



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

ESTUDIO E IMPLEMENTACIÓN DE PROTECCIONES ADAPTATIVAS PARA
MICRO-RED DE LABORATORIO ANTE CAMBIOS TOPOLÓGICOS DEFINIDOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

NICOLÁS ANTONIO SILVA MUÑOZ

PROFESOR GUÍA:
PATRICIO MENDOZA ARAYA

MIEMBROS DE LA COMISIÓN:
PABLO MEDINA COFRÉ
PABLO GONZÁLEZ INOSTROZA

SANTIAGO DE CHILE
2020

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: NICOLÁS ANTONIO SILVA MUÑOZ
FECHA: 2020
PROF. GUÍA: PATRICIO MENDOZA ARAYA

ESTUDIO E IMPLEMENTACIÓN DE PROTECCIONES ADAPTATIVAS PARA MICRO-RED DE LABORATORIO ANTE CAMBIOS TOPOLÓGICOS DEFINIDOS

Las micro-redes han tenido un aumento en su utilización, debido a que representan una solución económica, que permite la incorporación de las energías renovables y que es capaz de abastecer de energía eléctrica a sectores remotos como islas o zonas rurales. Sin embargo, grandes desafíos se han planteado entorno a ellas en materias de estabilidad, economía y protecciones. Es precisamente en este último tópico en el cual se enfoca el presente trabajo de título.

Una protección es un dispositivo destinado a desenergizar un sistema, circuito o artefacto cuando en ellos se alteran las condiciones normales de funcionamiento. Es necesario generar estrategias y esquemas de protección que se ajusten a los requerimientos de cada tipo de sistema a proteger. Tal es el caso de las micro-redes, las cuales no suelen utilizar protecciones convencionales de grandes SEP debido al bajo aporte de corrientes de falla por parte de los recursos distribuidos, la variabilidad del recurso renovable, los flujos bidireccionales y el constante cambio topológico que experimentan estos sistemas de baja o media tensión.

En base a lo expresado en los párrafos anteriores, se plantea como objetivo el estudio y desarrollo de una estrategia de protección adaptativa sobre la micro-red presente en el Laboratorio de energía y accionamientos de la Universidad de Chile. Este laboratorio fue seleccionado dada la versatilidad que ofrece, pues permite experimentar con técnicas como la estrategia adaptativa dentro de un ambiente seguro y controlado.

En cuanto al sistema de protección propuesto, éste incorpora tres dispositivos de lectura y/o protección integrados a un servidor común llamado OPC. Mediante un programa computacional basado en Python, se ejecuta una solicitud de mediciones al servidor OPC y, en base a ellas, se identifica la topología imperante en la micro-red, para luego seleccionar una configuración que garantice la seguridad y protección del sistema. Dicha selección se realiza mediante *Lookup Table*.

Por un lado, el presente trabajo incorpora nuevas herramientas de operación a la micro-red como botoneras, almacenamiento de datos en SQL, configuración continua de curvas de protección en un relé utilizado y la selección discreta de uno de seis *Setting Group* disponibles. Por otro lado, se introduce una metodología y un avance en la configuración de los sistemas de protección y/o medición, pudiendo ser un ejercicio replicable a futuro para otros dispositivos, abriendo nuevas posibilidades de investigación. Un ejemplo de ello sería realizar una optimización entera-mixta para seleccionar la mejor configuración de las protecciones o agregar nuevos elementos que compartan protocolo MODBUS o serial.

Finalmente, se comprueba la operación y el correcto funcionamiento de las herramientas y procedimientos realizados. Lo anterior se realiza mediante la emulación de dos fallas consecutivas, producidas con el arranque de un motor de inducción, sobre dos barras de la micro-red. Se observa la acción del sistema de protección ante dichas fallas mediante un *Event Recorder* propio del relé utilizado, comprobando una adaptación de las protecciones en 14[s] para la primera falla y 16[s] en el caso de la segunda contingencia.

Agradecimientos

Quisiera agradecer, en primer lugar, a mi familia. Sin lugar a dudas siempre fueron y son mi refugio, un hogar donde encontré paz y contención en los momentos de mayor dificultad. A ustedes les agradezco y dedico todos mis logros.

A mis padres, Jorge y Verónica, que han estado presentes y preocupados por mi formación educativa desde el primer día. Por enseñarme el verdadero significado del esfuerzo, de la honestidad y la mayoría de los valores que hoy llevo orgulloso en mi persona. Definitivamente me encuentro muy feliz por su hermosa labor de padres, criando y educando a sus 3 hijos. Gracias.

A mis hermanos, los cuales son el motor de mis sueños. Por darme la motivación, la compañía y todo el apañe posible. Gracias por siempre estar dispuestos a la ayuda, a una conversación y una palabra amistosa cuando más se necesita.

A mi polola, Dani, chinita, gracias por el apañe, comprensión y todos los consejos. Este trabajo de título y año estudiantil no hubiese sido posible sin su compañía y comprensión. Gracias por la palabra de ánimo, por la palabra exacta y el abrazo perfecto.

Agradecer a mis grandes amigos del DIE: Moraga, Diland, King Zúñiga, Fesa y todos los GilesTeam. Gracias por la amistad, por los buenos momentos. Por tantas horas de bromas mientras estudiábamos o "estudiábamos". Gracias por recibirme en sus casas sin problemas cuando necesitaba pasar noches cerca de la U para cumplir con algún proyecto. Sin duda fueron lo mejor de la universidad y me ayudaron mucho con mi estadía en la facultad. Espero que la amistad perdure y nos reencontremos en algún futuro profesional.

A cada profesor que ha aportado en mi formación y en este trabajo de título. Alex Urrutia, por ser un buen guía y amigo durante las primeras etapas del proyecto. De forma especial, a Pablo González por siempre estar atento y dispuesto a ayudarme con cualquier problema del trabajo, gracias por el ánimo y la preocupación. Al profesor Mendoza, por guiar el trabajo, confiar en mis capacidades para llevarlo a cabo y recibir cualquier consulta de mi parte mediante correos o presencialmente.

Finalmente, agradecer a General Electric, específicamente a Icaro de Moura, de la sede latinoamericana. Por su ayuda técnica respecto al dispositivo de protección utilizado en el trabajo.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	3
1.2.1. Objetivo General	3
1.2.2. Objetivos específicos	3
1.3. Alcances	3
1.4. Estructura de trabajo	4
2. Marco Teórico	6
2.1. Micro-red	6
2.1.1. Operación conectada a la red principal (On-Grid)	7
2.1.2. Operación aislada de la red principal (Off-Grid)	9
2.2. Protecciones	11
2.2.1. Sistemas de protección	11
2.2.2. Tipos de sistemas protección	12
2.2.3. Coordinación de protecciones	16
2.2.4. Protección adaptativa	17
2.3. Laboratorio de Energía y Accionamientos	19
2.3.1. Topología de Micro-red	19
2.4. Estado del Arte y experiencias anteriores	21
2.4.1. Trabajos anteriores	21
2.4.2. Experiencia Internacional	22
3. Metodología	24
3.1. Estrategia Metodológica	24
3.2. Trabajo Previo a Laboratorio	26
3.2.1. Definición de Objetivos, Alcances y Revisión Bibliográfica	26
3.2.2. Recopilación de datos de elementos de laboratorio	26
4. Implementación en laboratorio	27
4.1. Captura de datos a OPC cliente	27
4.1.1. Visualización de Datos en OPC Cliente	30
4.2. Enlace de datos desde OPC a Python	32
4.3. Almacenamiento de datos SQLite	33
4.4. Lógica y selección de instrucciones mediante Look Up Tables	36
4.5. Configuraciones sobre equipos de protección	37

4.5.1.	Establecimiento de comunicación	38
4.5.2.	Configuración de funciones lógicas	39
4.5.3.	Acciones asociadas a salidas virtuales	40
4.6.	Envío y ejecución de instrucciones	41
4.7.	Selección y preparación de pruebas	43
4.7.1.	Prueba 1: Identificación de topología y selección de Setting Group . .	44
4.7.2.	Prueba 2: Emulación de falla mediante motor de inducción. Caso: Micro-red conectada a red principal de energía	45
5.	Resultados	49
5.1.	Experiencia de activación de <i>Setting Group</i> ante cambio topológico	49
5.1.1.	Caso 1: Sin almacenamiento de mediciones en SQL.	49
5.1.2.	Caso 2: Con almacenamiento de mediciones en SQL.	52
5.2.	Experiencia de falla y protección adaptativa. Caso conectado a la red principal de energía	54
5.2.1.	Primera falla, barra 5	54
5.2.2.	Segunda Falla, barra 3	56
6.	Conclusiones	59
	Bibliografía	62
7.	Anexos	63
7.1.	Direcciones MODBUS utilizados	63
7.1.1.	Multiple Feeder System F35 -UR Series	63
7.1.2.	Power Logic ION8600	65
7.1.3.	Power Logic Power Meter PM750	66
7.2.	Código Fuente de Python	67
7.3.	Código de SQLite	77

Índice de Tablas

2.1. Ventajas y desventajas de sistemas de protección	15
2.2. Parámetros red de laboratorio	20
2.3. Datos motor de inducción trifásico utilizado en pruebas y simulación	20
2.4. Valores base para micro-red	21
4.1. Direcciones IP de dispositivos de medición.	28
4.2. Selección de Setting Group (SG) operando de acuerdo a cada combinación topológica posible (Sin considerar anillos)	37
4.3. Valores configurados sobre curvas TOC para primera falla de motor de inducción (Barra 5)	47
4.4. Valores configurados sobre curvas TOC para segunda falla de motor de inducción (Barra 3)	48
5.1. Tiempos de respuesta ante cambio topológico y selección de Setting Group sin mediciones	51
5.2. Tiempos de respuesta ante cambio topológico y selección de Setting Group con mediciones	53
7.1. Resumen de direcciones MODBUS comunes para Multilin F35	63
7.2. Resumen de direcciones MODBUS utilizadas en Multilin F35 por fuente	64
7.3. Resumen de direcciones MODBUS utilizadas para ION8600	65
7.4. Resumen de direcciones MODBUS utilizadas para dispositivos PM750	66

Índice de Ilustraciones

2.1. Modelo general de una Micro-red	7
2.2. Modelo On-Grid con retiro de potencia	8
2.3. Modelo On-Grid con inyección de potencia	8
2.4. Modelo Off-Grid con autosuficiencia de potencia	10
2.5. Esquema con interruptor automático de selección	11
2.6. Ejemplo de curva de tiempo inverso	12
2.7. Curva típica de protección por baja tensión	14
2.8. Resumen de características de corriente de falla	16
2.9. Esquema de coordinación básico mediante retardo de tiempo en curvas de sobre-corriente	17
2.10. Condiciones de micro-red aptas para protección adaptativa	18
2.11. Topología Micro-red presente en laboratorio	19
3.1. Diagrama de Flujo de la estrategia metodológica utilizada	25
4.1. Pantalla de inicio a INAT TCP/IP-H1 OPC Server	28
4.2. Ventana de configuración de dispositivos	28
4.3. Ventana avanzada de configuración de dispositivos	28
4.4. Pantalla Path Access de INAT TCP/IP-H1 OPC Server con todos los dispositivos	29
4.5. Representación de ingreso de direcciones MODBUS a Servidor OPC	30
4.6. Menú para inscripción de etiquetas a visualizar en OPC cliente	31
4.7. Visualización OPC Test Client	31
4.8. Visualización de datos almacenados mediante Db browser	35
4.9. Configuración para establecer comunicación con F35	38
4.10. Bloque lógico 1. Activación de salida virtual por curva IOC	39
4.11. Bloque lógico 2. Activación de salida virtual por curvas TOC y UV	40
4.12. Bloque lógico 3. Cambio de Setting Group por botón o input digital	40
4.13. Ejecución de instrucciones para salidas virtuales V04 hasta V06	41
4.14. Ejecución de instrucciones para salidas virtuales VO7 hasta V011	41
4.15. Interfaz de usuario GUI	42
4.16. Diagrama de Flujo de prueba 1	44
4.17. Diagrama de Flujo de prueba 2	46
4.18. Simulación de corriente de encendido motor de inducción en conexión Estrella	47
4.19. Simulación de corriente de encendido motor de inducción en conexión Delta .	47
4.20. Diagrama resumen de sucesos observables durante prueba 2	48
5.1. Event Recorder de operación sin almacenamiento de mediciones en SQLite .	50

5.2.	Event Recorder de operación con almacenamiento de mediciones en SQLite .	52
5.3.	Reporte de primera falla con motor de inducción (Barra 5)	54
5.4.	Oscilografía de primera falla con motor de inducción (Barra 5)	55
5.5.	Event Recorder de acciones realizadas por protección ante primera falla . . .	56
5.6.	Reporte de segunda falla con motor de inducción (Barra 3)	57
5.7.	Event Recorder de acciones realizadas por protección ante segunda falla . . .	57

Código Fuente

4.1.	Establecimiento de comunicación con OPC	32
4.2.	Ejemplo de lectura de datos desde OPC a Python	32
4.3.	Creación de archivos y tablas SQL	33
4.4.	Ejemplo de almacenamiento de datos en SQLite para mediciones del PM750-2	34
4.5.	Código ejemplo de envío de instrucciones Python a OPC servidor	42
4.6.	Módulo para configuración de curva de tiempo definido por sobrecorriente	43
7.1.	Programa principal	67
7.2.	Programa de almacenamiento en SQL	77

Capítulo 1

Introducción

1.1. Motivación

Con el transcurso del tiempo, la utilización de micro-redes ha ido en aumento debido a una amplia gama de ventajas que estas incorporan. Por un lado, corresponden a una solución más económica, pues no existen grandes pérdidas por transmisión al tener generación cerca de los consumos [1]. Por otro lado, permiten la autosuficiencia de energía eléctrica en comunidades aisladas o de difícil acceso a la red de distribución principal (zonas rurales).

A pesar de lo expuesto en el párrafo anterior, las micro-redes conllevan una serie de problemas y desafíos que aún son motivos de investigación. Esto se debe a que el diseño de una micro-red se encuentra en constante expansión y cambios topológicos durante su operación. De acuerdo con [2], algunos de estos desafíos actuales son:

- **Confiabilidad y estabilidad:** La interacción entre la micro-red y la red principal (red de distribución) podría afectar la confiabilidad y estabilidad de esta última.
- **Programación y diseño:** Es necesario considerar esta materia como en constante progreso, debido a la interacción de gran cantidad de elementos. Debe garantizarse la suficiencia, la estructura de la micro-red, la interconexión de redes, entre otros bajo constante búsqueda de la optimización.
- **Control:** La diversidad de fuentes y elementos, la intermitencia y aleatoriedad de las fuentes primarias de energía (ERNC), dificultan el control de tensión y frecuencia en el sistema. Esto, sumado a la interacción entre micro-redes y la red de distribución hace necesario esquemas de control más complejos y eficientes.
- **Protecciones:** Debido a que existen flujos bidireccionales que dificulta las mediciones y esquemas de protección. Además es necesario contar con un sistema que proteja a la micro-red y a la red de distribución cuando una falla ocurre, de manera rápida y selectiva. Adicionalmente, las variadas tecnologías de generación distribuida pueden incorporar distintos esquemas de protección y criterios, por lo que es necesario coordinarlos eficientemente.

- **Planificación de la Operación:** Se requiere gestionar y planificar la operación óptima de la micro-red cumpliendo los requerimientos de calidad de suministro, suficiencia y considerando la variabilidad de la generación renovable.

El presente trabajo de título se encuentra orientado al tópico de protecciones, dada su importancia en el desarrollo de los demás desafíos nombrados anteriormente. Es importante desarrollar los sistemas de protecciones en las micro-redes para permitir la operación y experimentación segura sobre ellas, facilitando el avance de la investigación sobre este tipo de sistemas de energía cada día más utilizados.

El tema de protecciones en micro-redes se ha desarrollado de forma particular debido a condiciones propias de la corriente en estos sistemas y que difieren de los SEP convencionales. Estas condiciones, particulares en este tipo de sistemas, se deben principalmente a que una micro-red posee [3]:

- Magnitudes eléctricas similares dentro de ella, dado sus bajos niveles de consumo y cercanía de estos a las fuentes de generación.
- Flujos de potencia bidireccionales, debido a la Generación Distribuida.
- Diversos puntos de conexión de recursos distribuidos, los cuales influyen en el punto de operación normal del sistema. Esto se debe a cambios topológicos frecuentes debido a la conexión/desconexión de generadores, sistema de almacenamientos y cargas.
- La incorporación de tecnologías basadas en energías renovables, las cuales poseen una variabilidad asociada a sus fuentes primarias de producción que puede afectar la calidad de suministro eléctrico y su despacho. Lo anterior no es menor, pues una micro-red permite la penetración de este tipo de energías, pudiendo tener una alta correlación la variabilidad de la generación con su operación normal.
- Corrientes de sistema limitadas, debido a la amplia presencia de equipos de electrónica de potencia.
- Dos posibles modos de operación (conectado o aislado de la red).

Lo anterior se traduce en que existe una amplia gama de puntos de operación que deben ser considerados para la correcta planificación y dimensionamiento de un sistema de protecciones. Además, es necesario considerar los bajos niveles de corriente de falla, los cuales al ser similares a las magnitudes nominales pueden ocasionar problemas de confiabilidad y seguridad del sistema al activar protecciones en situaciones de operación normal, o no activándolos cuando una contingencia ocurre. Bajo el contexto expuesto, nace la necesidad de desarrollar sistemas de protecciones adaptativas, las cuales permitirían realizar una fácil adecuación para enfrentar los cambios operacionales asociados a una micro-red.

Con estas nociones presentes, se propone desarrollar un sistema de protecciones adaptativo sobre la micro-red presente en el Laboratorio de Energía y Accionamientos, perteneciente al Departamento de Ingeniería Eléctrica de la Universidad de Chile. Lo anterior permitirá emular dentro de un ambiente controlado distintas topologías existentes de micro-redes, con el fin de realizar pruebas de operación e incluso ser una futura experiencia docente.

El aporte de esta memoria está en complementar el tema de protecciones en la micro-red del laboratorio mencionado, permitiendo realizar futuras investigaciones sobre ella de for-

ma segura y controlada. Profundizando este punto, el presente trabajo de título pretende introducir las primeras herramientas del control adaptativo en la micro-red del laboratorio, realizando la coordinación OPC-Python, comunicación entre dispositivos (centralizando la operación de la micro-red), la ejecución de acciones e instrucciones sobre el relé Multilin F35, entre otros.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar e implementar un sistema de protecciones adaptativas en la micro-red del laboratorio de energía y accionamientos del Departamento de Ingeniería Eléctrica de la Universidad de Chile. Dicho sistema de protección adaptativa deberá cumplir y preparar el sistema para situaciones de falla, tanto en operación aislada como en operación conectada a la red, incorporando recursos distribuidos presentes en el laboratorio.

La finalidad es la implementación de un sistema seguro y confiable para futuras investigaciones desarrolladas sobre la micro-red del laboratorio, incluyendo la posibilidad de replicar sistemas existentes sobre un ambiente controlado.

1.2.2. Objetivos específicos

1. Desarrollar, proponer e implementar un esquema de protecciones adaptativas sobre la micro-red del laboratorio que sea capaz de enfrentar fallas en diversas configuraciones del sistema y puntos de operación.
2. Obtención de una correcta configuración de los dispositivos de protección presentes en el laboratorio mediante la utilización de un servidor y cliente OPC, y programas basados en Python. En particular, el relé *Multilin F35 Feeder Protection System* emerge como principal protagonista a utilizar.
3. Generar un programa computacional, basado en Python, capaz de realizar la captura de datos, lectura de parámetros y envío de instrucciones a la micro-red. Dicho programa debe incorporar una interfaz de usuario para visualizar y operar la micro-red de manera intuitiva y directa.
4. Crear una metodología con el procedimiento realizado, con el propósito de ser una ayuda para la configuración del sistema de protección ante futuras expansiones o adaptaciones de la micro-red del laboratorio.

1.3. Alcances

El presente trabajo de título tiene por finalidad implementar y configurar un conjunto de protecciones eléctricas de la micro-red existente en el Laboratorio de Energía y Accionamientos del departamento. A través de esta implementación, se pretende garantizar el funcionamiento confiable y seguro de la micro-red, permitiendo la incorporación de los recursos distribuidos

disponibles en el laboratorio. Lo expresado anteriormente otorga la posibilidad de ocupar la micro-red para replicar modelos existentes alrededor del mundo, ofrecer experiencias docentes e incluso expandir la micro-red de forma controlada y segura.

Es importante destacar que los valores configurados responden a una topología particular y orientadas a pruebas específicas, por lo que ante cambios topológicos como incorporación de nuevas barras, sistemas de generación distribuida o cargas deben realizarse los ajustes pertinentes. Este trabajo de título pretende presentar las herramientas para realizar dicha configuración de manera intuitiva y directa, sin embargo, no corresponde a los alcances de la memoria ofrecer las configuraciones correspondientes para cada caso de estudio.

A través de la sección de metodología del presente informe, se explicitará la instrucción para repetir de forma correcta y rápida del proceso de adaptación de las protecciones. Lo anterior con el objetivo de dotar al departamento las facilidades para expandir o cambiar la configuración interna de la micro-red (incorporando nuevas barras, líneas de transmisión, consumos o fuentes de generación).

1.4. Estructura de trabajo

El presente trabajo se divide en 7 capítulos, los cuales son detallados a continuación:

El primer capítulo corresponde a la introducción, la cual se descompone en una motivación que presenta el problema, los objetivos, alcances del trabajo de título y la estructura del presente informe.

El segundo capítulo contiene el marco teórico, el cual tiene por finalidad ser una base conceptual para el lector y facilitar el entendimiento de este trabajo de título.

En el tercer capítulo se presenta la metodología de trabajo, en ella se describen las etapas que el autor deberá realizar para alcanzar los objetivos y alcances planteados a comienzo del trabajo.

El capítulo cuarto presenta la implementación y desarrollo de la metodología planteada en el capítulo anterior. En este capítulo se incorpora la explicación esencial del programa creado, además de incluir la configuración del relé F35 y los supuestos utilizados durante el trabajo de título. Por último, se exponen los preparativos y condiciones de prueba con el fin de constatar el cumplimiento de los objetivos y alcances propuestos.

Dentro del quinto capítulo, se exponen los resultados y primeros análisis respecto a las experiencias desarrolladas. Se espera que los resultados obtenidos tengan relación con la teoría y permitan alcanzar los objetivos planteados.

Las conclusiones del trabajo de memoria se encuentran presentes en el sexto capítulo, incluyendo tareas pendientes, trabajo futuros y desafíos encontrados durante el trabajo.

Posterior al sexto capítulo, se presenta la bibliografía utilizada durante todo el trabajo de tí-

tulo, incluyendo el sustento para un marco teórico, la revisión de experiencias internacionales, revisión de estado del arte e investigación sobre condiciones de la micro-red del laboratorio.

Finalmente, el séptimo capítulo corresponde a los Anexos. Incorpora toda la información adicional y específica utilizada durante este trabajo. Cabe destacar que en esta sección se encuentra disponible el código computacional completo utilizado durante la experiencia.

Capítulo 2

Marco Teórico

La presente sección del informe pretende otorgar una base conceptual acerca de las definiciones, descripciones y teoría necesaria para facilitar el entendimiento del trabajo realizado. En términos generales, el capítulo actual está participando en 4 ejes principales:

1. Definiciones principales sobre Micro-redes
2. Sistemas de protección y su coordinación para Micro-redes
3. Componentes de laboratorio
4. Estado del arte y trabajos anteriores

2.1. Micro-red

Como se indica en [4] y [1], una micro-red es un conjunto de pequeñas fuentes de generación, sistemas de almacenamiento y cargas controlables operando integradamente en baja tensión. Estos sistemas están marcados por tener sus centros de generación cerca de los consumos, reduciendo las pérdidas por transmisión. Además, una micro-red es capaz de funcionar tanto en corriente continua (DC) como en corriente alterna (AC), por lo que equipos de electrónica de potencia, inversores y controladores son normalmente utilizados para interconectar ambos tipos de red.

