, cabria esperar que en esta última con la Barra 5 se presentara el perfil con la menor tensión. Sin embargo, la Barra 6 es que posee el perfil de tensión más alto. Todas las barras fueron probadas para identificar el medidor que corresponde a cada barra, por lo que se descarta que los medidores estuvieran desordenados. Por otro lado, este comportamiento

4.3.2.2. Conexión de Carga Trifásica

En lo que sigue, se exponen los resultados obtenidos al llevar a cabo el segundo experimento con la micro-red del Laboratorio de Energía. En la Figura 4.23 se observa la potencia activa, potencia reactiva y la tensión F-F durante el experimento.

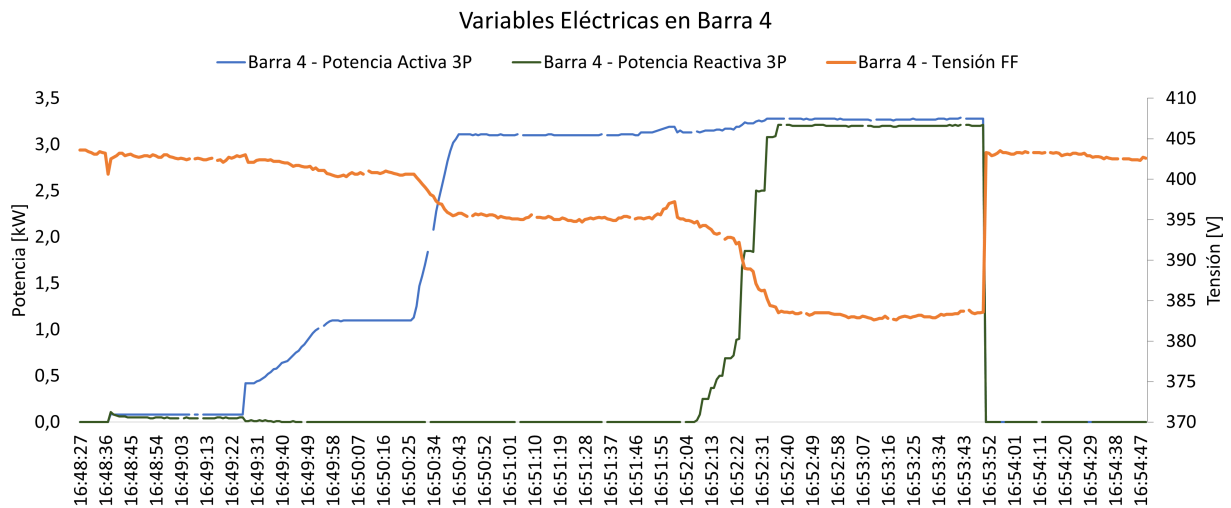


Figura 4.23: Potencia Activa, Potencia Reactiva y Tensión F-F en la Barra 4

En este gráfico, se puede observar el acoplamiento entre tensión y potencia activa, el cual es característico en redes resistivas, en donde $R > X$. Sin embargo, al elevar la potencia reactiva, la tensión baja con una pendiente mayor, lo cual delata que la parte inductiva de la impedancia no es tan despreciable con respecto a la resistiva.