Las energías renovables están en auge en las micro-redes, dado sus bajos niveles de carga y fácil penetración. En este contexto, la energía solar, eólica e hidráulica son las fuentes de energía renovable más utilizadas [1][5].

Basado en [1] y [6], las micro-redes presentan las siguientes ventajas sobre los SEP convencionales:

- Aumentar la eficiencia del sistema
- Minimizar el consumo de energía total
- Reducir la cantidad de emisiones de gases de efecto invernadero y contaminantes varios
- Mejorar la calidad de servicio y confiabilidad del sistema

- Poseer un costo eficiente de reemplazo de la infraestructura
- Poseer gran robustez frente a fuertes eventos climáticos
- Participar en el mercado eléctrico cuando la micro-red se encuentra conectada a la red principal de energía

Por otro lado, una micro-red puede funcionar en modo aislado o en un modo de funcionamiento paralelo a la red principal de energía. Esto radica en la capacidad de la micro-red para suplir demandas en lugares remotos para los cuales una red principal no es capaz de entregar el suministro (Off-Grid), o la opción de participar en el mercado eléctrico inyectando o consumiendo energía desde el sistema principal (On-Grid)[4]. A continuación, se presenta en la Figura 2.1 un esquema de micro-red usual.



Figura 2.1: Modelo general de una Micro-red

Es importante destacar que el interruptor que separa la red principal de la micro-red permite seleccionar el modo de operación de esta. Esta interconexión se realiza en el punto de acoplamiento común o PCC (Por su sigla en inglés "Point of Common Couplin"). A continuación se presenta una explicación general acerca de los modos de operación On-Grid u Off-Grid.

2.1.1. Operación conectada a la red principal (On-Grid)

La operación conectada a la red principal se realiza a través del PCC, permitiendo el intercambio de potencia entre ambos sistemas. Dentro de este contexto existen 3 situaciones posibles dependiendo del balance entre generación y demanda:

1. **Balance perfecto entre demanda y generación:** En este caso, los elementos distribuidos de generación logran suplir la demanda de forma eficiente y justa. Cuando esto sucede, la corriente a través del PCC es igual a 0.
2. **Nivel de consumo superior a la generación:** Ocurre cuando el nivel de consumo es mayor a la generación producida por los recursos distribuidos, es decir, el flujo de potencia activa y reactiva proveniente desde la red es distinta de cero ($\Delta P \neq 0$ y

$\Delta Q \neq 0$). Lo anterior implica que existe una corriente en el PCC, proveniente del sistema eléctrico principal, que solventa el déficit de generación. A continuación se presenta en la Figura 2.2 la situación descrita anteriormente a través de un ejemplo de una micro-red construida en Brasil [7].

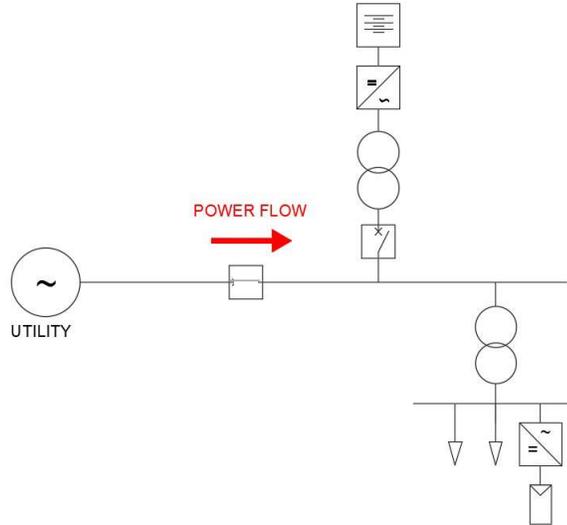


Figura 2.2: Modelo On-Grid con retiro de potencia

3. **Nivel de generación superior a consumo:** La presente situación es contraria a la expuesta en el punto anterior. Se tiene que el exceso de producción es evacuada hacia la red principal de energía hasta producir el equilibrio de operación.

La Figura 2.3, perteneciente a la micro-red de Brasil, ejemplifica esta situación.

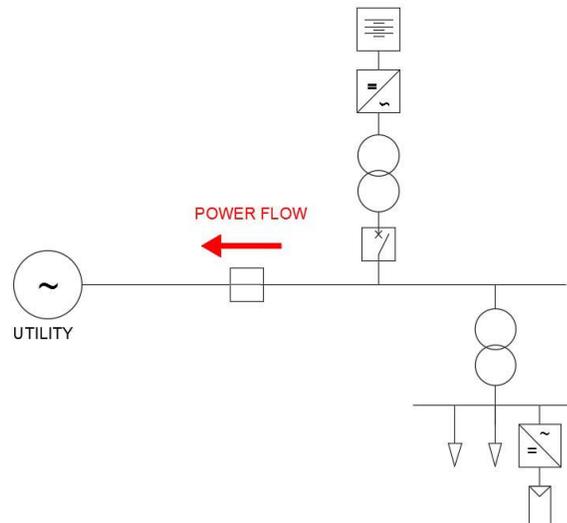


Figura 2.3: Modelo On-Grid con inyección de potencia

Es importante notar que los flujos de potencia son tratados de forma independiente para la potencia activa P y reactiva Q . Considerando las 3 situaciones antes descritas, existen las siguientes 5 combinaciones posibles:

- a) Inyección de potencia activa P y reactiva Q hacia red principal.
- b) Inyección de potencia activa P y retiro de potencia reactiva Q .
- c) Retiro de potencia activa P e inyección de potencia reactiva Q .
- d) Retiro de potencia activa P y reactiva Q desde la red principal de energía.
- e) Suficiencia perfecta de energía con generación propia, es decir, sin la participación de la red principal de energía (Corriente por PCC igual a cero).

Por otro lado, este modo de operación conlleva a que la micro-red participe en el mercado eléctrico [6], pudiendo ser modelado como una barra que inyecta o retira energía desde el sistema eléctrico principal.

La red principal de energía, al ser un sistema más grande y robusto, impone parámetros eléctricos como la frecuencia y el nivel de tensión sobre la micro-red, además de aportar estabilidad al sistema ante perturbaciones internas. Sin embargo, ante ciertas situaciones de falla ,internas o externas a la micro-red, el PCC es desconectado para proteger los elementos y mantener la calidad de suministro. Cuando esto ocurre, el modo aislado es activado y la micro-red debe solventar los problemas de estabilidad y protección de forma autónoma.

2.1.2. Operación aislada de la red principal (Off-Grid)

Una red aislada se encuentra desconectada de la red principal de energía. Esto puede ocurrir tanto de forma temporal como de forma permanente. Este último caso ocurre en zonas rurales, islas o cualquier localidad con dificultades para abastecer su consumo desde la red principal de electricidad.

Continuando con la red de Brasil [7], se presenta en la Figura 2.4 la situación del caso Off-Grid, notando un flujo de potencia proveniente de la generación distribuida de la micro-red.

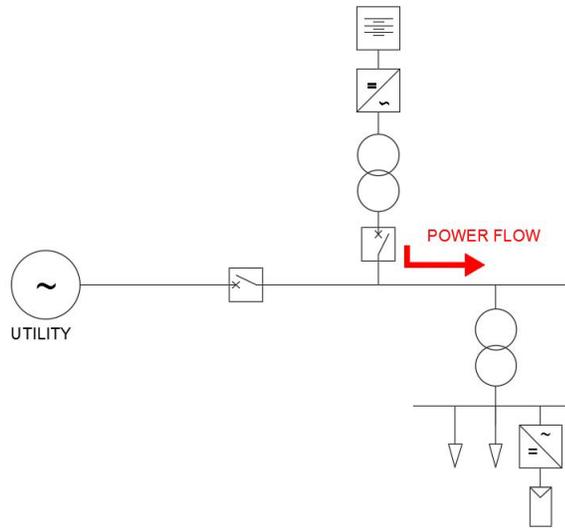


Figura 2.4: Modelo Off-Grid con autosuficiencia de potencia

De acuerdo con [8], la capacidad de la micro-red para funcionar en modo isla, además del modo conectado a la red principal de energía, aumenta la confiabilidad y suficiencia del suministro eléctrico.

Cuando el sistema se encuentra operando de forma aislada, existen menos unidades de generación que aportan corrientes de falla. Además, junto con la integración de las Energías Renovables No Convencionales (ERNC en adelante), la electrónica de potencia que limita la corriente de cortocircuito es muy abundante en este tipo de sistemas. Lo anterior se traduce en que las corrientes de cortocircuito en micro-redes son similares a las corrientes nominales de operación del sistema, dificultando la utilización de dispositivos y esquemas de protección convencionales [4][9].

En la figura 2.5, mostrada a continuación, se presenta un esquema donde el interruptor inteligente es aquel destinado para que la micro-red funcione en modo aislado (interruptor abierto) o conectado a la red principal de energía.

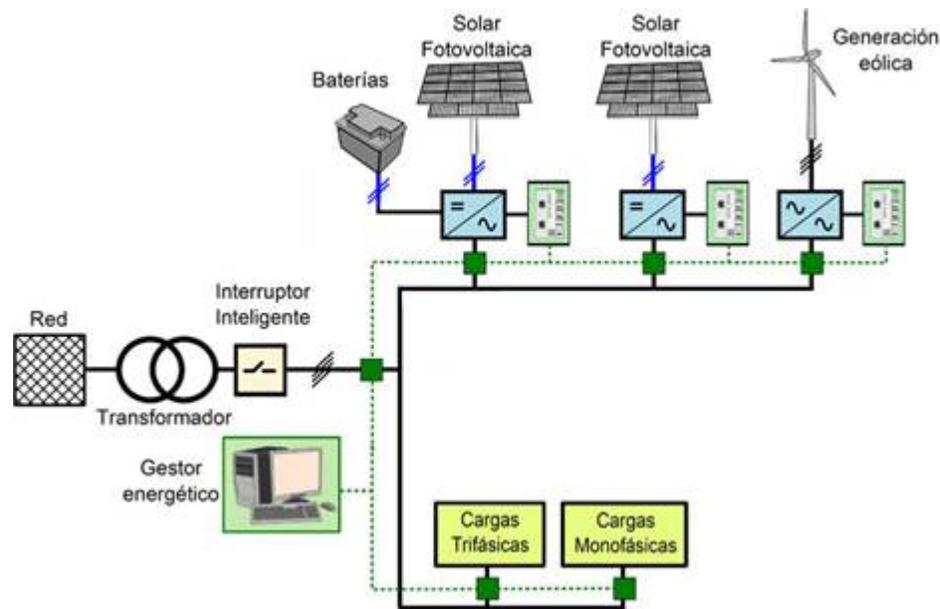


Figura 2.5: Esquema con interruptor automático de selección

2.2. Protecciones

2.2.1. Sistemas de protección

De acuerdo con la Superintendencia de Electricidad y Combustible (SEC), una protección es un dispositivo destinado a desenergizar un sistema, circuito o artefacto cuando en ellos se alteran las condiciones normales de funcionamiento [10].

Los dispositivos de protección poseen una serie de requerimientos y condiciones necesarios para ser considerados como tal, entre ellos se encuentra:

- **Sensibilidad:** Los dispositivos deben ser capaces de detectar de buena forma cuando existen situaciones anormales en la operación de la red. En este contexto se encuentran los conceptos de seguridad, confiabilidad y fiabilidad.
 1. **Sistema seguro:** Esto ocurre si los componentes no actúan cuando no deben hacerlo, es decir, no detectan falsos positivos que activen las desconexiones en operación normal.
 2. **Sistema confiable:** Situación donde los dispositivos de protección actúan cuando deben hacerlo y de forma efectiva.
 3. **Sistema fiable:** Este término corresponde a la situación en que el sistema es seguro y confiable a la vez.
- **Selectividad:** Los artefactos de protección deben ser capaces de aislar la menor cantidad de circuito posible. Lo anterior tiene el objetivo de minimizar las consecuencias a otros sectores de la red.
- **Rapidez:** El sistema debe operar en el menor tiempo posible para proteger la integridad de otros dispositivos y artefactos.

- **Flexibilidad:** El sistema de seguridad tiene que adaptarse ante cambios de la operación del sistema.

A continuación se expone el funcionamiento de los dispositivos de protección más comúnmente utilizados en la industria.

2.2.2. Tipos de sistemas protección

La gran diversidad en tipos de sistemas inducen a variadas estrategias utilizadas para garantizar la operación segura de la red. En la presente sección, se estudiarán 4 tipos de esquemas de protección basado en [11] y [12]:

1. **Dispositivos de sobrecorriente:** Se basa en la medición constante de corriente a través de un circuito. Cuando ocurre una falla, la magnitud de la corriente aumenta superando un umbral fijado sobre el dispositivo. Cuando dicho umbral, denominado corriente de *Pick-up*, es superado, la protección se activa de forma instantánea (curva de tiempo definido) o con un tiempo inversamente proporcional a la magnitud de la falla (curva de tiempo inverso (Ver figura 2.6)).

Este tipo de protección es muy utilizado en la industria dado su fácil entendimiento y gran poder de selectividad. Con una correcta configuración de las curvas de tiempo inverso del artefacto, se puede operar de forma selectiva e incluso coordinada.

Se presenta en la figura 2.6 una curva de tiempo inverso típica para el caso de protección por sobrecorriente. En ella se aprecia que a mayores corrientes de falla el tiempo de operación del sistema de protección es menor. Adicionalmente, es posible notar la corriente de pick-up, la cual define el umbral en que la protección comienza a actuar.

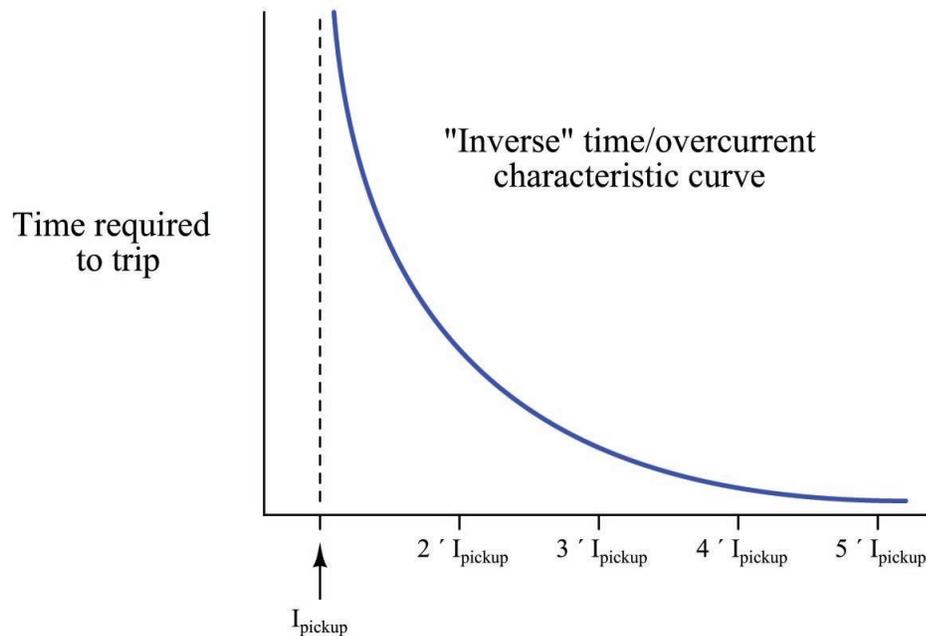


Figura 2.6: Ejemplo de curva de tiempo inverso

La protección de sobrecorriente presente en el dispositivo del laboratorio (Multilin F35), se encuentra determinada por 16 curvas seleccionables reunidas en 4 tipos:

- IEEE
- IEC
- GE Type IAC
- Otras Ajustables

En este caso en particular, se utilizarán las curvas IEC debido a que son las utilizadas en la industria y en trabajos anteriores realizados sobre la micro-red [13]. Dichas curvas IEC se encuentran determinadas por la ecuación 2.1, que establece una curva tipo inversa como la expuesta en la Figura 2.6.

$$T = TDM * \frac{K}{\left(\frac{I}{I_{pickup}}\right)^E - 1} \quad (2.1)$$

Donde:

T[s]: Corresponde al tiempo en segundos en que se activa la protección por sobrecorriente.

TDM[s]: De su sigla en inglés "Time Delay Multiplier". Corresponde a un ponderador de tiempo de la curva. Equivale a desplazar la curva verticalmente para generar un *Delay* luego de superado la corriente umbral (Tiempo inverso).

I: Medición de corriente en tiempo real del dispositivo, medido en Amperes.

I_{pickup} : Valor de corriente umbral definido en Amperes . A partir de ella, la curva de protección comienza a actuar.

K,E: Parámetros definidos por el tipo de curva seleccionado. Simulan de manera digital el comportamiento termomagnético de la protección.

2. **Dispositivos basados en voltaje:** Poseen un funcionamiento similar a los sistemas de sobrecorriente. Los sistemas basados en voltaje utilizan mediciones periódicas de voltaje para comparar con una tensión de *Pick-Up*. Cuando una falla tiene lugar, existen grandes pérdidas que se traducen en caídas intempestivas de voltaje que superan el umbral fijado en las protecciones.

Al igual que en el caso de sobrecorriente, la configuración de los dispositivos se realiza mediante curvas de tiempo definido o tiempo inverso, lo que implica gran poder selectivo y de coordinación.

En la figura 2.7, se expone una curva de protección por baja tensión. Es importante destacar el valor de voltaje Pick-up (línea vertical color celeste) que posiciona el umbral en que la protección comienza a actuar. .

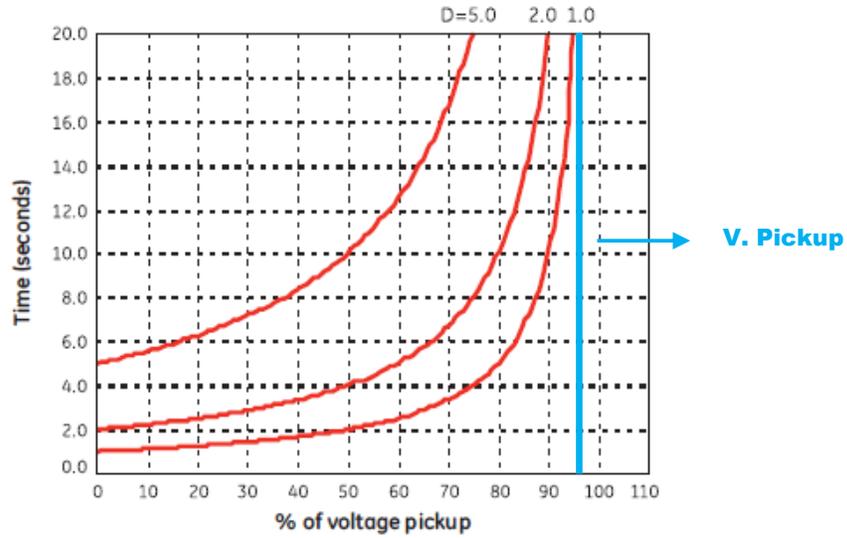


Figura 2.7: Curva típica de protección por baja tensión

En el caso particular de este trabajo, se utilizará una función por curva de baja tensión (Como el expuesto en la figura 2.7). Mientras mayor sea la desviación de voltaje respecto al valor umbral, el tiempo de actuación es menor. Este comportamiento está descrito por la ecuación 2.2.

$$T = \frac{D}{1 - \frac{V}{V_{pickup}}} \quad (2.2)$$

Donde:

T [s]: Corresponde al tiempo de activación, medido en segundos, de la protección por baja tensión.

D [s]: Corresponde a un ponderador de tiempo de la curva. Equivale a desplazar la curva verticalmente para generar un *Delay* luego de superado el voltaje umbral (D=0 corresponde a activación instantánea).

V: Medición de voltaje en tiempo real del dispositivo.

V_{pickup} : Valor de voltaje umbral definido. A partir de su valor, la curva de protección comienza a actuar.

3. **Dispositivos de distancia:** Este tipo de protección realiza mediciones constantes de voltaje y corriente para determinar la impedancia del sistema. El valor que se obtiene de impedancia se relaciona directamente con la distancia entre la falla y el sistema de protección, por lo tanto, es posible definir zonas de protección para cada artefacto. A través de sistemas de comunicación es posible mejorar la selectividad de este tipo de protecciones.

4. **Dispositivos diferenciales:** Se encuentra basado en la ley de Kirchhoff, la cual indica que la corriente que ingresa y sale de un nodo posee igual magnitud. En otras palabras, estos dispositivos comparan la corriente de entrada y salida de un circuito de zona definida. Cuando ocurre una falla las magnitudes discrepan y el sistema de desconexión es activado. Este sistema de protección es muy utilizado en contextos domésticos y de protección de artefactos como transformadores.

A continuación, se presenta la tabla 2.1 que explícita ventajas y desventajas de estos sistemas de protección [14].

Esquema de protección	Ventajas	Desventajas
Basado en Bajo Voltaje	Independencia de valor y dirección de corriente de falla	-Vulnerable a incidentes transientes -Deficiencia para detectar fallas de alta impedancia -Dificultad para coordinación
Sobre-corriente	Localización eficiente de la falla en alimentadores radiales con penetración de recursos distribuidos	-Coordinación necesaria para fallas en ambos sentidos -Configuración adaptativa puede ser requerida
Diferencial	-Independiente de magnitud de corriente de falla -Independiente de localización, tamaño y participación de elementos distribuidos	-No posee protección de respaldo para zonas vecinas -Necesita altos niveles de comunicación -Sincronización del sistema es necesario para grandes líneas
Distancia	Solución muy apropiada para micro-redes aisladas	Presenta problemas para calcular impedancia en sistemas con recursos distribuidos
Adaptativo	Controla de manera sensible y selectiva para cada condición de operación de la micro-red	-Necesidad de comunicación confiable -Necesidad de gran cantidad de información para operación en tiempo real -Dificultad para ser diseñado apropiadamente

Tabla 2.1: Ventajas y desventajas de sistemas de protección

Es importante destacar que un quinto esquema de protección ha sido añadido en la tabla 2.1, el control adaptativo. Dicho sistema de protección será explicado con más detalle en la sección 2.2.4, debido a ser el esquema escogido para el trabajo de título.

Es necesario hacer hincapié en que los sistemas convencionales de protección, ampliamente utilizados en grandes SEP, no aplican de igual forma a las micro-redes. Las razones principales para que esto suceda son debido a que las micro-redes presentan [3]:

- Magnitudes eléctricas similares dentro de ella, dado sus bajos niveles de consumo y cercanía de estos a las fuentes de generación.
- Flujos de potencia bidireccionales producto de la generación distribuida y sus ubicaciones relativas.
- Diversos puntos de conexión de recursos distribuidos, los cuales influyen en el punto de operación normal del sistema alterando los flujos de potencia y generando distintas pérdidas por su posición relativa con los consumos.
- La incorporación de tecnologías basadas en energías renovables, las cuales poseen una variabilidad asociada a sus fuentes primarias de producción que puede afectar la calidad de suministro eléctrico y su despacho. Lo anterior no es menor, pues una micro-red permite la penetración de este tipo de energías, pudiendo tener una alta correlación la variabilidad de la generación con su operación normal.
- Corrientes de sistema limitadas, debido a la amplia presencia de equipos de electrónica de potencia.
- Dos posibles modos de operación (conectado o aislado de la red).

Razones que se resumen en la figura (2.8).