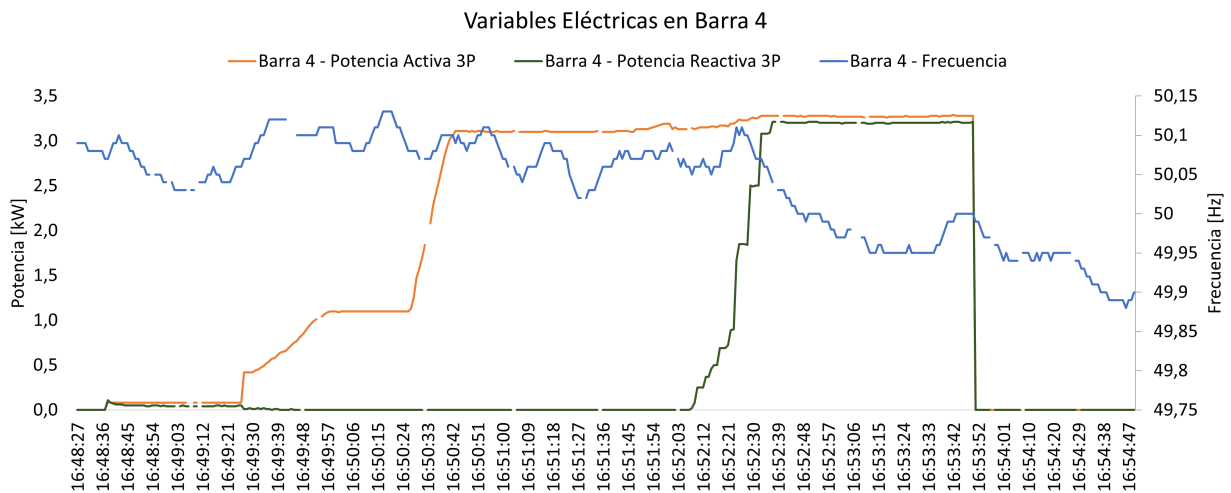


Figura 4.24: Potencia Activa, Potencia Reactiva y Frecuencia en la Barra 4

En la Figura 4.24 se grafican la potencia activa, la potencia reactiva y la frecuencia durante el experimento. En esta, no se observa un fuerte acoplamiento entre la frecuencia y la potencia activa, como es de esperar en redes resistivas. Más aún, si es posible identificar que

al elevar la potencia reactiva generada, la frecuencia decae notablemente, cayendo desde los 50,11 [Hz] hasta los 49,95 [Hz].

En la Figura 4.25 se observa la corriente media del consumo en la Barra 4, junto con la potencia activa y reactiva consumida. Se observa que la corriente media del consumo crece a medida que aumenta la potencia activa, y, posteriormente, la potencia reactiva.

En la Figura 4.26 se gráfica el factor de potencia del consumo, en conjunto con la potencia activa y reactiva de la carga. Mientras que la carga solo presenta parte resistiva, el factor de potencia se mantiene en 1, pero a medida que aumenta la potencia reactiva, el factor de potencia decae hasta llegar a los 0,7149.