Resumen de características I_{cc} (Dispositivos reaccionarios)					
Niveles son similares a lo largo de la micro-red	Pueden presentar flujos bidireccionales	Influye punto conexión DG	Bajos niveles con electrónica potencia presente	Afectada por intermitencia fuentes renovable	Influye el modo de operación

Figura 2.8: Resumen de características de corriente de falla

2.2.3. Coordinación de protecciones

El concepto de coordinación de los sistemas de protecciones nace de la necesidad de otorgar selectividad y eficiencia para el despeje activo de fallas en una red. Esto refiere a que al presentarse una falla, esta debe ser atendida por la protección más cercana (inmediatamente aguas arriba) de la falla, aislando la menor zona y con la mayor rapidez posible.

El proceso de coordinación de un sistema de protecciones provoca un retardo en la actuación del mismo, ya sea por otorgar *delays* convenientemente sobre la acción de las protecciones o para realizar la coordinación misma de los dispositivos a través de un sistema de comunicación común. Es por esto que habitualmente se enfrentan los conceptos de protección y coordinación de protecciones [11].

Tal y como se indica en [11], una estrategia habitual de coordinación consiste en utilizar sistemas primarios y de respaldo (*Backup Protection*). Este concepto posee la lógica de que cuando una falla ocurre, el sistema de protección más cercano debe actuar (pasando a ser el sistema primario), sin embargo, cuando el sistema primario falla o no actúa, la siguiente protección más cercana (*Backup*) debe hacerlo. Esto podría replicarse las veces que sea necesario, siendo la responsabilidad de aislar la falla del elemento de protección disponible y operativo siguiente.

Para realizar lo expuesto en el párrafo anterior, se utilizan diversas estrategias, siendo una de las más comunes el *Time Grading* [11]. Consiste en configurar tiempos de actuación de los elementos de protección de acuerdo con su distancia a la falla. Si la protección primaria no actúa en el tiempo definido, la protección de respaldo lo hará bajo su propio tiempo de retardo o *delay*. En los sistemas de protección basados en sobrecorriente (Sección 1) es posible coordinar el *Time Grading* utilizando las curvas de tiempo inverso (TOC) con su *delay* propio [15], TDM en el caso de la ecuación 2.1.

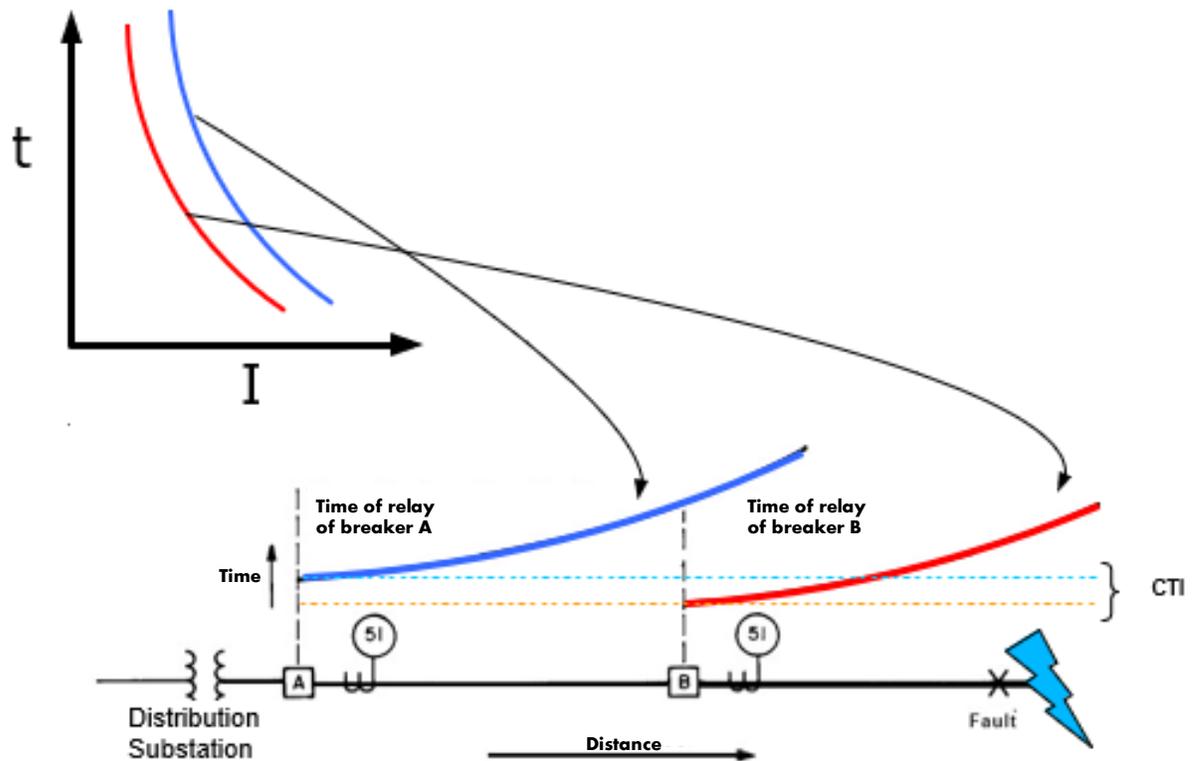


Figura 2.9: Esquema de coordinación básica mediante retardo de tiempo en curvas de sobrecorriente

La figura 2.9 expone la coordinación de dos elementos de protección caracterizado uno por el color rojo y el otro por azul. Cuando la falla sucede, la protección de curva roja debe actuar en primera instancia, si esto no ocurre, la protección de curva azul se acciona para proteger la zona aguas arriba de la red. Esto se realiza mediante el ajuste del tiempo *delay* (TDM) de la curva de sobrecorriente, lo cual significa un movimiento vertical de la curva en el gráfico.

2.2.4. Protección adaptativa

De acuerdo con [11] y [12], un sistema de protección adaptativo permite proteger varias configuraciones de micro-red frente a distintas condiciones de fallas. Lo anterior se logra calculando diferentes configuraciones para los relés y guardándolos en una base de datos. En este contexto, el sistema adaptativo es capaz de proteger al sistema tanto en operación aislada como en su funcionamiento conjunto con la red principal de energía.

Para tener un sistema más efectivo, es común encontrar complejos sistemas de comunicación integrados a la protección. Con su presencia los relés son configurados periódicamente con mediciones en tiempo real y con mejores índices de coordinación. Así mismo, es posible

configurar las curvas de tiempo definido o inverso de los relés para cada modo de operación. De esta forma, cuando la micro-red se encuentra funcionando en modo aislado, se deben configurar los relés de sobre corriente para funcionar de acuerdo a curvas de tiempo definido [9], debido a que la magnitud de la corriente no será excesivamente grande para realizar diferenciación entre tiempos de ejecución de los sistemas de protección.

Las protecciones adaptativas satisfacen de gran manera la flexibilidad que un sistema de protección debe tener. En una micro-red con constantes cambios en su topología (debido a la Generación Distribuida, sus modos de operación, etc) el sistema adaptativo debe reconfigurarse constantemente para enfrentar fallas bajo diversos puntos de operación.

Para fines de este trabajo de título, el control adaptativo es seleccionado dada la versatilidad que ofrece frente a una micro-red con constantes cambios de topología. Además, satisface las necesidades y problemáticas presentadas en la figura 2.10 y que resume tanto la motivación como el aporte del presente trabajo.

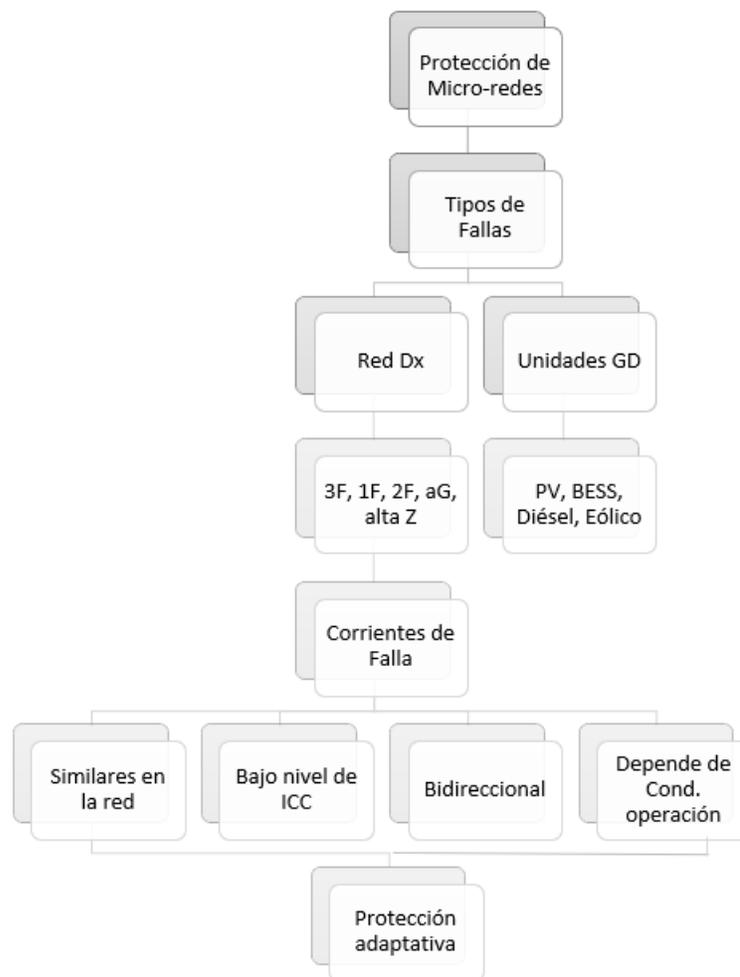


Figura 2.10: Condiciones de micro-red aptas para protección adaptativa

2.3. Laboratorio de Energía y Accionamientos

2.3.1. Topología de Micro-red

La micro-red presente en el laboratorio de energía y accionamientos del departamento de ingeniería eléctrica cuenta con un sistema trifásico compuesto por 6 barras. Su configuración es radial y posee capacidad de conexión de cargas a cada una de las barras presentes. Se conecta a la red principal de energía mediante un transformador 1:1 con un voltaje fase-fase aproximado de 380[V]. La figura 2.11 es un esquemático de la micro-red descrita.

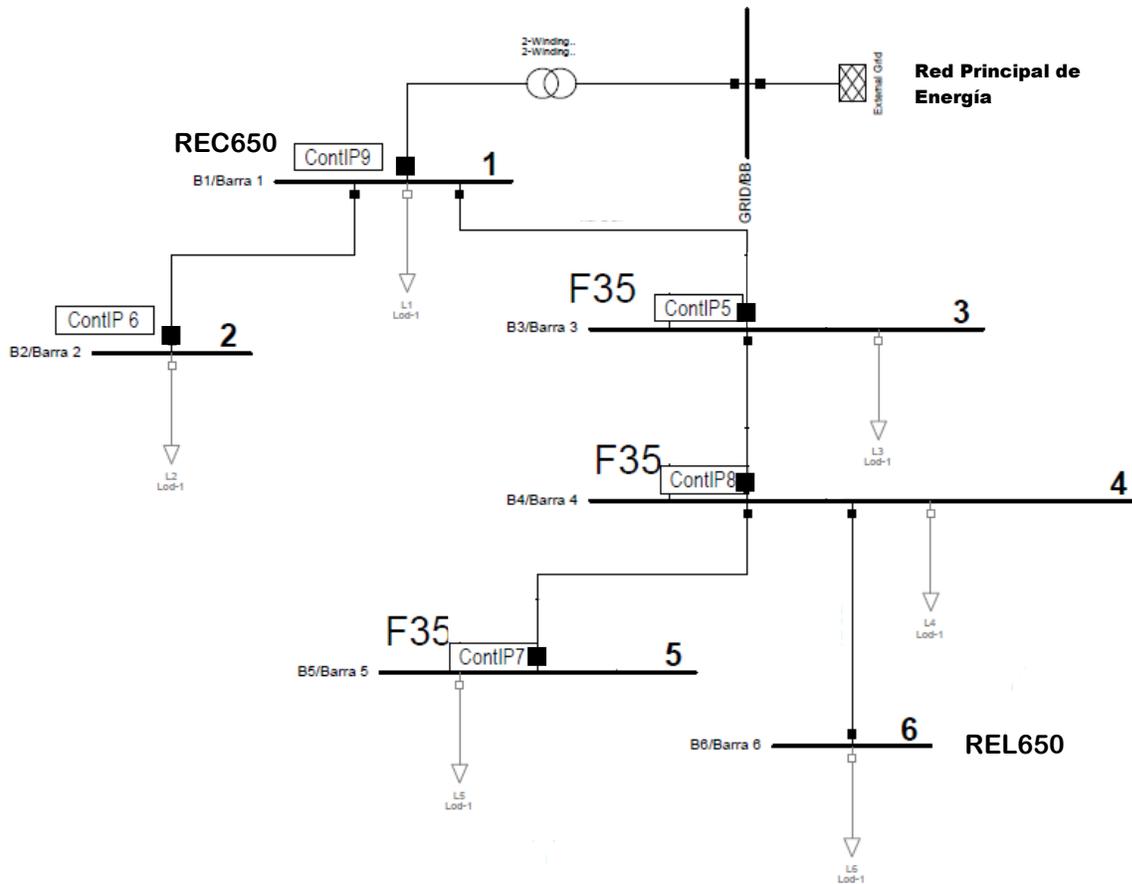


Figura 2.11: Topología Micro-red presente en laboratorio

Además, en la figura 2.11 se exponen los interruptores auxiliares con sus relés asociados, los cuales son elementos fundamentales para el desarrollo de la investigación. Estos son:

- **Relé ABB REC650**: Utilizado para el monitoreo y protección de diversos SEP. Posee la capacidad de adaptarse a distintos tipos de estaciones y configuraciones. Sus funciones principales son otorgar protección frente a sobrecorrientes, aumento de temperatura en conductores y funciones basadas en baja o sobretensión. También es importante destacar la capacidad de realizar reconexiones automáticas y de gran velocidad sobre los interruptores del sistema.
- **Relé ABB REL650**: Se encuentra diseñado para líneas aéreas de transmisión. Utilizado para sistemas con alto nivel de carga y de tensión en los cuales la apertura inmediata es primordial. Posee 5 zonas de protección y el relé es capaz de monitorear cada una de ellas de forma periódica e independiente. Para realizar la configuración de ambos equipos ABB se utiliza el software **Wavewin**.
- **Relé General Electric F35**: Es un dispositivo multifuncional capaz de ofrecer protección a 5 o 6 alimentadores dependiendo de la presencia de mediciones de tensión en las barras. Proporciona protección para casos de sobrecorriente y caídas de tensión. Posee funciones para la reconexión de interruptores, diagnósticos de falla e informes de las mismas. El software utilizado para la configuración de estos dispositivos es **EnerVista**.

Por otro lado, se presentan en la tabla 2.2 los parámetros de la micro-red del laboratorio. Cabe destacar que el sistema se encuentra diseñado para simular tanto líneas de transmisión como de distribución. Los parámetros son:

Línea	R[Ω]	X[Ω]
L12	0.78590384	0.37474596
L13	0.88565017	0.72806038
L34	1.00113985	0.48939002
L45	1.27147659	0.55283342
L46	0.83676405	0.35729613

Tabla 2.2: Parámetros red de laboratorio

Los datos del motor eléctrico de inducción trifásico utilizado en la experiencia, particularmente en la sección de pruebas, se exponen en la tabla 2.3.

Tipo de conexión	Velocidad [RPM]	Potencia [kW]	C. Nominal [A]	C. en vacío [A]
Estrella Y	1440	1.3	3	1.4
Delta	1440	4.0	9	4.5

Tabla 2.3: Datos motor de inducción trifásico utilizado en pruebas y simulación

Sumado a lo anterior, se presentan en la tabla 2.4, los valores base utilizados habitualmente para cálculos en esta micro-red.

Parámetro Base	Magnitud	Unidad
S	10000	[VA]
V	380	[V]
Z	14.44	[Ω]
I	26.32	[A]

Tabla 2.4: Valores base para micro-red

2.4. Estado del Arte y experiencias anteriores

Como se ha indicado, la materia de protecciones presenta grandes desafíos. Esto se debe a que las micro-redes son sistemas recientes, a pesar de que su utilización se ha expandido por todo el mundo. En esta primera instancia, se expondrán los avances realizados por memoristas anteriores sobre la micro-red del Laboratorio de Energía para luego presentar algunas experiencias internacionales.

2.4.1. Trabajos anteriores

En materia de protecciones sobre la red del Laboratorio de Energía y Accionamientos, la presente investigación se encuentra basada en tres trabajos anteriores expuestos a continuación.

Infraestructura de Hardware y Software para Desarrollo de Sistemas de Protección en Micro-redes de Baja Tensión, Mauricio Neut

Mauricio Neut, a través de su trabajo de título [16], implementó un espacio destinado a la investigación y docencia de protecciones eléctricas de media y baja tensión. En este contexto se dotó al departamento de Ingeniería Eléctrica con dispositivos electrónicos inteligentes para el monitoreo y control de la red.

Durante este trabajo se consiguió la infraestructura física (Hardware), se realizó su instalación y la comprobación de su correcto funcionamiento.

Para realizar el monitoreo integrado de la red completa se instaló un sistema de comunicación mediante protocolo MODBUS/TCP (Exceptuando los dispositivos REL650 y REC650).

Al final del proyecto realizado por este trabajo de título, se logra observar un monitoreo en línea de los sistemas de protección y de las mediciones obtenidas en la micro-red de forma confiable. Además, se realiza la configuración de los dispositivos utilizados para su función de sobrecorriente y se comprueba su funcionamiento remoto a través de una simulación de falla.

Experiencia Docente de Protecciones para una Micro-red Basada en el Estudio de Funciones de Sobre Corriente y Baja Tensión, Pablo González

Durante este trabajo de título [13], Pablo González implementó una experiencia docente de laboratorio dirigido hacia el estudio de protecciones de sistemas eléctricos de potencia con

generación distribuida y micro-redes.

La experiencia desarrollada se basa en la configuración remota del relé F35 de acuerdo con un punto de operación del sistema dado en el modo conectado a la red principal. En este contexto se desarrolla la función de sobrecorriente y baja tensión por separado para luego ser integrados en una única prueba, probando que la redundancia de protecciones es ventajosa para el sistema.

A través del programa EnerVista se logra configurar los relés F35, calibrando la curva de tiempo definido o inverso de los dispositivos.

Por último se genera una experiencia docente, con una metodología clara para la recreación de la misma e incluso ser una fuente de recursos para otras experiencias prácticas futuras, tanto en el ámbito docente como en investigación.

Evaluación de Protecciones Adaptativas para una Micro-Red, Gustavo Moreno

Gustavo, mediante su trabajo de título [17], realiza un completo estudio teórico de protecciones en la micro-red del Laboratorio de Energía y Accionamientos. En su estudio simula mediante *Power Factory Digsilent* distintas topologías posibles adoptables por la red descrita. Acto seguido, obtiene sus corrientes de falla y determina la configuración necesaria de las curvas de protección de la Micro-red.

Analizando una gran cantidad de combinaciones topológicas, el estudio permite establecer *LookUp Tables* de curvas de protección. Cabe destacar que el estudio no incorpora pruebas experimentales en la Micro-red del laboratorio.

2.4.2. Experiencia Internacional

En [6] se expone la construcción de un laboratorio con una micro-red en Australia por CSIRO Energy Centre. En ella se presenta empíricamente el problema de la limitación presente en la corriente de cortocircuito cuando la micro-red pasa a funcionar en modo Off-Grid. Además, se implementa un sistema de protección por sobrecorriente y se expone su dificultad para la operación bidireccional y variable, debido a la generación distribuida.

En el documento de investigación [18] se presenta la teoría y aplicación, mediante ejemplo, de un sistema de protecciones adaptativo On-Grid y Off-Grid. En ella se utilizan relés numéricos para ajustar curvas de protección para ser guardadas en la memoria interna de los relés. Este caso se conoce como Off-line debido a que el cálculo se realiza fuera de la operación del sistema, siendo almacenado mediante *Look-up Tables*.

En [7] se presenta una micro-red construida para un condominio residencial en Fortaleza, Ceará, Brasil. En ella se ha gestionado un sistema de protección adaptativo preparado para ofrecer seguridad en el sistema tanto en el modo On-Grid como Off-Grid.

Por otro lado, en [19] se expone una micro-red construida en un Boston-Bar utilizando un sistema de protecciones basado en sobrecorriente y una red de comunicaciones. Además, en

él se plantea el problema y solución para alimentar los sistemas de protección en el modo Off-Grid de manera ininterrumpida.

En Estados Unidos, precisamente en la cárcel de Santa Julia, se utiliza una protección por sobre-corriente y detección de fallas externas para activar el modo de operación aislado. Para fallas internas, incorpora protección por baja/sobre tensión y otra para baja/sobre frecuencia. [20]

Capítulo 3

Metodología

El objetivo y alcance principal de la investigación está centrado en proponer y ejecutar un modelo que permita utilizar la micro-red del laboratorio en distintas topologías de forma controlada y segura. Con el fin de llevar a cabo lo anterior, es necesario leer, desarrollar y dar instrucciones sobre la micro-red.

Este capítulo se encuentra dividido en dos secciones principales. La primera de ellas, *Estrategia Metodológica*, indica el procedimiento conceptual y lógico utilizado durante el trabajo que permite satisfacer los objetivos y alcances planteados en el capítulo de introducción. El segundo ítem, *Trabajo Realizado*, data el procedimiento cronológico hecho por el autor para lograr los resultados y crear el producto final de trabajo.

3.1. Estrategia Metodológica

La estrategia seleccionada se encuentra definida tanto en base al estado del arte, particularmente de la micro-red presente en el laboratorio, como a los objetivos y alcances planteados por el autor. Dicha estrategia consiste en capturar o leer valores provenientes desde la micro-red en un software de programación que permita realizar cálculos y ejecución de instrucciones. Para ello, se utiliza Python debido a la gran oferta de librerías disponibles que son atingentes al deseo de utilizar un servidor OPC ya operativo en el laboratorio.

Una vez capturados dichos valores en Python, se realiza un tratamiento de datos que permite identificar la topología con la cual está operando la micro-red y corresponder las variables de entrada con una configuración predeterminada mediante un *Look-up Table*. Este tratamiento de datos refiere a que la obtención de ellos se recibe en formatos poco intuitivos para trabajar, como binario o hexadecimal.

Obtenida la configuración necesaria, Python envía las instrucciones al relé con las configuraciones de las curvas de protección y selección del *Setting Group* para poder afrontar una situación de falla en la topología actual.

Finalmente el proceso es iterativo leyendo los parámetros de la micro-red constantemente hasta detectar un cambio topológico nuevo y activar el sistema de protección adecuado. De esta manera, la micro-red se mantiene protegida en todo momento ante fallas, activación de interruptores que implican en un cambio topológico.

En la figura 3.1 se expone un diagrama de flujo que resume el proceso explicado en los párrafos anteriores. Es importante destacar que el sistema de protecciones, en este caso el relé Multilin F35, trabaja de forma simultanea con el programa de Python. Esto es, mientras el sistema de protección se encuentra adaptando sus configuraciones internas, la micro-red permanece monitoreada por el esquema de protección de ese momento.