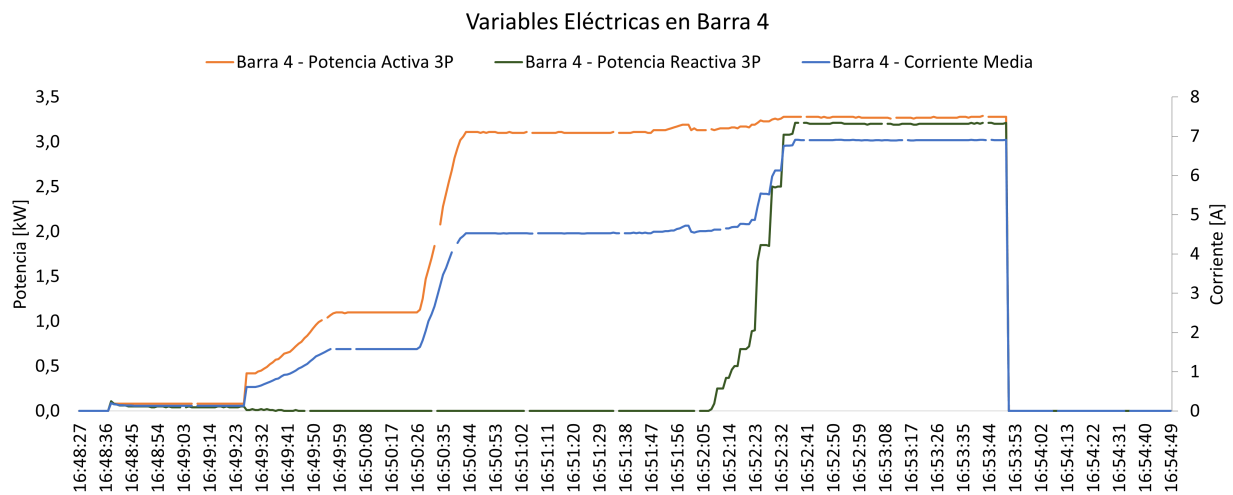


Figura 4.25: Potencia Activa, Potencia Reactiva y Corriente Media en la Barra 4

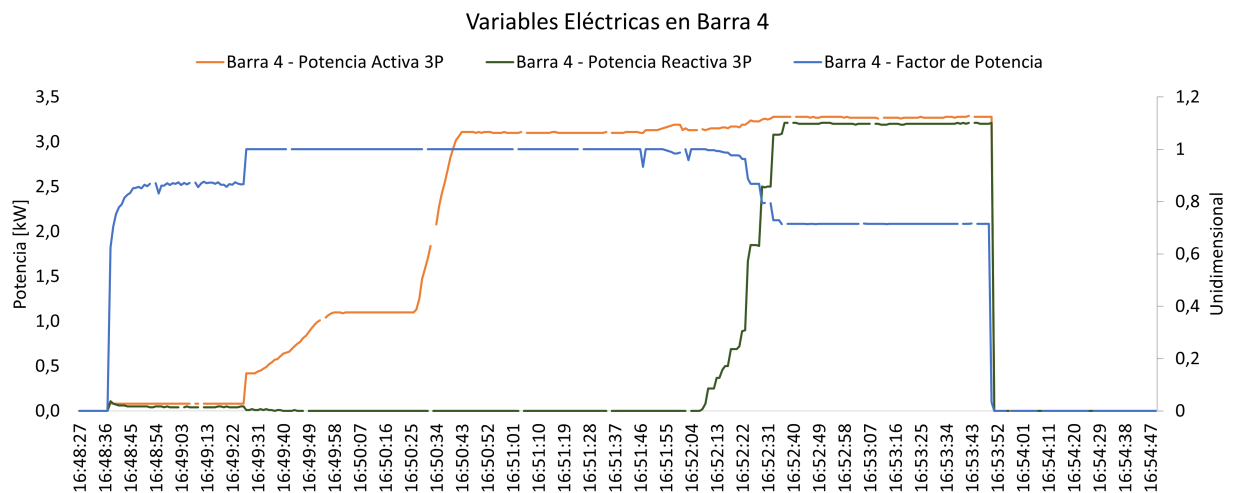


Figura 4.26: Potencia Activa, Potencia Reactiva y Factor de Potencia en la Barra 4

Por último, en la Figura 4.27 se muestra el perfil de tensión de la Barra 1 y 4, durante la

realización del experimento. Cuando aumenta la carga en $t = 60 [s]$, se observa una pequeña caída en tensión de la Barra 4 con respecto a la Barra 1, cuya diferencia aumenta más al subir la potencia a 3 [kW] en $t = 120 [s]$. Al aumentar la potencia reactiva en $t = 200 [s]$, la diferencia se hace aún mayor. Finalmente, al desconectar la carga intempestivamente, la diferencia de tensiones desaparece, y ambas vuelven a su valor inicial en $t = 0 [s]$.

En general, en todos los gráficos se observa que no existen mediciones en ciertos intervalos de tiempo. Este fenómeno es el mismo que ocurre en la validación en el entorno simulado.

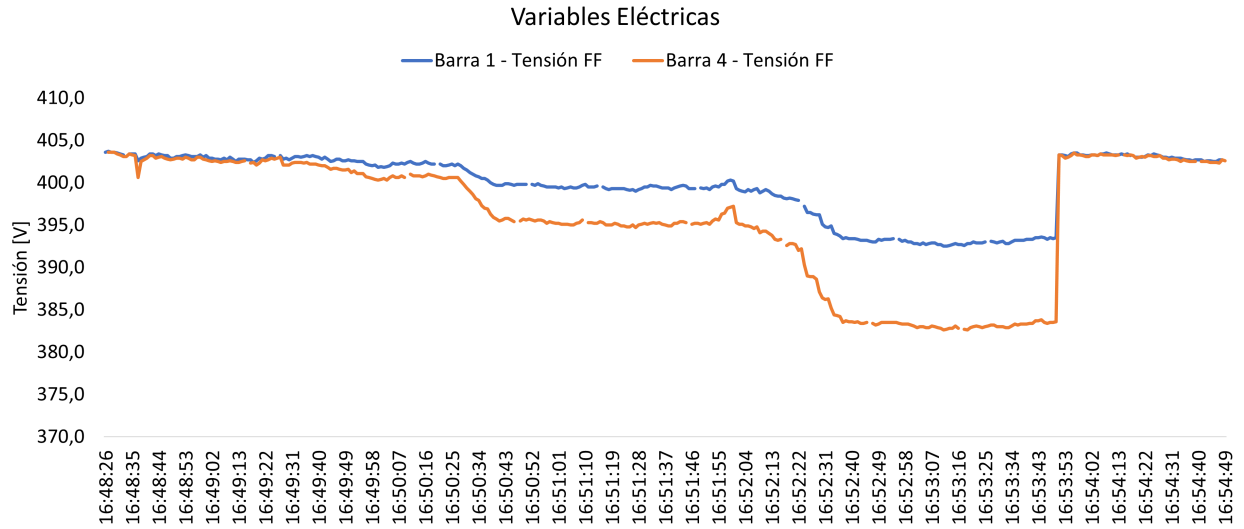


Figura 4.27: Perfil de Tensión en Barra 1 y 4

Capítulo 5

Conclusiones

En este trabajo de memoria se entregan los conocimientos básicos, a saber, sobre micro-redes, protocolos de comunicación y sistemas SCADA. Los tres conceptos mencionados están altamente relacionados, como se expone en el Capítulo 2. Las micro-redes requieren de una infraestructura de comunicaciones para llevar a cabo todas las operaciones que tiene para ofrecer y, es por aquello, que toma relevancia poner en discusión los sistemas SCADA y los protocolos de comunicación que lo conforman.

En la última década, se han publicado distintos protocolos de comunicaciones que se encuentran diseñados para su implementación en los sistemas eléctricos de potencia. Si bien estos toman en consideración problemas de sus antecesores y los solucionan en sus entregas, la documentación de estos es mucho más extensa y, por tanto, a pesar de la reducción en los esfuerzos de configuración gracias a su estandarización, también conlleva un alto costo de migración por las nuevas tecnologías. Es por esto, que aun predomina el uso de Modbus y DNP3.0, nombrados en orden por su antigüedad.

Con respecto a los sistemas SCADA, se ha puesto en evidencia que los productos comerciales no logran satisfacer por completo las necesidades de las micro-redes. Si bien, las soluciones dirigidas a las micro-redes tienen la mayoría de las capacidades para su operación como tal, no poseen una visualización activa del estado de la red. Lo anterior no es aceptable para micro-redes que conforman comunidades extensas que poseen una alta penetración de recursos distribuidos. En cambio, los sistemas SCADA para la gran industria, cuentan con una gran capacidad visual, pero en el apartado operativo no cumple con lo necesario para ser instalado en una micro-red.

En base a lo expuesto, en esta memoria se aborda la concepción de un sistema SCADA que logre cumplir con los requerimientos de las micro-redes, tanto en el apartado operacional y de visualización. Bajo los alcances de este trabajo, se implementa y con éxito un sistema de adquisición de datos y control sobre micro-redes que opera bajo el protocolo Modbus. El código del programa es escrito en el lenguaje de programación Python y, en su mayor parte, con construcción de objetos para hacer más fácil su lectura y posterior modificación.

El programa SCADA logra conectarse con éxito a esclavos Modbus presentes en la red Ethernet, permitiendo el monitoreo en tiempo real de las principales variables eléctricas de la micro-red. Se corrobora el funcionamiento del gestor de alarmas. Tal como se muestran en

los resultados, cuando la variable supera el valor umbral de la alarma, se imprime en consola el elemento monitoreado, el valor de sobrepaso y con el correspondiente nivel de alerta. Por otro lado, el historizador funciona correctamente durante el monitoreo activo. Es posible visualizar en un gráfico cómo ha sido el comportamiento de una variable cualquiera de los elementos monitoreados.

El aporte de este trabajo es significativo en cuanto a micro-redes se trata. A pesar de que en esta memoria no se implementa optimizador alguno de recursos y no se instauran políticas de operación, puede ser utilizado como base para desarrollar y testear nuevos algoritmos en un entorno real.

Finalmente, se menciona que el programa tiene varios apartados en donde podría ser mejorado, tanto en su parte de diseño y programación. El código puede ser comprimido, ya que, desde su inicio hasta su estado de desarrollo, se tenían concepciones distintas de como sería implementado. En su diseño, y a modo de ejemplo, se menciona que el formulario de configuración de elementos puede no ser la mejor forma de implementación para el usuario, ya que muestra demasiada información de forma instantánea.

5.1. Trabajo Futuro

Como trabajo futuro se proponen los siguientes puntos:

- Segregar el algoritmo de adquisición de datos, de manera que cada solicitud de datos de medición a los esclavos Modbus sea un proceso independiente del resto y del proceso principal que es la interfaz gráfica. Este punto apunta a que el programa mejore su escalabilidad, en donde el *trade-off* es aumentar el uso de CPU. Adicionalmente, el hecho de que este proceso actualmente sea serial tiene impactos sobre la eficiencia del programa cuando uno de los dispositivos de medida es desconectado, ya que provoca un *timeout* de 3 [s].
- Se propone desarrollar e implementar un estimador de estado de la red. De esta manera, se pueden obtener valores aproximados de los elementos sin medición en base a los que sí la tienen.
- Se propone desarrollar e implementar una política de transición de operación entre modo isla y conectada a la red.
- Implementación de un sistema EMS al programa SCADA, el cual logre optimizar los recursos de la red actuando como un control secundario.