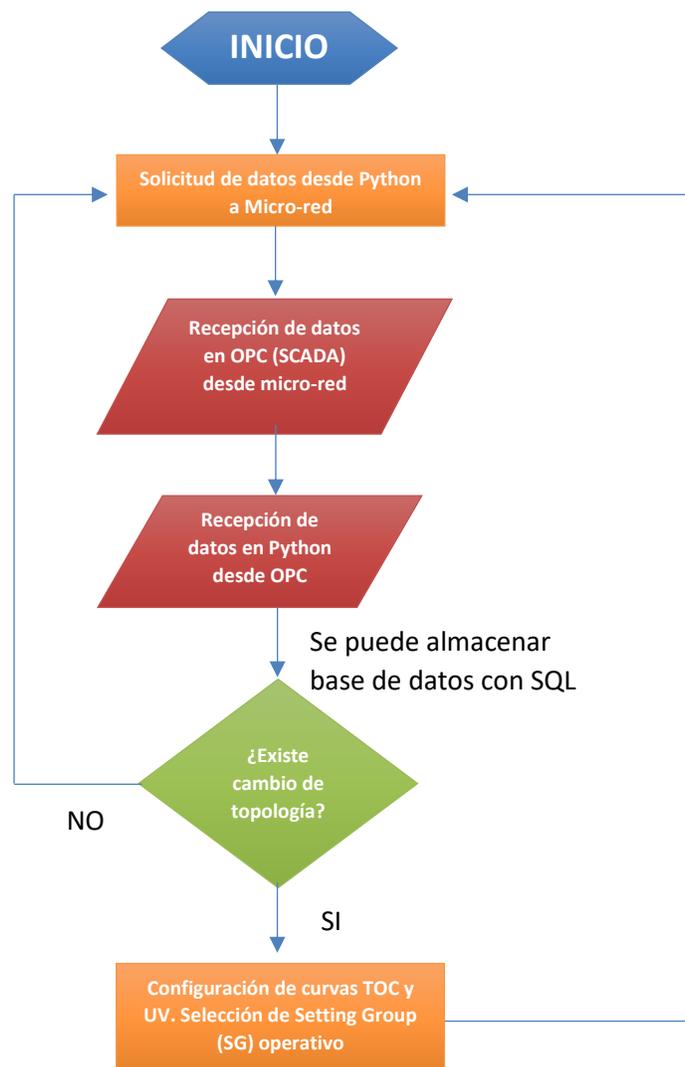


Figura 3.1: Diagrama de Flujo de la estrategia metodológica utilizada

3.2. Trabajo Previo a Laboratorio

3.2.1. Definición de Objetivos, Alcances y Revisión Bibliográfica

1. Objetivos específicos: Desarrollar y proponer metodología para configurar las protecciones de la micro-red. Para lo anterior, se requiere leer y escribir datos sobre los sistemas de medición y de actuación de la misma.
2. Alcances: Proporcionar una configuración de protecciones que permita la operación bajo diversas topologías de operación entendiendo que las condiciones de falla de cada esquema es diferente. Permite favorecer estudios prácticos de operación de la micro-red de forma segura.
3. Revisión bibliográfica: Se realiza una revisión bibliográfica orientada a la identificación de futuros problemas y soluciones a partir de experiencias anteriores, tanto a nivel local como internacional.

3.2.2. Recopilación de datos de elementos de laboratorio

Como indica el título, este paso consiste en la toma de información relevante sobre todos los elementos que se involucrarán en esta experiencia. Entre ellos se encuentra:

1. Estudiar estado de cargas RLC presente en el Laboratorio y del motor de inducción utilizado para emulación de falla.
2. Conseguir parámetros de la micro-red, resistencia/reactancia de líneas en modo distribución. Buscar y conseguir información de los relés.
3. Capacidad de relés disponibles en el laboratorio, comprobación de que los elementos estén operativos, incluido el software y los elementos de la red del laboratorio. Para esto se propone no sólo revisar las fichas técnicas de los relés (manuales F-35, REL y REC de marca ABB) sino que se propone recrear la experiencia realizada durante el curso EL7045-1 Micro-redes y generación distribuida [13] para la familiarización y prueba de los dispositivos de protección presentes en la micro-red.

Capítulo 4

Implementación en laboratorio

4.1. Captura de datos a OPC cliente

La captura de datos provenientes de la micro-red es realizada desde los dispositivos de medición PM750, ION 8600 pertenecientes a Schneider Electric y el equipo Multiple Feeder System F35 de la serie UR de la marca General Electric. Este último se impone como el principal actor dentro del esquema, debido a su capacidad de medición de las barras y de la actuación del relé sobre los elementos de protección termomagnéticos.

Los dispositivos antes mencionados se encuentran comunicados a través de protocolo TCP/IP por lo que se ha seleccionado un servidor especializado en la recepción de mediciones mediante este protocolo. El servidor seleccionado es el *INAT TCP/IP-H1 OPC Server*, el cual permite la medición de controladores multi-protocolo. Es importante destacar que el servidor escogido permite realizar los procedimientos sin necesidad de ser un experto en protocolo MODBUS. La escritura y lectura de datos sobre la micro-red se realiza mediante direcciones MODBUS, explicadas en las siguientes subsecciones.

Para incorporar los dispositivos de medición al servidor, que ordena la captura de datos, se debe iniciar el servidor obteniendo una pantalla de inicio como la expuesta en la figura 4.1.

Una vez abierto el servidor, la prioridad es establecer la comunicación con cada uno de los dispositivos. Para esto se procede a seleccionar en el menú la opción *Show Access Path List* y mediante click derecho seleccionar la opción *New Access Path*. A continuación emergerá una ventana nueva (Ver Fig. 4.2) con el resumen de datos, particularmente en esta sección se inscribe el nombre del Path que será creado.

Al seleccionar la opción *Server Settings* se despliega la ventana expuesta en la figura 4.3. En esta nueva ventana se configura la dirección IP del dispositivo, el número serial (En caso de los PM), el tipo de protocolo mediante el cual se entregarán las instrucciones, en este caso, MODBUS al cual le pertenece el puerto 502.

Esta información debió ser obtenida previamente, a través de investigación de la instala-

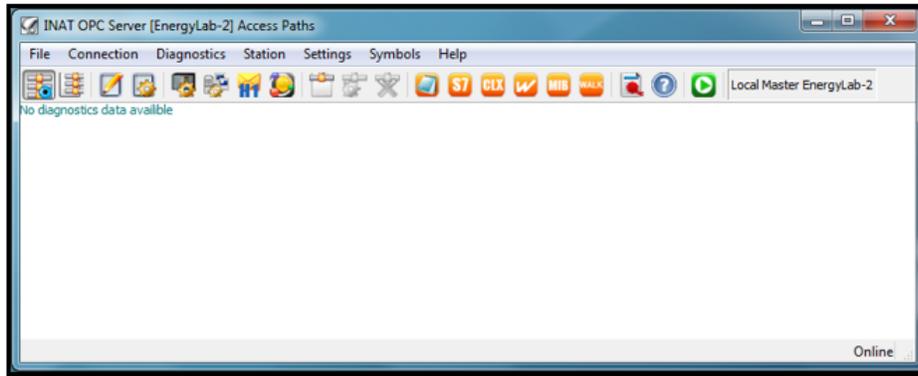


Figura 4.1: Pantalla de inicio a INAT TCP/IP-H1 OPC Server

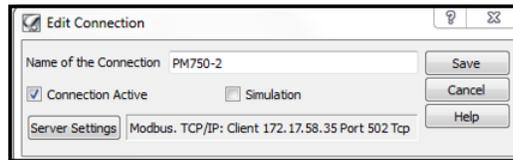


Figura 4.2: Ventana de configuración de dispositivos

ción de los dispositivos [16] y el manual del dispositivo [21].

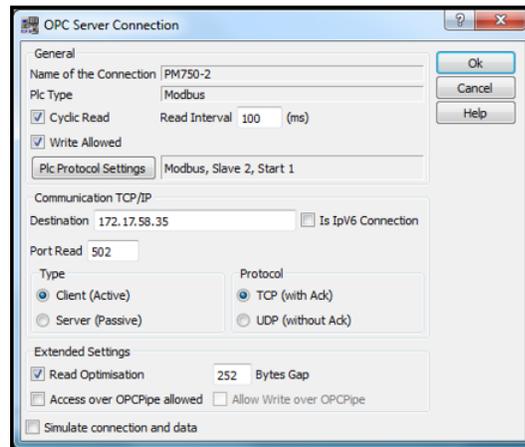


Figura 4.3: Ventana avanzada de configuración de dispositivos

La tabla 4.1, presentada a continuación, data las direcciones IP utilizadas de los dispositivos de laboratorio.

Dispositivo	Dirección IP
Medidor PM750(*)	172.17.58.35
Medidor ION8600	172.17.58.36
Relé F35 Multilin	172.17.58.37

Tabla 4.1: Direcciones IP de dispositivos de medición.

Es importante destacar que los dispositivos de medición PM750 presentes en la red del laboratorio, comparten su dirección IP. Sin embargo, dentro de las opciones del servidor (En

el menú **PLC Protocol Setting** (Ver Figura 4.3, estos pueden ser diferenciados a través de una *ID Slave* única, siendo el PM750-2 el de ID igual a 1 y creciendo respectivamente hasta el PM750-9 que corresponde a un ID de 8.

Por otro lado, es importante seleccionar que el método de envío de instrucciones se realiza mediante el protocolo MODBUS TCP/IP. Una vez realizada esta acción para todos los dispositivos, se obtiene la siguiente pantalla de visualización (Ver Fig. 4.4).

Type	Access Path Name	Job	NetProt	Dest Address	Parameters	Parameter Details
Ip	F35	Modbus	TCP/IP Client	172.17.58.37	Port 502 Tcp	Write, Interval 1000, HPP
Ip	ION8600	Modbus	TCP/IP Client	172.17.58.36	Port 502 Tcp	Write, Interval 100
Ip	PM750-2	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-3	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-4	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-5	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-6	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-7	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-8	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP
Ip	PM750-9	Modbus	TCP/IP Client	172.17.58.35	Port 502 Tcp	Write, Interval 100, HPP

Figura 4.4: Pantalla Path Access de INAT TCP/IP-H1 OPC Server con todos los dispositivos

El siguiente paso consiste en la identificación e inscripción de los ítems o etiquetas mediante direcciones MODBUS. Cada ítem responde a un parámetro de la micro-red que se requiera leer o escribir. Para realizar cualquier medición o acción es necesario ingresar la instrucción mediante una dirección MODBUS asociada al ítem. Por ejemplo, para solicitar y recibir una medición de voltaje fase-fase de una barra y fase específica, se utiliza una dirección MODBUS y se asocia a un ítem generado para esa medición en particular. Sin embargo, la solicitud del mismo parámetro para otra barra o fase requiere una dirección e ítem distinto al de la primera medición.

Para obtener las direcciones se utilizan los *Memory Maps*, los cuales son únicos para cada tipo de dispositivo utilizado. Es importante destacar que las direcciones puestas en el *Memory Map* del Multiple Feeder System F35 [22] se encuentran en formato hexadecimal, mientras que en el servidor INAT TCP/IP-H1 OPC Server son recibidos en formato decimal. Las direcciones utilizadas durante el presente trabajo se encuentran disponibles, de manera ordenada, en la sección Anexos 7.1 del presente informe.

La inscripción de las direcciones MODBUS se realiza en el INAT TCP/IP-H1 OPC Server, en la sección de *Server Symbol Edit* presente en el menú principal. En él, se selecciona el dispositivo y se crea un nombre para el ítem y la dirección MODBUS que se quiera medir y asociar a ese nombre, como se indica en la figura 4.5. Mediante la utilización de puntos "...^{es} posible generar secciones de nombre, en este caso *Mediciones* y *Curvas* son dos secciones que incorporan un conjunto de ítems.

Una vez inscritos los códigos o direcciones necesarios para el estudio se procede a guardar la configuración y reiniciar el servidor (Botón verde del menú principal).

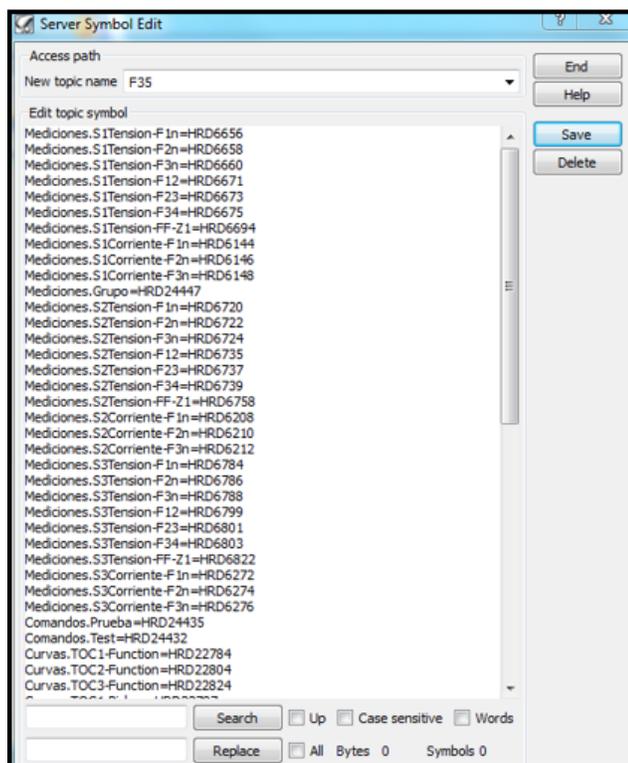


Figura 4.5: Representación de ingreso de direcciones MODBUS a Servidor OPC

4.1.1. Visualización de Datos en OPC Cliente

Para revisar el estado de conexión y poder visualizar las mediciones en tiempo real, se utiliza el OPC Test Client. De manera similar a la formación de los grupos en el servidor INAT, al abrir el cliente se deben crear grupos que concentrarán las etiquetas inscritas en el servidor. Estos grupos se crean y visualizan en el OPC cliente, a la izquierda de la pantalla mostrada en la figura 4.6

La elección de etiquetas para cada ítem a utilizar se realiza seleccionando el grupo y abriendo el menú *Add Ítem*. Si las direcciones fueron ingresadas correctamente en el servidor, aparecen directamente en el *Root* del software cliente. Esta situación se expone en la figura 4.6

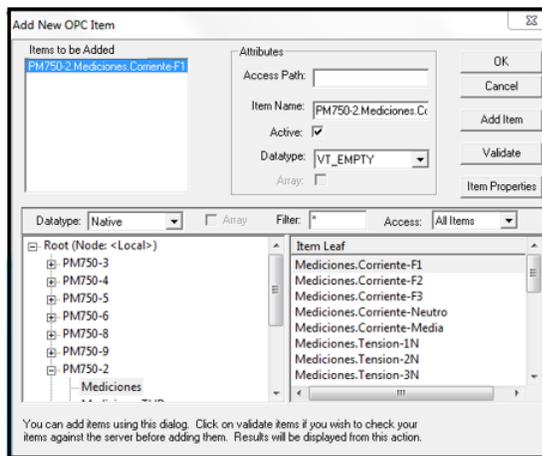


Figura 4.6: Menú para inscripción de etiquetas a visualizar en OPC cliente

En este caso particular, se genera un grupo para las mediciones de cada PM750, el ION8600 y el relé F35. Un grupo extra se crea encargado de la revisión de los contactos de toma de carga y para la ejecución de la instrucción de apertura de estos mismos. Los demás grupos se encuentran creados para la escritura de las configuraciones de protección sobre el relé Multilin F35, curvas de protección y el envío de entradas virtuales para la ejecución de instrucciones lógicas. La figura 4.7 expone el OPC cliente una vez que todas las etiquetas (Medición y Acción) han sido cargadas y agrupadas (Grupos a la izquierda de la imagen).

ItemID	Sub Value	Sub Quality	Sub Updates	Update Rate	Run. Avg
PM750-3.Mediciones.Corriente-F1	0	Good	1	0	0.005979
PM750-3.Mediciones.Corriente-F2	0	Good	1	0	0.005979
PM750-3.Mediciones.Corriente-F3	0	Good	1	0	0.005979
PM750-3.Mediciones.Corriente-Ne...	0	Good	1	0	0.005979
PM750-3.Mediciones.Corriente-Me...	0	Good	1	0	0.005979
PM750-3.Mediciones.Tension-1N	2341	Good	18	0	0.107614
PM750-3.Mediciones.Tension-2N	2350	Good	24	0	0.143486
PM750-3.Mediciones.Tension-3N	2333	Good	23	0	0.137507
PM750-3.Mediciones.Tension-FF...	2342	Good	22	0	0.131529
PM750-3.Mediciones.Tension-12	4065	Good	22	0	0.131529
PM750-3.Mediciones.Tension-23	4057	Good	22	0	0.131529
PM750-3.Mediciones.Tension-31	4044	Good	23	0	0.137507
PM750-3.Mediciones.Tension-FF...	4057	Good	23	0	0.137507
PM750-3.Mediciones.Potencia-Acti...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Acti...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Acti...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Acti...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Rea...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Rea...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Rea...	0	Good	1	0	0.005979
PM750-3.Mediciones.Potencia-Rea...	0	Good	1	0	0.005979
PM750-3.Mediciones.FP	0	Good	1	0	0.005979
PM750-3.Mediciones.Frecuencia	4996	Good	18	0	0.107614

Figura 4.7: Visualización OPC Test Client

4.2. Enlace de datos desde OPC a Python

El traspaso de información entre el servidor OPC y Python se realiza mediante la librería especializada *OpenOPC*. Dicha librería es de fuente abierta y se puede utilizar mediante un módulo de Python o a través de una instalación directa de la librería (`pip install`, un archivo ejecutable, entre otros). Para este caso se utiliza la librería mediante un módulo, por lo que es estrictamente necesario que los programas compartan la misma carpeta.

Es destacable que *OpenOPC* requiere las librerías *Pyro* y *Pywin32* para su correcto funcionamiento, las cuales son instaladas directamente mediante los comandos `pip install Pyro` y `pip install Pywin32` utilizados en la consola del sistema.

Durante las primeras líneas de código, se inicia la comunicación con el servidor. Es necesario inscribir las direcciones IP del ordenador que captura datos desde el OPC (Host) y del computador en el cual se está utilizando Python (Gateway). Particularmente ambas direcciones son iguales debido a que es el mismo computador quien ejecuta ambas funciones. Es posible modificar la dirección Gateway para una utilización remota del programa. Además, se ingresa el nombre del servidor OPC utilizado, en este caso, *INAT TcpIpH1 OPC Server*.

Por último, se inicia la conexión utilizando estos parámetros como se muestra en el Código 4.1.

```
1 #OPENOPC VARIABLE DEFINITIONS
2 gateway = '172.17.58.28' #server computer IPv4
3 opchost = '172.17.58.28' #server computer IPv4
4 opcserv = 'INAT TcpIpH1 OPC Server' #server name (use 'python opc.py -i'
   or 'python opc.py -q' in cmd line to find)
5 opc = OpenOPC.open_client(gateway)
6 opc.connect(opcserv, opchost)
```

Código 4.1: Establecimiento de comunicación con OPC

La lectura de datos se realiza mediante el comando `"opc.read()"`. Para utilizarlo es necesario especificar en su argumento la(s) etiqueta(s) a leer. Para obtener todas las etiquetas de una sección (Por ejemplo "Mediciones") de forma genérica, se utiliza el comando `opc.list()` con el dispositivo y sección que contiene las etiquetas. La lectura de datos se escribe en un único arreglo de vector (guardada en la variable "v" del código), compuesto por listas de 4 elementos de la forma:

(Nombre del ítem, Valor, Estado, Hora y Fecha)

Un bloque de código de lectura, a modo de ejemplo, es presentado en el Código 4.2. El código completo de Python se encuentra en la sección Anexos del presente trabajo.

```
1 def leer_F35():
2     tags=opc.list('F35'+ '.'+opc.list('F35')[0])+opc.list('F35'+ '.'+opc.
   list('F35')[1])
3     opc.read(tags,group='F35')
```

```

4     v=opc.read(group='F35')
5     opc.remove('F35')
6     return(v)

```

Código 4.2: Ejemplo de lectura de datos desde OPC a Python

Finalmente, se realiza la lectura iterativa para cada uno de los dispositivos seleccionados: Medidores PM, ION8600 y relé Multilin F35, almacenando cada medición en un archivo de base de datos SQL mediante la librería SQLite de Python, el cual es explicado a continuación.

4.3. Almacenamiento de datos SQLite

Un archivo de base de datos (con extensión .db) permite realizar una colección de datos. Un SQL es un lenguaje estándar para crear, modificar y leer una base de datos bajo un formato más interactivo y ordenado.

Para efectos de este trabajo de título, el almacenamiento de datos en un archivo SQL se realiza con el doble objetivo de facilitar el manejo de información de manera ordenada (tablas) y de proveer al laboratorio de un registro histórico de mediciones de la micro-red. Dado el volumen moderado de datos que se desea almacenar, se decide utilizar la librería *SQLite* de código abierto para Python.

Una vez importada la librería SQLite, se prosigue a abrir el archivo de almacenamiento (o crear en caso de que el archivo no exista). El archivo generado se denomina *Microrred.db*. A continuación, se genera a una tabla para cada dispositivo de medición y el relé F35, de manera de ordenar fácilmente la información. Lo expuesto durante el presente párrafo se visualiza en el código 4.3.

El tercer bloque de código consiste en generar una función capaz de inscribir variables en las tablas antes mencionadas, con el objetivo de generalizar la función para la lectura de cada uno de los dispositivos con sus mediciones respectivas.

```

1 import sqlite3
2 from sqlite3 import Error
3
4 def sql_connection():
5     try:
6         con=sqlite3.connect('Microrred.db')
7         return con
8     except Error:
9         print(Error)
10
11 def sql_table(con):
12     cursorObj=con.cursor()
13     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM2(Medicion text, Valor
14     real, Estado text, FechaHora text)")
15     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM3(Medicion text, Valor
16     real, Estado text, FechaHora text)")

```

```

15 cursorObj.execute("CREATE TABLE IF NOT EXISTS PM4(Medicion text, Valor
16 real, Estado text, FechaHora text)")
17 cursorObj.execute("CREATE TABLE IF NOT EXISTS PM5(Medicion text, Valor
18 real, Estado text, FechaHora text)")
19 cursorObj.execute("CREATE TABLE IF NOT EXISTS PM6(Medicion text, Valor
20 real, Estado text, FechaHora text)")
21 cursorObj.execute("CREATE TABLE IF NOT EXISTS PM7(Medicion text, Valor
22 real, Estado text, FechaHora text)")
23 cursorObj.execute("CREATE TABLE IF NOT EXISTS PM8(Medicion text, Valor
24 real, Estado text, FechaHora text)")
25 cursorObj.execute("CREATE TABLE IF NOT EXISTS PM9(Medicion text, Valor
26 real, Estado text, FechaHora text)")
27 cursorObj.execute("CREATE TABLE IF NOT EXISTS ION8600(Medicion text,
28 Valor real, Estado text, FechaHora text)")
29 cursorObj.execute("CREATE TABLE IF NOT EXISTS F35(Medicion text, Valor
30 long, Estado text, FechaHora text)")
31 con.commit()
32
33 def sql_insert(con,entities ,table):
34     cursorObj= con.cursor()
35     cursorObj.execute('INSERT INTO '+table+'(Mediccion,Valor,Estado,
36 FechaHora) VALUES(?,?,?,?)',entities)
37     con.commit()

```

Código 4.3: Creación de archivos y tablas SQL

El módulo *MiSQL.py* es utilizado en el código principal. Se genera el archivo, se crean las tablas y se guarda iterativamente cada medición. El siguiente código 4.4 expone las instrucciones mediante las cuales se envían los datos de la medición del PM750-2 al archivo de destino.

```

1     if tipo=='Mediciones':
2         con=sql.sql_connection() #abre archivo sql 'Microrred.db', Si no
3         existe crea uno (cambios en MiSQL.py)
4         sql.sql_table(con) #genera las tablas del archivo
5         Datos_PM2=leer_PM('PM750-2')
6         for var in Datos_PM2:
7             sql.sql_insert(con,var,'PM2')

```

Código 4.4: Ejemplo de almacenamiento de datos en SQLite para mediciones del PM750-2

Finalmente, mediante el software llamado *Db Browser* se visualizan las tablas generadas mediante código. Sin embargo, esta visualización es sólo ilustrativa ya que no cumple funcionalidad dentro del algoritmo. Se recomienda la utilización de este software para visualizar de manera intuitiva los datos históricos almacenados. La figura 4.8 expone la situación descrita anteriormente para uno de los dispositivos de medición.

	Medicion	Valor	Estado	FechaHora
1	PM750-3.Mediciones.Corriente-F1	0.0	Good	10/10/19 18:04:09
2	PM750-3.Mediciones.Corriente-F2	0.0	Good	10/10/19 18:04:09
3	PM750-3.Mediciones.Corriente-F3	0.0	Good	10/10/19 18:04:09
4	PM750-3.Mediciones.Corriente-Neutro	0.0	Good	10/10/19 18:04:09
5	PM750-3.Mediciones.Corriente-Media	0.0	Good	10/10/19 18:04:09
6	PM750-3.Mediciones.Tension-1N	2354.0	Good	10/10/19 18:04:09
7	PM750-3.Mediciones.Tension-2N	2358.0	Good	10/10/19 18:04:09
8	PM750-3.Mediciones.Tension-3N	2344.0	Good	10/10/19 18:04:09
9	PM750-3.Mediciones.Tension-FN-Media	2352.0	Good	10/10/19 18:04:09
10	PM750-3.Mediciones.Tension-12	4085.0	Good	10/10/19 18:04:09
11	PM750-3.Mediciones.Tension-23	4072.0	Good	10/10/19 18:04:09
12	PM750-3.Mediciones.Tension-31	4061.0	Good	10/10/19 18:04:09
13	PM750-3.Mediciones.Tension-FF-Media	4073.0	Good	10/10/19 18:04:09
14	PM750-3.Mediciones.Potencia-Activa-F1	0.0	Good	10/10/19 18:04:09
15	PM750-3.Mediciones.Potencia-Activa-F2	0.0	Good	10/10/19 18:04:09
16	PM750-3.Mediciones.Potencia-Activa-F3	0.0	Good	10/10/19 18:04:09
17	PM750-3.Mediciones.Potencia-Activa-Total	0.0	Good	10/10/19 18:04:09
18	PM750-3.Mediciones.Potencia-Reactiva-F1	0.0	Good	10/10/19 18:04:09
19	PM750-3.Mediciones.Potencia-Reactiva-F2	0.0	Good	10/10/19 18:04:09
20	PM750-3.Mediciones.Potencia-Reactiva-F3	0.0	Good	10/10/19 18:04:09
21	PM750-3.Mediciones.Potencia-Reactiva-Total	0.0	Good	10/10/19 18:04:09
22	PM750-3.Mediciones.FP	0.0	Good	10/10/19 18:04:09
23	PM750-3.Mediciones.Frecuencia	5001.0	Good	10/10/19 18:04:09
24	PM750-3.Salidas.Estado_Digital-1	0.0	Good	10/10/19 18:04:09
25	PM750-3.Comandos.Ejecutar	0.0	Good	10/10/19 18:04:09

Figura 4.8: Visualización de datos almacenados mediante Db browser

4.4. Lógica y selección de instrucciones mediante Look Up Tables

En las mediciones capturadas por el programa, bajo la metodología de la subsecciones anteriores, toma principal importancia las señales de entrada digital de contacto. Las anteriores, llamadas *Contact Inputs* datan el estado de apertura/cierre de los interruptores auxiliares ubicados en la micro-red, específicamente, dichos interruptores se encuentran posterior a cada línea de transmisión (previo a cada barra).

Conocer el estado de apertura o cierre de cada interruptor permite reconocer la topología actual en la que opera la micro-red. Estas entradas de información posibilitan elegir y seleccionar qué configuración de protección es la que corresponde utilizar. Todo lo anterior se encuentra basado en *Look-up Tables*, por lo tanto es necesario definir escenarios de estudio, los que son expuestos a continuación:

1. **Operación conectada a la red principal**
 - (a) Todas las barras operando (interruptores auxiliares cerrados)
 - (b) Desconexión de barra 5 (Contact Input 7)
 - (c) Desconexión de barra 3 y aguas abajo (Contact Input 5)
2. **Operación aislada de la red principal con GD**
 - (a) Todas las barras operando (interruptores auxiliares cerrados)
 - (b) Desconexión de barra 5 (Contact Input 7)
 - (c) Desconexión de barra 3 y aguas abajo (Contact Input 5)

La selección de las barras 3 y 5 (*Contact inputs 5 y 7 respectivamente* se realizó debido a que la toma de carga de la barra 5 es la más utilizada por los alumnos en las experiencias docentes. En tanto el interruptor de la barra 3 se selecciona debido a su importancia en la topología, al ser el interruptor aguas arriba de las barras 4,5 y 6, por lo que aportaría protección en caso de una falla en ese sector de la micro-red.

Para cada uno de los escenarios antes mencionados, se ha asignado un *Setting Group* específico del relé. Respetando el orden en que fueron enlistados anteriormente, se inscribe un número entre 1 y 6 que corresponde a su *Setting Group* particular.

Además, es importante destacar que existen jerarquías de importancia respecto a los contactos de entrada que se leen (debido al orden de barras agua abajo que existe). Por ejemplo, si la micro-red está conectada a la red y la barra 3 no opera, será el tercer caso de estudio, independiente de si la barra 5 está operando o no. Lo anterior se resume para todas las combinaciones en la tabla 4.4, además se recomienda revisar la Figura 2.11 para el correcto entendimiento de los contactos de entrada y su interruptor asociado.

Caso	Cont IP 9 (1=On-Grid, 0=Off-Grid)	Cont Ip 6 (Barra 2)	Cont IP 5 (Barra 3)	Cont IP 8 (Barra 4)	Cont IP 7 (barra 5)	SG
1	1	1	1	1	1	1
2	1	1	1	1	0	2
3	1	1	1	0	1	1
4	1	1	1	0	0	2
5	1	1	0	1	1	3
6	1	1	0	1	0	3
7	1	1	0	0	1	3
8	1	1	0	0	0	3
9	1	0	1	1	1	1
10	1	0	1	1	0	2
11	1	0	1	0	1	1
12	1	0	1	0	0	2
13	1	0	0	1	1	3
14	1	0	0	1	0	3
15	1	0	0	0	1	3
16	1	0	0	0	0	3
17	0	1	1	1	1	4
18	0	1	1	1	0	5
19	0	1	1	0	1	4
20	0	1	1	0	0	5
21	0	1	0	1	1	6
22	0	1	0	1	0	6
23	0	1	0	0	1	6
24	0	1	0	0	0	6
25	0	0	1	1	1	4
26	0	0	1	1	0	5
27	0	0	1	0	1	4
28	0	0	1	0	0	5
29	0	0	0	1	1	6
30	0	0	0	1	0	6
31	0	0	0	0	1	6
32	0	0	0	0	0	6

Tabla 4.2: Selección de Setting Group (SG) operando de acuerdo a cada combinación topológica posible (Sin considerar anillos)

4.5. Configuraciones sobre equipos de protección

La presente sección tiene por objetivo explicitar de manera gráfica y directa las configuraciones proporcionadas al dispositivo de protección Multilin F35. Es importante destacar que cualquier configuración y opción interna del dispositivo se modifica mediante el software EnerVista presente en el laboratorio. Sin embargo, una vez hecho el cambio es posible cerrar dicho software sin interferir la operación normal del dispositivo.

4.5.1. Establecimiento de comunicación

En primera instancia, se establece la comunicación con el Relé F35 para poder realizar las próximas configuraciones. Para realizar esto, en el software EnerVista se abre la ventana *Device Setup*. En ella se procede a seleccionar la interface *Ethernet* y se asigna la dirección IP de acuerdo a la Tabla 4.1. Además, se selecciona el nombre del dispositivo y el *Order Code* disponible en la placa física presente en la parte superior del dispositivo.

La figura 4.9 ilustra las configuraciones básicas dadas sobre el dispositivo en materia de comunicación expresadas en el párrafo anterior.

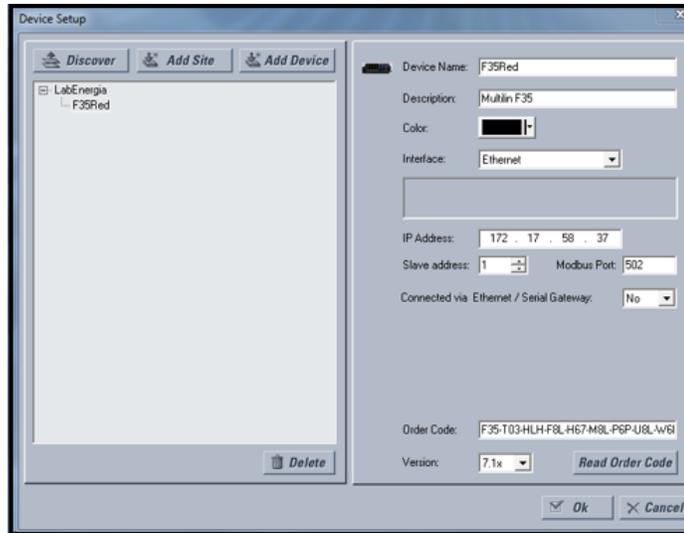


Figura 4.9: Configuración para establecer comunicación con F35

4.5.2. Configuración de funciones lógicas

Para otorgar al relé las instrucciones necesarias para su funcionamiento, se utilizan los elementos *FlexLogic*. Dichos elementos se encuentran basados en lógica Booleana. A continuación, se procede a explicar 3 tipos de bloques lógicos que dominan las acciones del dispositivo F35. Para acceder a su configuración es necesario ir en *Enervista* al menú **Flex-Logic**>**FlexLogic Equation Editor**. Para visualizarlo de mejor manera se recomienda seleccionar la opción *View*, opción disponible bajo el nombre de las columnas.

Los primeros dos tipos de bloque lógico están enfocados a la funcionalidad de la protección, activando los interruptores en caso de que la medición actual supere el umbral de seguridad establecido. El tercer tipo se encuentra enfocado directamente a la activación de los llamados *Settings Groups* que permiten guardar hasta 6 configuraciones predeterminadas en el dispositivo y poder seleccionar una de ellas dependiendo de la situación.

El primer bloque lógico, ilustrado en la figura 4.10, está basado en la protección de curvas instantáneas IOC (Instantaneous OverCurrent) sobre la barra 3. Este bloque combina la señal digital cuando el umbral de corriente ha sido superado con la señal de que el interruptor auxiliar está disponible para operar (*CONT IP 5(W1A)*). Cuando ambas variables se activan, debido a la compuerta lógica *AND*, se activará la salida virtual *VIRT OP 1 (VO1)*.

De manera análoga, este bloque se replica para las barras 4 y 5, asignando sus variables correspondientes y activando las salidas virtuales *Virt OP 2 (VO2)* y *Virt OP 3 (VO3)* respectivamente. Se destaca que este bloque lógico no es frecuentemente utilizado en el laboratorio, dado que se prefiere utilizar las curvas de tiempo inverso TOC.

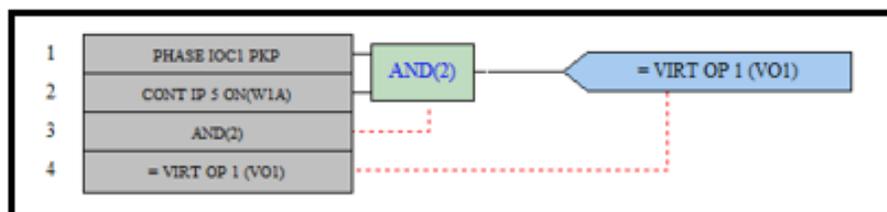


Figura 4.10: Bloque lógico 1. Activación de salida virtual por curva IOC

El segundo bloque lógico presenta la protección por sobrecorriente y baja tensión de manera combinada. Basada en las curvas TOC (Time Over Current) y UV (Under Voltage). En el caso de la barra 3, cuando la señal de superación de corriente umbral es activada (*Phase TOC1*) o el límite de baja de tensión es sobrepasado (*Phase UV1*), se activará la salida virtual llamada (*Virt OP 4(VO4)*). Nuevamente, lo anterior se cumple siempre y cuando el interruptor auxiliar se encuentre listo para operar (*CONT IP 5(W1A)*).

La figura 4.11 hace referencia a esta situación.

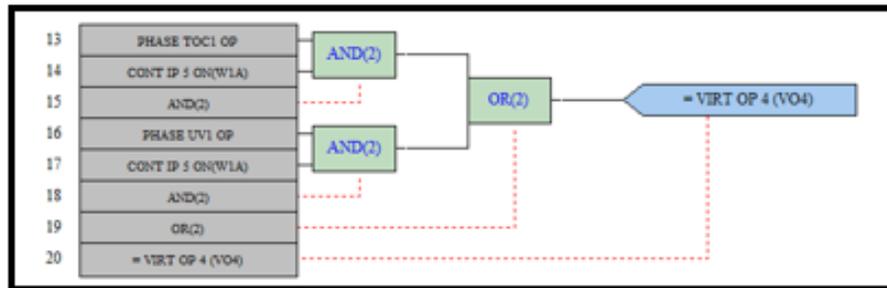


Figura 4.11: Bloque lógico 2. Activación de salida virtual por curvas TOC y UV

El tercer tipo de bloque está diseñado para realizar el cambio de *Setting Group*, ya sea mediante la operación de un botón físico presente en el dispositivo o mediante una entrada virtual, la cual será la señal de disparo generada desde Python que desencadena el cambio de *Setting Group* cuando ciertas condiciones se cumplan (De acuerdo a LUT). Las salidas virtuales asignadas para estas funciones van desde la 7 hasta el número 11. El bloque lógico expresado en este párrafo se expone en la figura 4.12.

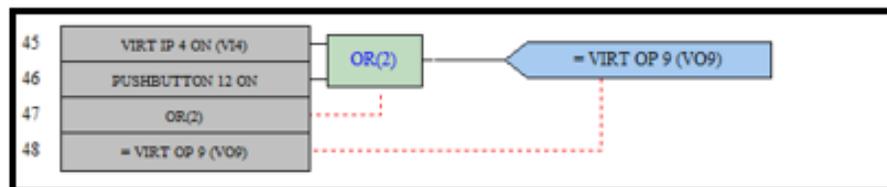


Figura 4.12: Bloque lógico 3. Cambio de Setting Group por botón o input digital

4.5.3. Acciones asociadas a salidas virtuales

Las salidas asociadas a las curvas de protección se encuentran directamente asociadas a la apertura de los interruptores auxiliares. Cuando se detecta y mide una falla, el FlexLogic activa una salida virtual que desencadena la operación de la bobina de disparo. Dicha bobina acciona el interruptor termomagnético correspondiente, despejando la falla y aislando la zona afectada

Concretamente, las salidas Virtual Op 4,5 y 6 están asociadas a los interruptores auxiliares 5,8 y 7 respectivamente. Esto se realiza en el Menú de *Contact Outputs* expuesto a continuación en la figura 4.13.

El resto de salidas están enfocadas al cambio de *Setting Group* operativo. Dicha asociación se realiza en el menú *Settings ->Setting Group*. Este menú se presenta en la figura 4.14.

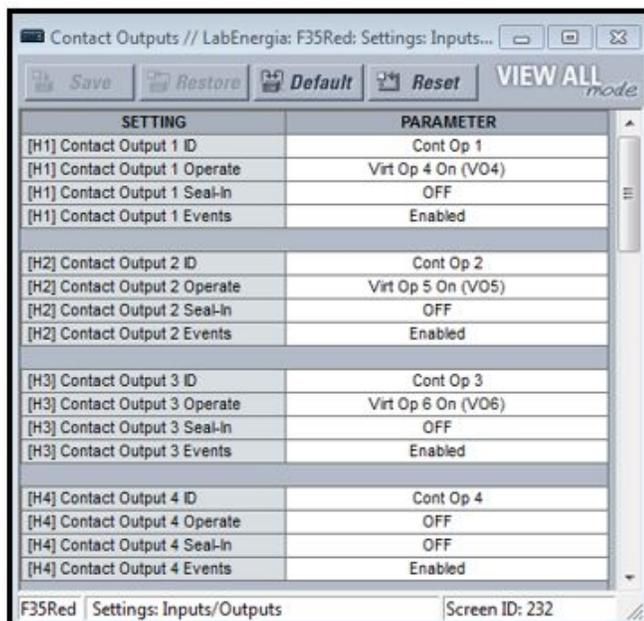


Figura 4.13: Ejecución de instrucciones para salidas virtuales V04 hasta V06

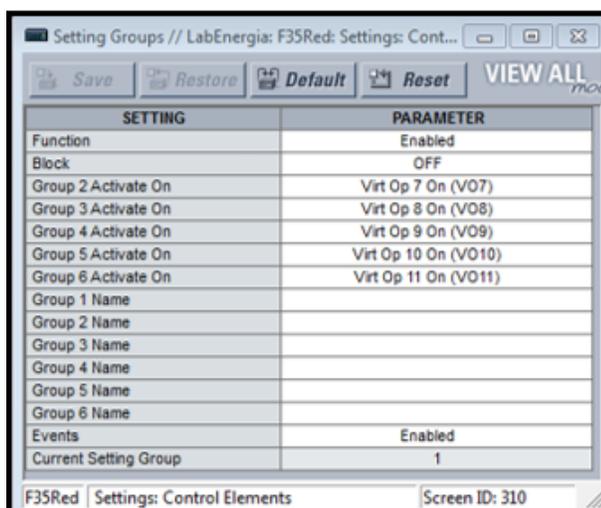


Figura 4.14: Ejecución de instrucciones para salidas virtuales VO7 hasta V011

4.6. Envío y ejecución de instrucciones

El proceso de ejecución de instrucciones se realiza de forma análoga a la lectura. En esta oportunidad el comando "*opc.write*" nos permite realizar la escritura de códigos de dirección MODBUS en cualquiera de las etiquetas disponibles para ello. (Revisar *Memory Map* en [22] para direcciones de escritura de relé F35).

La estructura de ejecución de una instrucción es de la forma *opc.write(etiqueta.Modbus_Adres)*. En la figura 4.5 se aprecia un ejemplo para el caso de apertura/cierre de los contactos de carga presente en cada dispositivo PM750. En este caso, el código 3321 realiza el cierre del contacto permitiendo la energización de la carga mientras que la dirección 3320 abre el cir-

cuito para desconectar la toma de carga correspondiente.

```
1 def OnOff_PM(dispositivo, accion): #cierra contacto ('ON') del PM o lo abre ('OFF'). Dispositivo de la forma 'PM750-X'  
2     if accion=='OFF':  
3         opc.write((dispositivo+'.Comandos.Ejecutar','3320'))  
4     elif accion=='ON':  
5         opc.write((dispositivo+'.Comandos.Ejecutar','3321'))
```

Código 4.5: Código ejemplo de envío de instrucciones Python a OPC servidor

A partir del código 4.5 y utilizando la librería *wxPython*, se ha creado una interfaz de usuario GUI (la cual se despliega automáticamente al ejecutar el programa de Python), que mediante botoneras permite la interacción intuitiva del usuario con la apertura y cierre de las tomas de carga correspondiente a cada PM750. También es posible conocer el estado de apertura de los interruptores auxiliares presentes antes de cada barra. Además, se coloca un visualizador en pantalla para conocer el estado digital del contacto de toma de carga, marcando en rojo si está desconectado o verde en caso contrario. La interfaz de usuario se muestra en la figura 4.15.

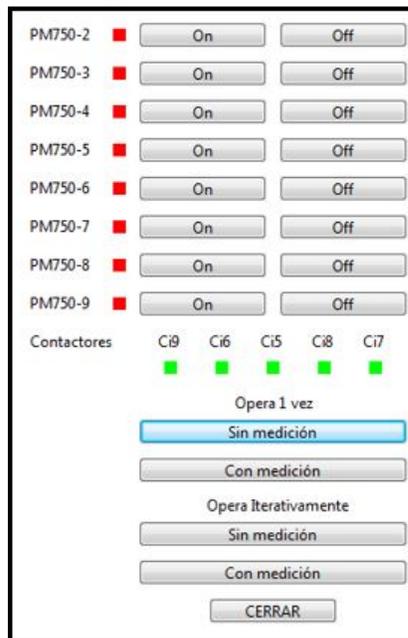


Figura 4.15: Interfaz de usuario GUI

De igual forma, se utiliza el comando *opc.write* para redefinir las curvas de protección del relé F35, es decir, configurar sus parámetros TDM y corriente de pick-up cada vez que sea necesario. Para realizar esto, se define mediante la etiqueta *F35.Comandos.Test* cual es el *Setting Group* con el que se desea establecer la comunicación para la configuración de sus curvas (Existen 6 Setting Groups disponibles). Una vez establecida la comunicación con el grupo requerido, se prosigue a definir los parámetros que rigen la curva de protección de sobrecorriente y de baja tensión. En particular se define el tipo de curva, corriente o voltaje de *Pickup* y el tiempo de retardo en la actuación de la protección.

La verificación de la configuración de las curvas de protección se puede realizar mediante el software especializado del relé F35, **EnerVista**.

Finalmente, se presenta en el Código 4.6 la configuración de la curva de tiempo definido para protección por sobrecorriente.

```

1
2 def Config_TOC(settinggroup, curve, PickUP1pu, PickUP2pu, PickUP3pu, TD1s, TD2s,
3   TD3s):
4     opc.write(('F35.Comandos.Test', str(To_Long(settinggroup-1))))
5     opc.write(('F35.Curvas.TOC1-Source', str(0)))
6     opc.write(('F35.Curvas.TOC2-Source', str(65536)))
7     opc.write(('F35.Curvas.TOC3-Source', str(65536*2)))
8     opc.write(('F35.Curvas.TOC1-Delay', str(int(100*To_Long(float(TD1s))+
9   curve)))) #Escala 0.01 Memory Map Address 5905
10    opc.write(('F35.Curvas.TOC2-Delay', str(int(100*To_Long(float(TD2s))+
11   curve))))
12    opc.write(('F35.Curvas.TOC3-Delay', str(int(100*To_Long(float(TD3s))+
13   curve))))
14    opc.write(('F35.Curvas.TOC1-Pickup', str(int(To_Long(float(PickUP1pu))
15   *1000))))
16    opc.write(('F35.Curvas.TOC2-Pickup', str(int(To_Long(float(PickUP2pu))
17   *1000))))
18    opc.write(('F35.Curvas.TOC3-Pickup', str(int(To_Long(float(PickUP3pu))
19   *1000))))

```

Código 4.6: Módulo para configuración de curva de tiempo definido por sobrecorriente

4.7. Selección y preparación de pruebas

A partir de la implementación expuesta en el presente capítulo, se realizan dos pruebas para constatar la correcta operación del sistema planteado. La primera de las pruebas pretende verificar la operación del programa desarrollado mediante Python de manera exclusiva, es decir, sin la activación de las protecciones por parámetros eléctricos. Esto se realiza mediante la apertura/cierre de interruptores de forma manual, esperando que el programa de Python reconozca esta variación de estado (cambio topológico) y seleccione otro *Setting Group* para operar. La segunda prueba consiste en incluir, al experimento anterior, la activación del sistema de protección debido a la superación de una corriente umbral establecida, lo cual desencadenará la apertura de interruptores, un cambio topológico y la actuación del programa de Python utilizado.

Las siguientes subsecciones presentan el detalle de las pruebas mencionadas, sus antecedentes y forma de medición.

4.7.1. Prueba 1: Identificación de topología y selección de Setting Group

La primera prueba consiste en realizar cambios topológicos controlados mediante la apertura/cierre de los interruptores auxiliares. A partir de esto, se pretende observar cómo el sistema es capaz de reconocer el cambio de estado e identificar la topología imperante de la micro-red para, posteriormente, seleccionar un *Setting Group* adecuado. Lo anterior se realiza mediante el programa de Python expuesto en la sección 7 de Anexos y en base a la Look-up Table (LUT) expuesta en la tabla 4.4.