Bibliografía

- [1] Ton, D. T. et Smith, MA (2012). The us department of energy's microgrid initiative. The Electricity Journal, 25(8), 84-94.
- [2] Nysesda Microgrid *Visitado el 2 de agosto de 2022*. <https://www.nysesda.ny.gov/All-Programs/NY-Prize>
- [3] A. Vasilakis, I. Zafeiratou, D. T. Lagos and N. D. Hatziargyriou, "The Evolution of Research in Microgrids Control," in IEEE Open Access Journal of Power and Energy, vol. 7, pp. 331-343, 2020, doi: 10.1109/OAJPE.2020.3030348.
- [4] Mina, J. D., Lozano, C. A., y Caicedo, E. F.. (2017). Una propuesta de integración de arquitecturas de generación descentralizada en ambientes de micro-redes. Entre Ciencia e Ingeniería, 11(22), 9-17. Accesed 06, 2023, http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1909-83672017000200009&lng=en&tlng=es
- [5] Y. Xu, C. -C. Liu, K. P. Schneider, F. K. Tuffner and D. T. Ton, "Microgrids for Service Restoration to Critical Load in a Resilient Distribution System," in IEEE Transactions on Smart Grid, vol. 9, no. 1, pp. 426-437, Jan. 2018, doi: 10.1109/TSG.2016.2591531.
- [6] "Modicon Modbus Protocol Reference Guide", June 1996
- [7] "MODBUS over Serial Line, Specification and Implementation Guide", December 2006
- [8] "MODBUS Messaging on TCP/IP Implementation Guide", October 2006
- [9] Clarke. G, Reynders, D (2004). Modern SCADA Protocols.
- [10] Falk. H (2018). IEC 61850 Demystified.
- [11] Pagina web de SCADA Data Gateway *Consultado el 4 de septiembre de 2022*. <https://www.trianglemicroworks.com/products/scada-data-gateway/>
- [12] Esquema de Tarifas Grupo de Usuarios DNP3 *Visitado el 5 de junio de 2022*. <https://www.dnp.org/Join-Purchase/Member-Fee-Guide>
- [13] "IEEE Recommended Practice for Data Communications Between Remote Terminal Units and Intelligent Electronic Devices in a Substation" in IEEE Std 1379-2000 , vol., no., pp.1-72, 16 March 2001, doi: 10.1109/IEEESTD.2001.92412.
- [14] Estadística de instalación de centrales solares y eólicas, Irena. *Visitado el 8 de junio de 2022*. <https://irena.org/Statistics/View-Data-by-Topic/Capacity-and-Generation/Statistics-Time-Series>
- [15] Pagina web de SIMATIC WinCC V7. *Visitado el 15 de mayo de 2022*. <https://new.siemens.com/global/en/products/automation/industry-software/automation-software/scada/simatic-wincc-v7.html>
- [16] Pagina web de Microgrid Control - a SICAM application. *Visitado el 16 de mayo de*

2022. <https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/microgrid/sicam-microgrid-controller.html>
- [17] Pagina web de eSCADA. *Visitado el 29 de abril de 2022.* <https://etap.com/packages/electrical-scada>
- [18] Pagina web de EcoStruxureTM Power Operation. *Visitado el 5 de mayo de 2022.* <https://www.se.com/ww/en/product-range/65405-ecostruxure-power-operation/>
- [19] Modulo de EcoStruxure, Advanced Reporting and Dashboards. *Visitado el 6 de mayo de 2022.* <https://www.se.com/us/en/product/PSA104112/ecostruxure-power-operation-advanced-reporting-and-dashboards-module/>
- [20] Pagina web de Energy Control Center. *Visitado el 6 de mayo de 2022.* <https://www.se.com/us/en/product-range/22570931-energy-control-center/>
- [21] Pagina web de VTScada. *Visitado el 8 de mayo de 2022.* <https://www.vtscada.com/what-is-vtscada/>
- [22] Listado de precios de la licencia de VTScada *Visitado el 9 de mayo de 2022.* https://www.vtscada.com/documents/pricing/VTScada_US_Pricing_Web.pdf
- [23] P. Leach, M. Mealling and R. Salz, “ A Universally Unique Identifier (UUID) URN Namespace”, July 2005. *Visitado el 23 de diciembre de 2022.* <https://www.rfc-editor.org/rfc/rfc4122>
- [24] Pagina web del programa Diagslave *Visitado el 18 de diciembre de 2022* <https://www.modbusdriver.com/diagslave.html>

Anexo

Codigo Base

A continuación se adjunta la dirección web del repositorio en donde se almacena el código de Micro-Grid SCADA para futuras referencias: <https://github.com/AGuerCar/Micro-Grid-Scada/>