La figura 4.16 explicita el diagrama de flujo de la prueba descrita. Mediante Python y OPC se estará midiendo periódicamente los parámetros de la micro-red hasta que la apertura/cierre de algún interruptor (interpretado como un cambio topológico) desencadene el resto de instrucciones (Rombo verde de la figura) y ejecute el cambio de *Setting Group*.



Figura 4.16: Diagrama de Flujo de prueba 1

Esta prueba se realiza en dos oportunidades. En primera instancia se utilizará un sistema que no considera el almacenamiento de los datos con SQL. La segunda oportunidad, se considera realizar el almacenamiento de datos mediante SQL.

Para la toma de resultados y verificación de la operación, se utilizará un *Event Recorder*, herramienta incluida en el software *EnerVista* que permite registrar, mediante hora exacta, cualquier evento y ejecución de ordenes lógicas en el relé Multilin F35. Con la medición de los tiempos de actuación se verificará el funcionamiento del sistema de activación de *Setting Group (SG)* y se compararán los tiempos de ejecución de ambas instancias.

4.7.2. Prueba 2: Emulación de falla mediante motor de inducción. Caso: Micro-red conectada a red principal de energía

La segunda prueba consiste en incluir la activación del sistema de protección debido a la superación de la corriente de pick-up (umbral). El experimento se realiza con el fin de verificar la funcionalidad completa del sistema de protección, teniendo por un lado la selección automática del *Setting Group* adecuado (Prueba anterior) y por el otro, la operación misma del sistema de protección, abriendo interruptores al superarse la corriente umbral, para aislar las zonas afectada de la micro-red.

A partir de esta prueba se espera que la micro-red, y en particular el relé F35, detecte la corriente no aceptada y logre aislar de manera rápida la barra acontecida. Una vez que esto sucede, el acto siguiente es que la protección se adapte, detectando el cambio de topología y reconfigurando su *Setting Group* para prepararse ante una nueva falla o emulación de esta. Esta situación se expone en la figura 4.17, donde la elección de Setting Group realizada mediante Python y la actuación del relé F35 son procesos paralelos. Este experimento se realizará 2 veces inmediatamente seguidas, utilizando el motor en la Barra 5 y posteriormente en la Barra 3, verificando que el sistema de protección opera una vez y se prepara para una nueva situación de contingencia en las barras aún disponibles.

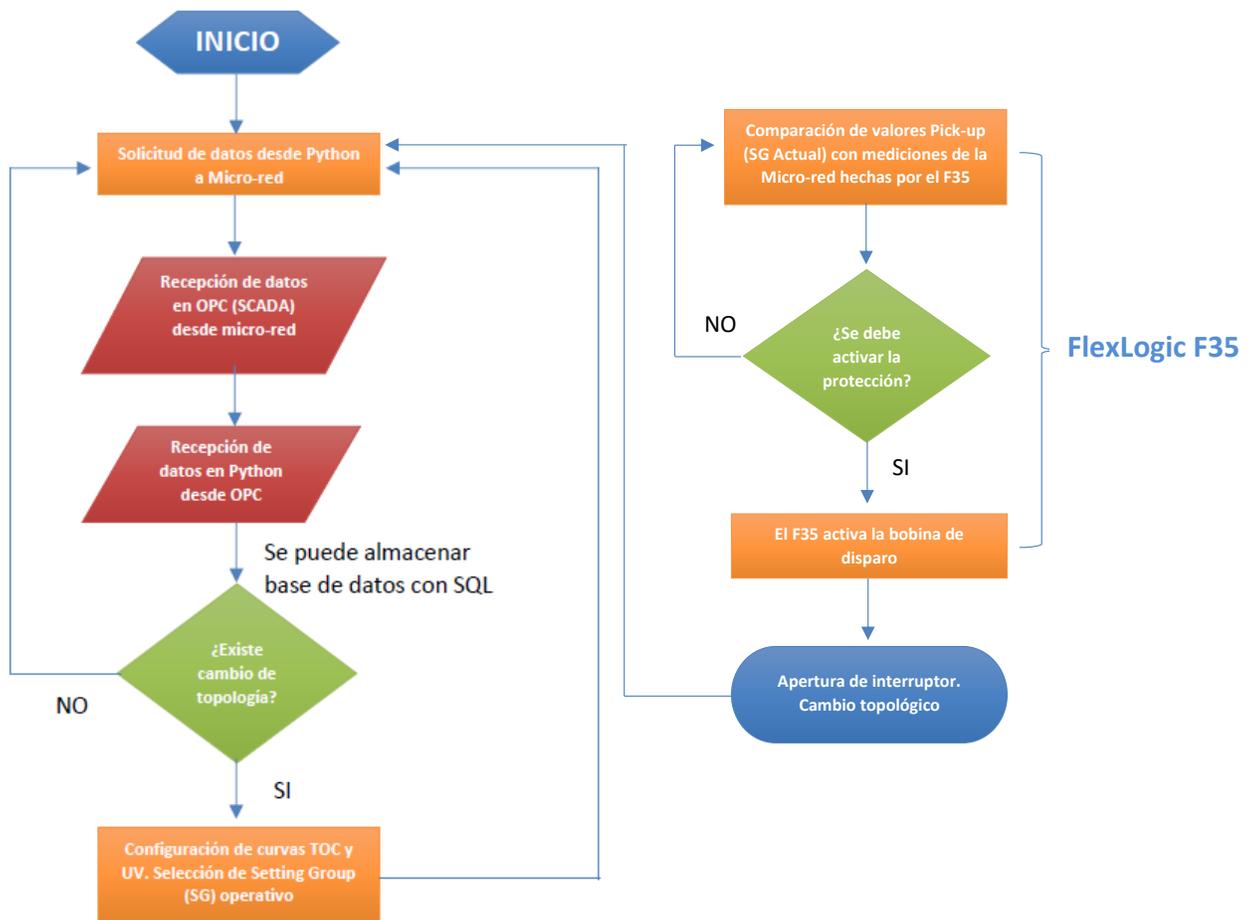


Figura 4.17: Diagrama de Flujo de prueba 2

La corriente de pick-up será superada mediante la partida de un motor de inducción. Durante el encendido del motor en conexión Estrella, se espera que el impulso de corriente obtenga un valor aceptable para la micro-red, la cual es una situación en que la protección no debe activarse. Sin embargo, al realizar la conexión del motor en configuración Delta o Triangulo se espera que la corriente *peak* supere el umbral de corriente definido y la activación de la protección desconecte las barras afectadas.

A través de un estudio de simulación en *Power Factory Digsilent* se prepara esta situación para calibrar las curvas de protección. Se propone emular las fallas con el motor de inducción en las barras 5 y 3.

A continuación, se presenta la simulación realizada y la justificación de los parámetros de las curvas de protección elegidos. En [13], se realiza y estudia una simulación de la Micro-red del laboratorio incorporando la partida del motor de inducción trifásico. Sus características eléctricas se encuentran presentes en la tabla 2.3. Dentro del estudio [13], se obtienen las siguientes curvas de comportamiento para las conexiones tipo Estrella y Delta (Figuras 4.18 y 4.19).

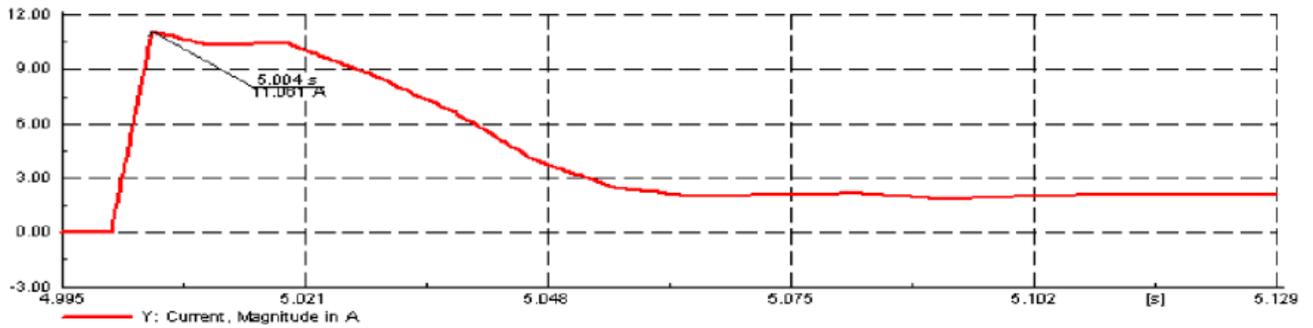


Figura 4.18: Simulación de corriente de encendido motor de inducción en conexión Estrella

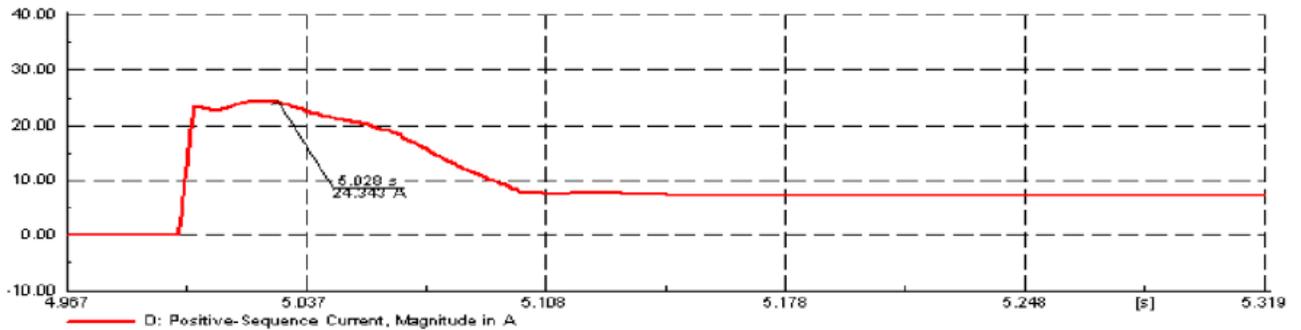


Figura 4.19: Simulación de corriente de encendido motor de inducción en conexión Delta

De las figuras anteriores (4.18 y 4.19) se calculan las curvas de protección de sobrecorriente (TOC) de tal manera que la protección alcance a actuar dentro del tiempo en que permanecen altos niveles de corriente. Particularmente es necesario que se active para 24[A] en Delta y no actúe con los 11[A] que produce la conexión Estrella. Además, se busca que la primera protección en actuar corresponda a la barra 5 y las siguientes protecciones funcionen como respaldo. De esta manera, se configuran las curvas de protección del *Setting Group 1* como indica la tabla 4.3.

Overcurrent Barra 3		Overcurrent Barra 4		Overcurrent Barra 5	
TDM	0.06	TDM	0.04	TDM	0.02
K	0.14	K	0.14	K	0.14
E	0.02	E	0.02	E	0.02
Ipickup	0.25 p.u	Ipickup	0.25 p.u	Ipickup	0.25 p.u

Tabla 4.3: Valores configurados sobre curvas TOC para primera falla de motor de inducción (Barra 5)

Por último, para la emulación de la segunda falla, se utilizará una curva sin retardo de tiempo TDM, de manera que la protección se active instantáneamente con el encendido del motor. De esta forma, los parámetros para el *Setting Group 2* se configuran en los valores de la tabla 4.4.

Overcurrent Barra 3		Overcurrent Barra 4		Overcurrent Barra 5	
TDM	0	TDM	0	TDM	0
K	0.14	K	0.14	K	0.14
E	0.02	E	0.02	E	0.02
Ipickup	0.9 p.u	Ipickup	0.9 p.u	Ipickup	0.9 p.u

Tabla 4.4: Valores configurados sobre curvas TOC para segunda falla de motor de inducción (Barra 3)

A continuación, se presenta en la figura 4.20, los sucesos observables que el sistema debiese realizar. Estos eventos serán comparados con los obtenidos en la sección de resultados para verificar la correcta operación del sistema de protección adaptativo.

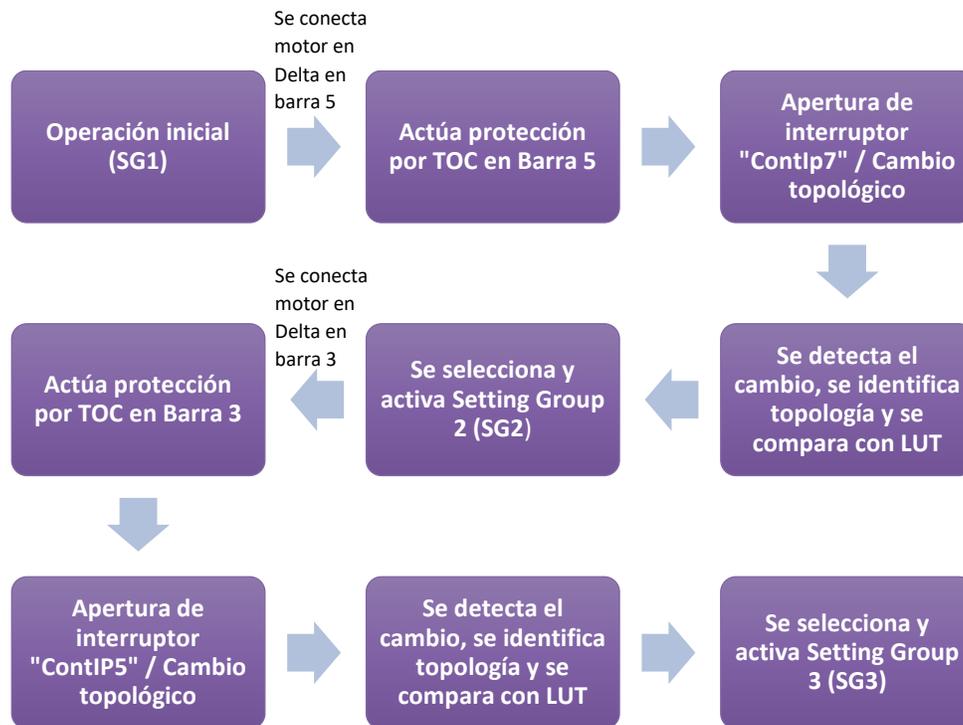


Figura 4.20: Diagrama resumen de sucesos observables durante prueba 2

Capítulo 5

Resultados

5.1. Experiencia de activación de *Setting Group* ante cambio topológico

Los resultados obtenidos con la primera experiencia, de metodología descrita en la sección anterior, se presentan en las figuras 5.1 y 5.2.

5.1.1. Caso 1: Sin almacenamiento de mediciones en SQL.

La figura 5.1 contiene la información de sucesos medibles por el F35 para la prueba descrita en la subsección 4.7.1. Esto consiste en la apertura/cierre de interruptores esperando que el programa detecte cambios topológicos y realice la transición automático de *Setting Group*. Cabe destacar que en esta experiencia no se activa la protección por sobre-sobrecorriente o baja-tensión.

El programa, ante ciertos cambios topológicos específicos (Presentes en el *Lookup Table* de la Tabla 4.4), responde activando el *Setting Group* necesario para satisfacer la contingencia en dicha configuración topológica.

Estos cambios ocurren específicamente cuando los contactos de entrada *Cont Ip 5, 7 y 9* cambian de estado, respondiendo al flujo "*aguas abajo*". De esta forma si *ContIp 5* se encuentra Off (Evento 6), el cambio de *ContIp 7* sería irrelevante debido a que no circula corriente por su barra correspondiente, por lo que el FlexLogic funciona de manera jerárquica.

Device	Event Number	Date/Time	Cause	Data
F35Red	25	Jan 08 2020 14:45:43.811141	SETTING GROUP ACT 4	
F35Red	24	Jan 08 2020 14:45:43.808640	Virt Op 10 Off	
F35Red	23	Jan 08 2020 14:45:43.208640	Virt Op 9 On	
F35Red	22	Jan 08 2020 14:45:35.984054	Cont Ip 7 On	
F35Red	21	Jan 08 2020 14:45:27.211141	SETTING GROUP ACT 5	
F35Red	20	Jan 08 2020 14:45:27.208640	Virt Op 11 Off	
F35Red	19	Jan 08 2020 14:45:26.408650	Virt Op 10 On	
F35Red	18	Jan 08 2020 14:45:22.926121	Cont Ip 8 On	
F35Red	17	Jan 08 2020 14:45:10.746730	Cont Ip 5 On	
F35Red	16	Jan 08 2020 14:44:58.918336	Cont Ip 6 On	
F35Red	15	Jan 08 2020 14:44:53.708640	Virt Op 8 Off	
F35Red	14	Jan 08 2020 14:44:53.311140	SETTING GROUP ACT 6	
F35Red	13	Jan 08 2020 14:44:53.308640	Virt Op 11 On	
F35Red	12	Jan 08 2020 14:44:45.847402	Cont Ip 9 Off	
F35Red	11	Jan 08 2020 14:44:36.469099	Cont Ip 6 Off	
F35Red	10	Jan 08 2020 14:44:36.246140	SETTING GROUP ACT 3	
F35Red	9	Jan 08 2020 14:44:36.243641	Virt Op 8 On	
F35Red	8	Jan 08 2020 14:44:36.243641	SETTING GROUP ACT 1	
F35Red	7	Jan 08 2020 14:44:36.241143	Virt Op 7 Off	
F35Red	6	Jan 08 2020 14:44:27.690308	Cont Ip 5 Off	
F35Red	5	Jan 08 2020 14:44:19.578638	SETTING GROUP ACT 2	
F35Red	4	Jan 08 2020 14:44:19.576142	Virt Op 7 On	
F35Red	3	Jan 08 2020 14:44:17.845489	Cont Ip 8 Off	
F35Red	2	Jan 08 2020 14:44:04.288554	Cont Ip 7 Off	
F35Red	1	Jan 08 2020 14:43:17.041142	EVENTS CLEARED	

Figura 5.1: Event Recorder de operación sin almacenamiento de mediciones en SQLite

Los eventos asociados a la apertura de un interruptor (regidos por LUT de tabla 4.4) y su cambio respectivo de *Setting Group* son:

- **Eventos 2 y 5:** La apertura del ContIP7 (Off) desencadena la activación del *Setting Group 2* (Caso 2 de LUT).
- **Eventos 6 y 10:** La apertura del ContIP5 (Off) desencadena la activación del *Setting Group 3* (Caso 8 de LUT).
- **Eventos 12 y 14:** La apertura de ContIp9 (Off), sumado a los ya abiertos ContIP7 y ContIP5 activa el *Setting Group 6* (Caso 32 de LUT).
- **Eventos 17 y 21:** Al cerrar ContIP5 (ON), manteniendo abierto ContIP9 y ContIP7, activa el SG 5 (Caso 18 de LUT).
- **Eventos 17 y 21:** Al cerrar ContIP7 (ON) , manteniendo abierto ContIP9, activa el SG 4 (Caso 17 de LUT).

Es posible constatar los tiempos de respuesta del sistema. Se medirá el tiempo entre que un contacto de entrada relevante cambia de estado (*Cont Ip 5,7 o 9*) y la activación del *Setting Group* correspondiente.

La tabla 5.1 expone el resumen de tiempos medidos, entre lo que se destaca un respuesta máxima de 16.47 [s], una respuesta mínima de 7.46 [s] y un promedio de respuesta en torno a los 11.5 [s].

Eventos	Hora Inicio	Hora Final	Duración [s]
2,3,4,5	14:44:04.29	14:44:19.58	15.29
6,7,8	14:44:27.69	14:44:36.25	8.56
12,13,14	14:44:45.85	14:44:53.31	7.46
17,18,19, 20,21	14:45:10.74	14:45:27.21	16.47
22,23,24,25	14:45:35.98	14:45:45.81	9.83

Tabla 5.1: Tiempos de respuesta ante cambio topológico y selección de Setting Group sin mediciones

Finalmente, se comprueba el correcto funcionamiento lógico de los sistemas, acorde a lo instruido al programa mediante la *Lookup Table* y a la lógica Booleana perteneciente al FlexLogic del dispositivo Multilin F35.

5.1.2. Caso 2: Con almacenamiento de mediciones en SQL.

De forma similar al experimento del caso 1, se obtienen los eventos registrados por el software *EnerVista*, constatando las acciones y tiempos de actuación del sistema. Estos se presentan en la Figura 5.2.

Device	Event Number	Date/Time	Cause	Data
F35Red	25	Jan 08 2020 14:53:37.280005	Cont Ip 9 On	
F35Red	24	Jan 08 2020 14:53:08.546141	SETTING GROUPACT 4	
F35Red	23	Jan 08 2020 14:53:08.543640	Virt Op 10 Off	
F35Red	22	Jan 08 2020 14:53:07.943641	Virt Op 9 On	
F35Red	21	Jan 08 2020 14:52:58.054247	Cont Ip 7 On	
F35Red	20	Jan 08 2020 14:52:37.928637	SETTING GROUPACT 5	
F35Red	19	Jan 08 2020 14:52:37.926140	Virt Op 11 Off	
F35Red	18	Jan 08 2020 14:52:37.126141	Virt Op 10 On	
F35Red	17	Jan 08 2020 14:52:31.580394	Cont Ip 8 On	
F35Red	16	Jan 08 2020 14:52:13.330814	Cont Ip 5 On	
F35Red	15	Jan 08 2020 14:51:49.712547	Cont Ip 6 On	
F35Red	14	Jan 08 2020 14:51:38.908640	Virt Op 8 Off	
F35Red	13	Jan 08 2020 14:51:38.511139	SETTING GROUPACT 6	
F35Red	12	Jan 08 2020 14:51:38.508639	Virt Op 11 On	
F35Red	11	Jan 08 2020 14:51:24.344726	Cont Ip 9 Off	
F35Red	10	Jan 08 2020 14:51:08.381719	Cont Ip 6 Off	
F35Red	9	Jan 08 2020 14:51:05.211140	SETTING GROUPACT 3	
F35Red	8	Jan 08 2020 14:51:05.208640	Virt Op 8 On	
F35Red	7	Jan 08 2020 14:51:05.208640	Virt Op 7 Off	
F35Red	6	Jan 08 2020 14:50:43.753422	Cont Ip 5 Off	
F35Red	5	Jan 08 2020 14:50:19.005619	Cont Ip 8 Off	
F35Red	4	Jan 08 2020 14:50:03.628639	SETTING GROUPACT 2	
F35Red	3	Jan 08 2020 14:50:03.626146	Virt Op 7 On	
F35Red	2	Jan 08 2020 14:49:49.913678	Cont Ip 7 Off	
F35Red	1	Jan 08 2020 14:48:23.663638	EVENTS CLEARED	

Figura 5.2: Event Recorder de operación con almacenamiento de mediciones en SQLite

Los eventos asociados a la apertura de un interruptor (regidos por LUT de tabla 4.4) y su cambio respectivo de *Setting Group* son:

- **Eventos 2 y 4:** La apertura del ContIP7 (Off) desencadena la activación del *Setting Group 2* (Caso 2 de LUT).
- **Eventos 6 y 9:** La apertura del ContIP5 (Off) desencadena la activación del *Setting Group 3* (Caso 8 de LUT).
- **Eventos 11 y 13:** La apertura de ContIp9 (Off), sumado a los ya abiertos ContIP7 y ContIP5 activa el *Setting Group 6* (Caso 32 de LUT).
- **Eventos 16 y 20:** Al cerrar ContIP5 (ON), manteniendo abierto ContIP9 y ContIP7, activa el SG 5 (Caso 18 de LUT).
- **Eventos 21 y 24:** Al cerrar ContIP7 (ON) , manteniendo abierto ContIP9, activa el SG 4 (Caso 17 de LUT).

Se aprecia nuevamente la efectividad del sistema implementado, teniendo respuestas directas frente a los cambios topológicos identificados mediante el cambio de estado de los contactos de entrada *Cont Ip 5,7 y 9*.

La tabla 5.2 resume los tiempos de ejecución de las acciones tomadas por el sistema. Nuevamente se considera la identificación del cambio topológico como inicio y el cambio de *Setting Group* como el final de la medición.

Eventos	Hora Inicio	Hora Final	Duración [s]
2,3,4	14:49:49.91	14:50:03.63	13.72
6,7,8,9	14:50:43.75	14:51:05.21	21.46
11,12,13	14:51:24.34	14:51:38.17	13.83
16,17,18, 19,20	14:52:13.33	14:52:37.93	24.6
21,22,23, 24	14:52:56.05	14:53:08.55	12.5

Tabla 5.2: Tiempos de respuesta ante cambio topológico y selección de Setting Group con mediciones

En términos generales, los tiempos de actuación son mayores a los presentados en el Caso 1. Esto se debe directamente al tiempo que tarda el programa en capturar los datos desde el OPC y almacenarlos mediante el código de Python en el archivo de base de datos con SQL.

De manera específica, el mayor tiempo de actuación obtenido es 24.6[s], el menor es 12.5 [s] y un promedio de 17.2 [s] aproximadamente. Entre el caso 1 y 2 , en que se diferencia la toma de mediciones y almacenamiento con SQL, se obtiene una diferencia promedio de 5.7 [s] en los tiempos de actuación.

5.2. Experiencia de falla y protección adaptativa. Caso conectado a la red principal de energía

5.2.1. Primera falla, barra 5

Como se ha explicado en la sección 4.7.2. Esta prueba incluye la activación del sistema de protección cuando la corriente medida en la micro-red supere el valor de Pick-up establecido. Para ello, dos emulaciones de falla son aplicadas consecutivamente en las barras 5 y 3 mediante la conexión de un motor de inducción en Delta. En principio, se presenta el reporte de la primera falla, obtenido por el software **EnerVista** (Ver Figura 5.3).

Fault Report Data:			
Fault Number	19	ActiveSettingGroup	1
Relay Name	Relay-1	EventNumberAtTrigger	11
FirmwareRevision	7.11	FaultType	ABC
DateAndTime	2020/01/08 14:13:03.47	Location	0.0
TriggerOperand	Virt Op 6 On	LocationUnits	km
LineID	Barra5	ShotCount	0

Operated Elements:			
Element 1	PHASE TOC3	Element 3	SRC2 50DD
Element 2	SRC1 50DD	Element 4	SRC3 50DD

Prefault and Fault Phasors:				
Signal	Prefault Magnitude	Prefault Angle	Fault Magnitude	Fault Angle
Ia	0.000 A	0.0 deg	27.392 A	148.9 deg
Ib	0.000 A	0.0 deg	24.965 A	33.6 deg
Ic	0.000 A	0.0 deg	26.092 A	269.1 deg
In	0.000 A	0.0 deg	3.617 A	148.9 deg
Va	235.009 V	359.5 deg	123.098 V	6.5 deg
Vb	235.182 V	239.1 deg	114.010 V	247.9 deg
Vc	233.815 V	119.2 deg	123.530 V	128.8 deg

Figura 5.3: Reporte de primera falla con motor de inducción (Barra 5)

En el reporte de la primera falla, se aprecia que esta se detecta a las 13:03.47 alcanzando un corriente *peak* de 27.39 [A] en la fase A. Además, el reporte indica que la primera curva en actuar corresponde a la PHASE TOC 3, correspondiente a la barra 5, que es lo esperado debido a la conveniente configuración de las curvas TOC para tener un tiempo de activación menor en esta barra.

Además, se presenta en la figura 5.4, la oscilografía de la situación de falla descrita. Capturando un valor entre la activación de la bobina de disparo y la apertura del interruptor termomagnético de 0.030331 [s] (Cuadro amarillo figura 5.4).

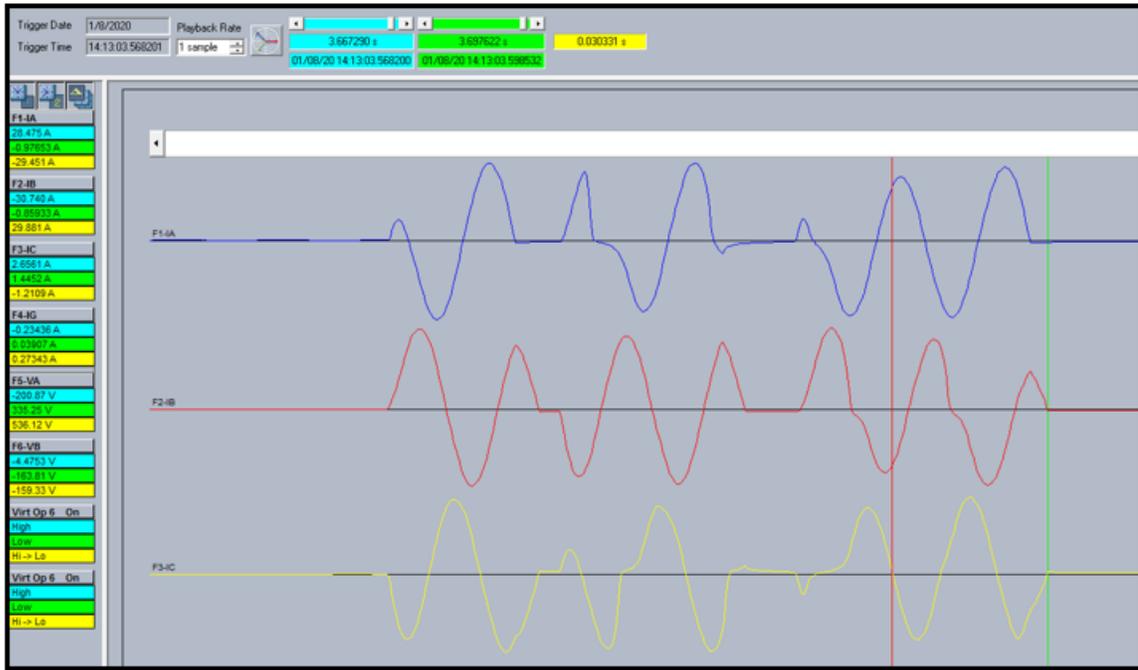


Figura 5.4: Oscilografía de primera falla con motor de inducción (Barra 5)

Una vez que la falla ha sido expuesta, se procede a revisar las acciones que el sistema de protección adaptativo ha realizado. Para esto se utiliza el *Event Recorder* mostrado en la figura 5.5.

Algunos eventos a destacar son:

- **Eventos 2 al 12:** Se observa la actividad de las curvas de protección, detectando la superación (PKP) y normalización (DPO) de los umbrales de corriente *Pickup* establecidos.
- **Evento 13:** La curva TOC3 entra en operación, activando en el evento siguiente (14) la salida virtual 6 a través del FlexLogic del relé.
- **Eventos 15 y 17:** Eventos de toma de muestras del software EnerVista. Tanto la oscilografía como el reporte de falla son capturados en dicho instante.
- **Evento 16:** El primero corresponde al envío de la señal digital de la bobina de disparo que permite la apertura del circuito a través del interruptor auxiliar asociado.
- **Evento 21:** El relé F35 identifica la correcta apertura del interruptor e ingresa la señal digital que desencadenará el cambio de Setting Group a través del programa de Python.
- **Evento 33:** Debido al programa de Python y el FlexLogic del relé, se activa la salida virtual 7, la cual provoca el cambio de Setting Group al grupo 2 (evento 34). Demorando 14 [s] desde detectada la falla hasta el cambio de *Setting Group* preparado para otra contingencia en la nueva topología.

F35Red	34	Jan 08 2020 14:13:17.512289	SETTING GROUPACT 2	
F35Red	33	Jan 08 2020 14:13:17.509790	Virt Op 7 On	
F35Red	32	Jan 08 2020 14:13:03.618246	PHASE TOC3 DPO C	
F35Red	31	Jan 08 2020 14:13:03.618246	PHASE TOC3 DPO B	
F35Red	30	Jan 08 2020 14:13:03.618246	PHASE TOC3 DPO A	
F35Red	29	Jan 08 2020 14:13:03.618246	PHASE TOC2 DPO C	
F35Red	28	Jan 08 2020 14:13:03.618246	PHASE TOC2 DPO B	
F35Red	27	Jan 08 2020 14:13:03.618246	PHASE TOC2 DPO A	
F35Red	26	Jan 08 2020 14:13:03.618246	PHASE TOC1 DPO C	
F35Red	25	Jan 08 2020 14:13:03.618246	PHASE TOC1 DPO B	
F35Red	24	Jan 08 2020 14:13:03.618246	PHASE TOC1 DPO A	
F35Red	23	Jan 08 2020 14:13:03.598219	Cont Op 3 Off	
F35Red	22	Jan 08 2020 14:13:03.598219	Virt Op 6 Off	
F35Red	21	Jan 08 2020 14:13:03.595062	Cont Ip 7 Off	
F35Red	20	Jan 08 2020 14:13:03.578204	PHASE TOC3 OP C	
F35Red	19	Jan 08 2020 14:13:03.578204	PHASE TOC3 PKPA	
F35Red	18	Jan 08 2020 14:13:03.578204	PHASE TOC2 PKPA	
F35Red	17	Jan 08 2020 14:13:03.570702	FAULT RPT TRIG	
F35Red	16	Jan 08 2020 14:13:03.568201	Cont Op 3 On	
F35Red	15	Jan 08 2020 14:13:03.568201	OSCILLOGRAPHY TRIG'D	
F35Red	14	Jan 08 2020 14:13:03.568201	Virt Op 6 On	
F35Red	13	Jan 08 2020 14:13:03.568201	PHASE TOC3 OP B	
F35Red	12	Jan 08 2020 14:13:03.558191	PHASE TOC3 DPO A	
F35Red	11	Jan 08 2020 14:13:03.558191	PHASE TOC2 DPO A	
F35Red	10	Jan 08 2020 14:13:03.498118	PHASE TOC3 PKPA	
F35Red	9	Jan 08 2020 14:13:03.498118	PHASE TOC2 PKPA	
F35Red	8	Jan 08 2020 14:13:03.498118	PHASE TOC1 PKPA	
F35Red	7	Jan 08 2020 14:13:03.488117	PHASE TOC3 PKP C	
F35Red	6	Jan 08 2020 14:13:03.488117	PHASE TOC3 PKP B	
F35Red	5	Jan 08 2020 14:13:03.488117	PHASE TOC2 PKP C	
F35Red	4	Jan 08 2020 14:13:03.488117	PHASE TOC2 PKP B	
F35Red	3	Jan 08 2020 14:13:03.488117	PHASE TOC1 PKP C	
F35Red	2	Jan 08 2020 14:13:03.488117	PHASE TOC1 PKP B	
F35Red	1	Jan 08 2020 14:12:45.389284	EVENTS CLEARED	

Figura 5.5: Event Recorder de acciones realizadas por protección ante primera falla

5.2.2. Segunda Falla, barra 3

Luego del cambio de *Setting Group* realizado con la activación de la protección en barra 5, el motor de inducción se utilizó para provocar una falla en la barra 3, obteniendo el reporte de falla expuesto en la figura 5.6.

Dentro de los datos obtenidos, es importante destacar la hora de la falla, 14:14:25, casi un minuto y medio después de la primera falla, realizada anteriormente. Se aprecia que la barra 3 es la afectada y que precisamente la curva de sobrecorriente TOC1 (Barra 3) es la que activa el sistema de protección y desconexión del interruptor auxiliar.

Se aprecia una corriente de falla de 41.663[A] muy superior a los valores de operación y los *Pick-up* seleccionados (Ya que 1[p.u] corresponde a 20[A]). En cuanto al voltaje, también se observa una caída de tensión importante durante la falla, lo que implica que la protección de baja tensión también podría activarse en caso de estar disponible.

Fault Report Data:				
Fault Number	20	ActiveSettingGroup	2	
Relay Name	Relay-1	EventNumberAtTrigger	35	
FirmwareRevision	7.11	FaultType	ABC	
DateAndTime	2020/01/08 14:14:25.91	Location	0.0	
TriggerOperand	Virt Op 4 On	LocationUnits	km	
LineID	Barra3	ShotCount	0	
Operated Elements:				
Element 1	PHASE TOC1	Element 3	SETTING GROUP	
Element 2	SRC1 50DD			
Prefault and Fault Phasors:				
Signal	Prefault Magnitude	Prefault Angle	Fault Magnitude	Fault Angle
Ia	0.000 A	0.0 deg	41.663 A	136.8 deg
Ib	0.000 A	0.0 deg	38.330 A	25.3 deg
Ic	0.000 A	0.0 deg	40.652 A	262.6 deg
In	0.000 A	0.0 deg	4.748 A	102.0 deg
Va	231.570 V	360.1 deg	167.486 V	1.5 deg
Vb	232.533 V	239.5 deg	168.745 V	241.1 deg
Vc	230.231 V	119.5 deg	168.635 V	121.9 deg

Figura 5.6: Reporte de segunda falla con motor de inducción (Barra 3)

A continuación se presenta en la figura 5.7, las acciones realizadas por el relé luego de activar el *Setting Group 2*, inmediatamente después de la primera falla.

Device	Event Number	Date/Time	Cause	Data
F35Red	53	Jan 08 2020 14:17:24.906924	Cont Ip 5 On	
F35Red	52	Jan 08 2020 14:14:41.543592	SETTING GROUP ACT 3	
F35Red	51	Jan 08 2020 14:14:41.541093	Virt Op 8 On	
F35Red	50	Jan 08 2020 14:14:41.541093	Virt Op 7 Off	
F35Red	49	Jan 08 2020 14:14:25.980020	PHASE TOC1 DPO C	
F35Red	48	Jan 08 2020 14:14:25.980020	PHASE TOC1 DPO B	
F35Red	47	Jan 08 2020 14:14:25.980020	PHASE TOC1 DPO A	
F35Red	46	Jan 08 2020 14:14:25.962494	Cont Op 1 Off	
F35Red	45	Jan 08 2020 14:14:25.962494	Virt Op 4 Off	
F35Red	44	Jan 08 2020 14:14:25.959510	Cont Ip 5 Off	
F35Red	43	Jan 08 2020 14:14:25.939985	PHASE TOC1 OP A	
F35Red	42	Jan 08 2020 14:14:25.939985	PHASE TOC1 PKPA	
F35Red	41	Jan 08 2020 14:14:25.932473	FAULT RPT TRIG	
F35Red	40	Jan 08 2020 14:14:25.929972	Cont Op 1 On	
F35Red	39	Jan 08 2020 14:14:25.929972	Virt Op 4 On	
F35Red	38	Jan 08 2020 14:14:25.929972	PHASE TOC1 OP C	
F35Red	37	Jan 08 2020 14:14:25.929972	PHASE TOC1 PKP C	
F35Red	36	Jan 08 2020 14:14:25.929972	PHASE TOC1 OP B	
F35Red	35	Jan 08 2020 14:14:25.929972	PHASE TOC1 PKP B	
F35Red	34	Jan 08 2020 14:13:17.512289	SETTING GROUP ACT 2	

Figura 5.7: Event Recorder de acciones realizadas por protección ante segunda falla

Algunos eventos a destacar son:

- **Eventos 36 y 38:** Se aprecia la activación de las curvas de protección de sobre-corriente TOC1 debido a la superación de la corriente de *Peak-up*.
- **Evento 39:** El sistema de FlexLogic activa la salida virtual 4.
- **Evento 40:** Se activa la bobina de disparo, realizando la apertura del circuito en el interruptor auxiliar de la línea de transmisión que une las barras 1 y 3, aislando la

barra 3 del resto del sistema.

- **Evento 44:** El relé detecta el estado de apertura del interruptor y comienza su identificación de cambio de topología.
- **Evento 51 y 52:** El sistema ejecuta mediante FlexLogic la salida virtual 8 que desencadena la activación del *Setting Group* 3 (Evento 52). Demorando 16 [s] desde detectada la falla hasta el cambio de *Setting Group* preparado para otra contingencia en la nueva topología.

Finalmente se comprueba que el sistema implementado cumple el flujo de acciones buscado y expuesto en la figura 4.20. Esto verifica como mediante la LUT 4.4 es posible dar instrucciones al sistema de protecciones para que este cambie su operación de acuerdo con las variables de entrada que se requieran, en este caso, el estado de apertura/cierre de los interruptores auxiliares *ContIP* 5,7 y 9.

Capítulo 6

Conclusiones

En primer lugar, se comprueba el cumplimiento de los objetivos y alcances planteados en el capítulo de Introducción. Se logra el objetivo de desarrollar y mejorar el sistema de protecciones del Laboratorio de Energía y Accionamientos de la universidad, preparándolo ante situaciones de contingencias que produzcan sobrecorriente o baja de tensión. Esto se realiza mediante una incorporación de herramientas computacionales basadas en lenguaje Python, permitiendo a la micro-red tomar mediciones y almacenarlas en archivos con SQLite, conocer el estado de apertura de las tomas de carga e interruptores auxiliares, control intuitivo de acciones mediante botoneras, cambio automático de *Setting Group* (dependiendo de la topología imperante) y definición de parámetros de las curvas de sobrecorriente y baja tensión del dispositivo Multilin F35.

De manera específica, se realizó la configuración interna y lógica del dispositivo de protección Multilin F35, basados en los elementos FlexLogic, logrando incluir nuevas funcionalidades e interfaces útiles para las futuras experiencias. Se asignaron salidas virtuales propias para cada acción requerida, teniendo algunas enfocadas en la activación de las protecciones al momento de una falla y otras para el cambio de *Setting Group*, ya sea de forma manual (utilizando los botones del dispositivo) o automática mediante Python.

Además, se genera un programa de Python capaz de solicitar datos medibles desde un servidor OPC, compararlos con un *Lookup Table* (Tabla 4.4) y ejecutar instrucciones en base a esa comparación, detectando las aperturas de los interruptores y seleccionando un *Setting Group* adecuado para dicha topología.

Mediante el presente informe se genera una metodología replicable respecto al trabajo realizado, de modo de ser una ayuda o guía frente a trabajos futuros en protecciones o posibles modificaciones sobre la micro-red. Específicamente, se incorpora la inscripción de elementos y etiquetas a los servidores OPC, configuraciones básicas utilizadas, bloques esenciales del programa que comanda las operaciones y estrategia lógica utilizada.

Respecto a los resultados obtenidos, se comprueba que la ejecución completa del programa funciona de manera correcta. En las figuras 5.1 y 5.2 se constata como el sistema de pro-

tecciones responde frente a cambios topológicos detectados mediante entradas de medición digital. Se comprueba además que los tiempos son menores cuando el sistema no almacena datos mediante SQLite y sólo realiza la selección del *Setting Group* activo, obteniendo un tiempo medio de 11.5[s] de ejecución.

A partir de la prueba con el motor de inducción como simulador de falla, se comprueba que el sistema se encuentra preparado para adaptarse frente a fallas consecutivas, detectando cambios topológicos claves y seleccionando un *Setting Group* adecuado para su configuración. Se obtiene una adaptación de la protección en 14[s] desde la primera falla y 16[s] desde la segunda.

Como trabajo futuro se propone:

- Replicar la experiencia de protecciones descrita en el presente informe en modo de operación aislado de la red principal de energía. Esto supone la incorporación de elementos distribuidos y la apertura del interruptor auxiliar previo a la Barra 1 (Reconocido como ContIP por el programa Python y EnerVista). Con esta prueba se espera la utilización de los *Setting Group* 4,5 y 6 los cuales quedaron dedicados para este caso de operación.
- Realizar la incorporación de los dispositivos de protección ABB, para tener la posibilidad de controlar también las barras 1 y 6 del sistema. Esto puede realizarse mediante un servidor OPC que integre todos los protocolos de los dispositivos y repitiendo la metodología del actual trabajo.
- En base al punto anterior, se propone complementar el sistema actual configurando protecciones que respondan a topologías de tipo anillo con alta penetración de recursos distribuidos presentes en el laboratorio (Paneles solares, aerogenerador, almacenamiento).
- Se propone además el trabajo en una protección de tiempo real, apoyado en la herramienta de configuración continua en las curvas de protección incorporado en este trabajo de título. Dicho avance deberá utilizar las mediciones capturadas, calcular corrientes de cortocircuito y configurar las curvas de protección TOC y UV a cada instante.

Finalmente, es importante destacar el valor del presente trabajo de título existente en la creación, desarrollo y prueba de una estrategia lógica y de herramientas para lograr un sistema de protecciones avanzado, que permita al Departamento de Ingeniería Eléctrica expandir sus horizontes de investigación y docentes en la micro-red del laboratorio, de una forma segura y controlada.

Bibliografía

- [1] Alexandre Oudalov and Antonio Fidigatti. Adaptative network proteccion un microgrids. *International Journal of Distributed Energy Source*, 4:1 – 5, 2009.
- [2] Ben Yu, Junke Guo, Changxin Zhou, Zhiyong Gan, Jinshan Yu, and Fei Lu. A review on microgrid technology with distributed energy. *Proceedings - 2017 International Conference on Smart Grid and Electrical Automation, ICSGEA 2017*, 2017-January:143–146, 2017.
- [3] G. Buigues, A. Dysko, V. Valverde, I. Zamora, and E. Fernández. Microgrid protection: Technical challenges and existing techniques. *Renewable Energy and Power Quality Journal*, 1(11):222–227, 2013.
- [4] Sachit A. Gopalan, Victor Sreeram, and Herbert H.C. Iu. A review of coordination strategies and protection schemes for microgrids. *Renewable and Sustainable Energy Reviews*, 32:222 – 228, 2014.
- [5] H. Al-Nasseri, M. A. Redfern, and F. Li. A voltage based protection for micro-grids containing power electronic converters. In *2006 IEEE Power Engineering Society General Meeting*, pages 7 pp.–, June 2006.
- [6] D. J. Cornforth, A. Berry, and T. Moore. Building a microgrid laboratory. In *8th International Conference on Power Electronics - ECCE Asia*, pages 2035–2042, May 2011.
- [7] B. A. Pacheco, M. A. I. Martins, C. Q. Pica, and N. Rodrigues. A case study of adaptive microgrid protection during transitions and operations. In *2017 Brazilian Power Electronics Conference (COBEP)*, pages 1–5, Nov 2017.
- [8] B. Yu, J. Guo, C. Zhou, Z. Gan, J. Yu, and F. Lu. A review on microgrid technology with distributed energy. In *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pages 143–146, May 2017.
- [9] L. Che, M. E. Khodayar, and M. Shahidehpour. Adaptive protection system for microgrids: Protection practices of a functional microgrid system. *IEEE Electrification Magazine*, 2(1):66–80, March 2014.
- [10] Superintendencia de Electricidad y Combustibles. Instalaciones de Consumo en Baja

Tensión, 2003.

- [11] I. Almutairy. A review of coordination strategies and techniques for overcoming challenges to microgrid protection. In *2016 Saudi Arabia Smart Grid (SASG)*, pages 1–4, Dec 2016.
- [12] K. Sarwagya and P. K. Nayak. An extensive review on the state-of-art on microgrid protection. In *2015 IEEE Power, Communication and Information Technology Conference (PCITC)*, pages 862–866, Oct 2015.
- [13] P. González. *Experiencia Docente de Protecciones para una Micro-red Basada en el Estudio de Funciones de Sobre Corriente y Baja Tensión*. Thesis, Universidad de Chile, 2018.
- [14] J. Shiles, E. Wong, S. Rao, C. Sanden, M. A. Zamani, M. Davari, and F. Katiraei. Microgrid protection: An overview of protection strategies in north american microgrid projects. In *2017 IEEE Power Energy Society General Meeting*, pages 1–5, July 2017.
- [15] M. Meisinger and E. H. Camm. Improved distribution system overcurrent protection using tolerance-based time-current coordination. In *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pages 332–335, March 2019.
- [16] M. Neut. *Infraestructura de Hardware y Software para Desarrollo de Sistemas de Protección en Micro-redes de Baja Tensión*. Thesis, Universidad de Chile, 2016.
- [17] G. Moreno. *Evaluación de Protecciones Adaptativas para una Micro-red*. Thesis, Universidad de Chile, 2017.
- [18] M. Khederzadeh. Adaptive setting of protective relays in microgrids in grid-connected and autonomous operation. In *11th IET International Conference on Developments in Power Systems Protection (DPSP 2012)*, pages 1–4, April 2012.
- [19] F. Katiraei, C. Abbey, S. Tang, and M. Gauthier. Planned islanding on rural feeders — utility perspective. In *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–6, July 2008.
- [20] E. Alegria, T. Brown, E. Minear, and R. H. Lasseter. Certs microgrid demonstration with large-scale energy storage and renewable generation. *IEEE Transactions on Smart Grid*, 5(2):937–943, March 2014.
- [21] G E Digital Energy. F35 Multiple Feeder Protection System, Instruction Manual.
- [22] G E Digital Energy. UR Series Communications Guide.

Capítulo 7

Anexos

7.1. Direcciones MODBUS utilizados

7.1.1. Multiple Feeder System F35 -UR Series

Label	DEC	HEX
Setting Group for Modbus Comms	24432	5F70
Contact Inputs	5376	1500
Virtual Input 1	1024	400
Virtual Input 2	1025	401
Virtual Input 3	1026	402
Virtual Input 4	1027	403
Virtual Input 5	1028	404
Virtual Input 6	1029	405
Virtual Input 7	1030	406

Tabla 7.1: Resumen de direcciones MODBUS comunes para Multilin F35

Label	Source 1 (Barra 3)		Source 2 (Barra 4)		Source 3 (Barra 5)	
	DEC	HEX	DEC	HEX	DEC	HEX
Vag RMS	6656	1A00	6720	1A40	6784	1A80
Vbg RMS	6658	1A02	6722	1A42	6786	1A82
Vcg RMS	6660	1A04	6724	1A44	6788	1A84
Vab RMS	6671	1A0F	6735	1A4F	6799	1A8F
Vbc RMS	6673	1A11	6737	1A51	6801	1A91
Vca RMS	6675	1A13	6739	1A53	6803	1A93
V-1 Magnitude	6694	1A26	6758	1A66	6822	1AA6
Ia RMS	6144	1800	6208	1840	6272	1880
Ib RMS	6146	1802	6210	1842	6274	1882
Ic RMS	6148	1804	6212	1844	6276	1884
Phase IOC Function	23040	5A00	23059	5A13	23078	5A26
Phase IOC Pickup	23042	5A02	23061	5A15	23080	5A28
Phase IOC Source	23041	5A01	23060	5A14	23079	5A27
Phase IOC Delay	23043	5A03	23062	5A16	23081	5A29
Phase TOC Function	22784	5900	22804	5914	22824	5928
Phase TOC Pickup	22787	5903	22807	5917	22827	592B
Phase TOC Curve	22788	5904	22808	5918	22828	592C
Phase TOC TD multiplier	22789	5905	22809	5919	22829	592D
Phase TOC Source	22785	5901	22805	5915	22825	5929
Phase UV Function	28672	7000	28689	7011	28706	7022
Phase UV Pickup	28674	7002	28691	7013	28708	7024
Phase UV Curve	28675	7003	28692	7014	28709	7025
Phase UV Delay	28676	7004	28693	7015	28710	7026
Phase UV Minimun Voltage	28677	7005	28694	7016	28711	7027
Phase UV source	28673	7001	28690	7012	28707	7023

Tabla 7.2: Resumen de direcciones MODBUS utilizadas en Multilin F35 por fuente

7.1.2. Power Logic ION8600

Label	Adress
Vin a	40011
Vin b	40012
Vin b	40013
Vin avg	40014
Ia	40019
Ib	40020
Ic	40021
Iavg	40022
Freq	40025
P. Phase a	40027
P. Phase b	40029
P. Phase c	40031
P. Total	40033
Q. Phase a	40035
Q. Phase b	40037
Q. Phase c	40039
Q. Total	40041
P.factor a	40051
P.factor b	40053
P.factor c	40055

Tabla 7.3: Resumen de direcciones MODBUS utilizadas para ION8600

7.1.3. Power Logic Power Meter PM750

Label	Adress (DEC)
Current-F1	4020
Current-F2	4021
Current-F3	4022
Current-N	4023
Current Avg.	4012
THD I-F1	4045
THD I-F2	4046
THD I-F3	4047
Voltage F1-N	4033
Voltage F2-N	4034
Voltage F3-N	4035
Voltage FN Avg.	4011
THD V-F1	4049
THD V-F2	4050
THD V-F3	4051
Voltage F1-2	4030
Voltage F2-3	4031
Voltage F3-1	4032
Voltage FF Avg.	4010
THD V-F12	4052
THD V-F23	4053
THD V-F31	4054
P-F1	4036
P-F2	4037
P-F3	4038
P-Total	4006
Q-F1	4042
Q-F2	4043
Q-F3	4044
Q-Total	4008
FP	4009
Frecuency	4013
Estado Digital 1	4114
Ejecutar	4126

Tabla 7.4: Resumen de direcciones MODBUS utilizadas para dispositivos PM750

7.2. Código Fuente de Python

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Sep 14 19:44:13 2019
4 @author: nicol
5 """
6 import OpenOPC #library to read data from OPC server
7 import struct
8 import MySQL as sql
9 import wx
10
11
12 #OPENOPC VARIABLE DEFINITIONS
13 gateway = '172.17.58.28' #server computer IPv4
14 opchost = '172.17.58.28' #server computer IPv4
15 opcserv = 'INAT TcpIpH1 OPC Server' #server name (use 'python opc.py -i'
16   or 'python opc.py -q' in cmd line to find)
17 opc = OpenOPC.open_client(gateway)
18 opc.connect(opcserv, opchost)
19
20 #INITIATING OPENOPC CONNECTION
21 def leer_PM(grupoOPC):
22     pretags=opc.list(grupoOPC)
23     tags=opc.list(grupoOPC+'.'+pretags[0])+opc.list(grupoOPC+'.'+pretags
24 [2]+'.*Estado_Digital*')+opc.list(grupoOPC+'.'+pretags[4]+'.*Ejecutar+'
25 )
26     opc.read(tags,group=grupoOPC)
27     v=opc.read(group=grupoOPC)
28     opc.remove(grupoOPC)
29     return(v)
30
31 def leer_F35():
32     tags=opc.list('F35+'.'+opc.list('F35')[0])+opc.list('F35+'.'+opc.
33 list('F35')[1])
34     opc.read(tags,group='F35')
35     v=opc.read(group='F35')
36     opc.remove('F35')
37     return(v)
38
39 def leer_ION8600():
40     tags=opc.list('ION8600+'.'+opc.list('ION8600')[0])
41     opc.read(tags,group='ION8600')
42     v=opc.read(group='ION8600')
43     opc.remove('ION8600')
44     return(v)
45
46 def topologia():
47     v=opc.read('F35.Mediciones.ContIP')
48     ContIp=bin(v[0]/65536).lstrip('0b')
49     while len(ContIp)!=12:
50         ContIp='0'+ContIp
51     opc.remove('F35')
52     ContIp=(ContIp[1],ContIp[0],ContIp[3],ContIp[2],ContIp[4])
53     return(ContIp)
```

```

52
53 def salidas():
54     PM2=opc.read('PM750-2.Salidas.Estado_Digital-1')[0]
55     PM3=opc.read('PM750-3.Salidas.Estado_Digital-1')[0]
56     PM4=opc.read('PM750-4.Salidas.Estado_Digital-1')[0]
57     PM5=opc.read('PM750-5.Salidas.Estado_Digital-1')[0]
58     PM6=opc.read('PM750-6.Salidas.Estado_Digital-1')[0]
59     PM7=opc.read('PM750-7.Salidas.Estado_Digital-1')[0]
60     PM8=opc.read('PM750-8.Salidas.Estado_Digital-1')[0]
61     PM9=opc.read('PM750-9.Salidas.Estado_Digital-1')[0]
62     ContIp=topologia()
63     global Salidas
64     Salidas=(PM2,PM3,PM4,PM5,PM6,PM7,PM8,PM9,ContIp[0],ContIp[1],
ContIp[2],ContIp[3],ContIp[4])
65     return(Salidas)
66
67 def OnOff_PM(dispositivo,accion): #cierra contacto ('ON') del PM o lo abre
('OFF'). Dispositivo de la forma 'PM750-X'
68     if accion=='OFF':
69         opc.write((dispositivo+'.Comandos.Ejecutar','3320'))
70     elif accion=='ON':
71         opc.write((dispositivo+'.Comandos.Ejecutar','3321'))
72
73 def SG(grupo): #Activa y configura el setting group seleccionado
74     aux=[2,3,4,5,6,7]
75     for i in range(0,6): #Desactiva todos las curvas de protecci n
76         opc.write(('F35.Comandos.Test',str(65536*i)))
77     #     opc.write(('F35.Curvas.TOC1-Function',str(0)))
78     #     opc.write(('F35.Curvas.TOC2-Function',str(0)))
79     #     opc.write(('F35.Curvas.TOC3-Function',str(0)))
80     #     opc.write(('F35.Curvas.UV1-Function',str(0)))
81     #     opc.write(('F35.Curvas.UV2-Function',str(0)))
82     #     opc.write(('F35.Curvas.UV3-Function',str(0)))
83     if grupo==1:
84         for var in aux:
85             opc.write(('F35.Mediciones.VirtIp'+str(var),str(0)))
86     else:
87         aux.remove(grupo)
88         aux.remove(grupo+1)
89         opc.write(('F35.Mediciones.VirtIp'+str(grupo),str(65536)))
90         for var in aux:
91             opc.write(('F35.Mediciones.VirtIp'+str(var),str(0)))
92     #     Config_TOC(grupo,3,0.9,0.9,0.9,0,0,0)
93     opc.write(('F35.Comandos.Test',str((grupo-1)*65536)))
94     opc.write(('F35.Curvas.TOC1-Function',str(65536)))
95     opc.write(('F35.Curvas.TOC2-Function',str(65536)))
96     opc.write(('F35.Curvas.TOC3-Function',str(65536)))
97     #     opc.write(('F35.Curvas.UV1-Function',str(65536)))
98     #     opc.write(('F35.Curvas.UV2-Function',str(65536)))
99     #     opc.write(('F35.Curvas.UV3-Function',str(65536)))
100
101
102
103 def To_Long(x):
104     return(x*65536)
105

```

```

106 def Config_TOC(settinggroup, curve, PickUP1pu, PickUP2pu, PickUP3pu, TD1s, TD2s,
    TD3s):
107     opc.write(('F35.Comandos.Test', str(To_Long(settinggroup-1))))
108     opc.write(('F35.Curvas.TOC1-Source', str(0)))
109     opc.write(('F35.Curvas.TOC2-Source', str(65536)))
110     opc.write(('F35.Curvas.TOC3-Source', str(65536*2)))
111     opc.write(('F35.Curvas.TOC1-Delay', str(int(100*To_Long(float(TD1s))+
    curve)))) #Escala 0.01 Memory Map Address 5905
112     opc.write(('F35.Curvas.TOC2-Delay', str(int(100*To_Long(float(TD2s))+
    curve))))
113     opc.write(('F35.Curvas.TOC3-Delay', str(int(100*To_Long(float(TD3s))+
    curve))))
114     opc.write(('F35.Curvas.TOC1-Pickup', str(int(To_Long(float(PickUP1pu))
    *1000)))
115     opc.write(('F35.Curvas.TOC2-Pickup', str(int(To_Long(float(PickUP2pu))
    *1000)))
116     opc.write(('F35.Curvas.TOC3-Pickup', str(int(To_Long(float(PickUP3pu))
    *1000)))
117
118 def Config_IOC(settinggroup, PickUP1pu, PickUP2pu, PickUP3pu, TD1s, TD2s, TD3s):
    #Listo NO CAMBIAR NADA!
119     opc.write(('F35.Comandos.Test', str(To_Long(settinggroup-1))))
120     opc.write(('F35.Curvas.IOC1-Source', str(0)))
121     opc.write(('F35.Curvas.IOC2-Source', str(65536)))
122     opc.write(('F35.Curvas.IOC3-Source', str(65536*2)))
123     opc.write(('F35.Curvas.IOC1-Delay', str((To_Long(float(TD1s))+float(
    PickUP1pu)*10)*10)))
124     opc.write(('F35.Curvas.IOC2-Delay', str((To_Long(float(TD2s))+float(
    PickUP2pu)*10)*10)))
125     opc.write(('F35.Curvas.IOC3-Delay', str((To_Long(float(TD3s))+float(
    PickUP3pu)*10)*10)))
126
127 def Config_UV(settinggroup, curve, PickUP1pu, PickUP2pu, PickUP3pu, TD1s, TD2s,
    TD3s): #Listo NO CAMBIAR
128     opc.write(('F35.Comandos.Test', str(To_Long(settinggroup-1))))
129     opc.write(('F35.Curvas.UV1-Source', str(0)))
130     opc.write(('F35.Curvas.UV2-Source', str(65536)))
131     opc.write(('F35.Curvas.UV3-Source', str(65536*2)))
132     opc.write(('F35.Curvas.UV1-MinV', str(int(To_Long(0.2)*1000+TD1s*100)))
    )
133     opc.write(('F35.Curvas.UV2-MinV', str(int(To_Long(0.2)*1000+TD2s*100)))
    )
134     opc.write(('F35.Curvas.UV3-MinV', str(int(To_Long(0.2)*1000+TD3s*100)))
    )
135     opc.write(('F35.Curvas.UV1-Curve', str(int(To_Long(curve)+PickUP1pu
    *1000)))
136     opc.write(('F35.Curvas.UV2-Curve', str(int(To_Long(curve)+PickUP2pu
    *1000)))
137     opc.write(('F35.Curvas.UV3-Curve', str(int(To_Long(curve)+PickUP3pu
    *1000)))
138
139 #Config_TOC(1,3,0.6,0.6,0.6,1,1.02,1.03)
140 #Config_TOC(4,0,2.5,1.1,1,3,3.2,3.4)
141 #Config_IOC(4,2.1,1.5,0.5,1.1,1.4,2.1)
142 #Config_UV(4,0,1,1.5,0.5,2,2.4,2.7)
143

```

```

144 ##### Ejecuci n de modulos#####
145 def CambiaTop(tipo,iterativo,contactual):
146     v=opc.read('F35.Mediciones.ContIP')
147     print(contactual)
148     if v==contactual:
149         CambiaTop(tipo,iterativo,contactual)
150     else:
151         contactual=v
152         ejecutar(tipo,iterativo,contactual)
153
154 def ejecutar(tipo,iterativo,contactual):
155     #Tipo define si el programa se ejecuta de forma minimalista (eleccion
156     #de setting group) o si adem s se tomar n las mediciones
157     #iterativo se utilizar para determinar si el programa se ejecuta
158     #constantemente sin fin, solo una vez
159     #Recopilaci n de todos los datos a la vez
160
161     if tipo=='Mediciones':
162         con=sql.sql_connection() #abre archivo sql 'Microrred.db', Si no
163         existe crea uno (cambios en MiSQL.py)
164         sql.sql_table(con) #genera las tablas del archivo
165         Datos_PM2=leer_PM('PM750-2')
166         for var in Datos_PM2:
167             sql.sql_insert(con,var,'PM2')
168
169         Datos_PM3=leer_PM('PM750-3')
170         for var in Datos_PM3:
171             sql.sql_insert(con,var,'PM3')
172
173         Datos_PM4=leer_PM('PM750-4')
174         for var in Datos_PM4:
175             sql.sql_insert(con,var,'PM4')
176
177         Datos_PM5=leer_PM('PM750-5')
178         for var in Datos_PM5:
179             sql.sql_insert(con,var,'PM5')
180
181         Datos_PM6=leer_PM('PM750-6')
182         for var in Datos_PM6:
183             sql.sql_insert(con,var,'PM6')
184
185         Datos_PM7=leer_PM('PM750-7')
186         for var in Datos_PM7:
187             sql.sql_insert(con,var,'PM7')
188
189         Datos_PM8=leer_PM('PM750-8')
190         for var in Datos_PM8:
191             sql.sql_insert(con,var,'PM8')
192
193         Datos_PM9=leer_PM('PM750-9')
194         for var in Datos_PM9:
195             sql.sql_insert(con,var,'PM9')
196
197         Datos_F35=leer_F35()
198         for var in Datos_F35:
199             x=var[1]

```

```

197         if len(str(x))==10:
198             var=list(var)
199             var[1]=struct.unpack('!f',str(hex(x).lstrip('0x')).decode(
'hex'))[0]
200             var=tuple(var)
201             sql.sql_insert(con,var,'F35')
202
203
204     Datos_ION8600=leer_ION8600()
205     for var in Datos_ION8600:
206         sql.sql_insert(con,var,'ION8600')
207     #End IF 'Medicion'
208 #Ejecuci n de instrucciones LUT
209 Salidas=salidas()
210 if Salidas[12]=='1':
211     if Salidas[10]=='1':
212         if Salidas[11]=='1':
213             SG(1)
214         else:
215             SG(2)
216     else:
217         SG(3)
218 else:
219     if Salidas[10]=='1':
220         if Salidas[11]=='1':
221             SG(4)
222         else:
223             SG(5)
224     else:
225         SG(6)
226
227
228
229 if iterativo=='True':
230     CambiaTop(tipo,'True',contactual)
231 elif iterativo=='False':
232     print('Programa ejecutado una vez')
233 else:
234     print('No se ha especificado si iterar una o m s veces')
235 return()
236
237
238
239
240 ##### Ventana de Operaciones #####
241 class MiFrame(wx.Frame):
242     def __init__(self,*args,**kwargs):
243         wx.Frame.__init__(self,*args,**kwargs)
244
245     #Texto estatico
246     wx.StaticText(self,-1, u"PM750-2", size=(50,30), pos=(15,12.5))
247     wx.StaticText(self,-1, u"PM750-3", size=(50,30), pos=(15,42.5))
248     wx.StaticText(self,-1, u"PM750-4", size=(50,30), pos=(15,72.5))
249     wx.StaticText(self,-1, u"PM750-5", size=(50,30), pos=(15,102.5))
250     wx.StaticText(self,-1, u"PM750-6", size=(50,30), pos=(15,132.5))
251     wx.StaticText(self,-1, u"PM750-7", size=(50,30), pos=(15,162.5))

```

```

252     wx.StaticText(self,-1, u"PM750-8", size=(50,30), pos=(15,192.5))
253     wx.StaticText(self,-1, u"PM750-9", size=(50,30), pos=(15,222.5))
254     wx.StaticText(self,-1, u"Contactores", size=(70,30), pos
=(15,252.5))
255     wx.StaticText(self,-1, u"Ci9", size=(70,30), pos=(115,252.5))
256     wx.StaticText(self,-1, u"Ci6", size=(70,30), pos=(155,252.5))
257     wx.StaticText(self,-1, u"Ci5", size=(70,30), pos=(195,252.5))
258     wx.StaticText(self,-1, u"Ci8", size=(70,30), pos=(235,252.5))
259     wx.StaticText(self,-1, u"Ci7", size=(70,30), pos=(275,252.5))
260     wx.StaticText(self,-1, u"Opera 1 vez", size=(100,20), pos
=(175,300))
261     wx.StaticText(self,-1, u"Opera Iterativamente", size=(150,20), pos
=(153,380))
262
263     #Indicadores "LED" de color
264     self.Led2= wx.Panel(self,-1,size=(10,10),pos=(80,15))
265     self.Led3= wx.Panel(self,-1,size=(10,10),pos=(80,45))
266     self.Led4= wx.Panel(self,-1,size=(10,10),pos=(80,75))
267     self.Led5= wx.Panel(self,-1,size=(10,10),pos=(80,105))
268     self.Led6= wx.Panel(self,-1,size=(10,10),pos=(80,135))
269     self.Led7= wx.Panel(self,-1,size=(10,10),pos=(80,165))
270     self.Led8= wx.Panel(self,-1,size=(10,10),pos=(80,195))
271     self.Led9= wx.Panel(self,-1,size=(10,10),pos=(80,225))
272     self.Led10= wx.Panel(self,-1,size=(10,10),pos=(160,275))
273     self.Led11= wx.Panel(self,-1,size=(10,10),pos=(200,275))
274     self.Led12= wx.Panel(self,-1,size=(10,10),pos=(240,275))
275     self.Led13= wx.Panel(self,-1,size=(10,10),pos=(280,275))
276     self.Led14= wx.Panel(self,-1,size=(10,10),pos=(120,275))
277
278     #Botones
279     On2= wx.Button(self,-1, u"On", size=(100,20), pos=(100,10))
280     Off2 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,10))
281     On3 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,40))
282     Off3 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,40))
283     On4 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,70))
284     Off4 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,70))
285     On5 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,100))
286     Off5 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,100))
287     On6 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,130))
288     Off6 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,130))
289     On7 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,160))
290     Off7 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,160))
291     On8 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,190))
292     Off8 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,190))
293     On9 = wx.Button(self,-1, u"On", size=(100,20), pos=(100,220))
294     Off9 = wx.Button(self,-1, u"Off", size=(100,20), pos=(210,220))
295
296
297     Actualizar=wx.Button(self,-1, u"Sin medici n", size=(210,20),
pos=(100,320))
298     ActualizarM=wx.Button(self,-1, u"Con medici n", size=(210,20),
pos=(100,350))
299
300     iOperar=wx.Button(self,-1, u"Sin medici n", size=(210,20), pos
=(100,400))
301     iEjecutar = wx.Button(self,-1, u"Con medici n", size=(210,20),

```

```

pos=(100,430))
302     Cerrar = wx.Button(self, -1, u"CERRAR", size=(100,20), pos
=(155,460))
303
304     # Conectando el evento de un bot n
305     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On2,On2)
306     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off2,Off2)
307     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On3,On3)
308     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off3,Off3)
309     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On4,On4)
310     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off4,Off4)
311     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On5,On5)
312     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off5,Off5)
313     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On6,On6)
314     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off6,Off6)
315     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On7,On7)
316     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off7,Off7)
317     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On8,On8)
318     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off8,Off8)
319     self.Bind(wx.EVT_BUTTON, self.OnClickButton_On9,On9)
320     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Off9,Off9)
321
322     self.Bind(wx.EVT_BUTTON, self.OnClickButton_Actualizar,Actualizar)
323     self.Bind(wx.EVT_BUTTON, self.OnClickButton_ActualizarM,
ActualizarM)
324
325     self.Bind(wx.EVT_BUTTON, self.OnClickButtoni,iOperar)
326     self.Bind(wx.EVT_BUTTON, self.OnClickButtoni2,iEjecutar)
327
328     self.Bind(wx.EVT_BUTTON, self.cerrar,Cerrar)
329
330     # Mostrando la interfaz
331     self.SetBackgroundColour(wx.Colour(255,255,255))
332
333     self.Show()
334
335
336     def OnClickButton_On2(self, event):
337         OnOff_PM('PM750-2', 'ON')
338         self.Refresh()
339     def OnClickButton_Off2(self, event):
340         OnOff_PM('PM750-2', 'OFF')
341         self.Refresh()
342     def OnClickButton_On3(self, event):
343         OnOff_PM('PM750-3', 'ON')
344         self.Refresh()
345     def OnClickButton_Off3(self, event):
346         OnOff_PM('PM750-3', 'OFF')
347         self.Refresh()
348     def OnClickButton_On4(self, event):
349         OnOff_PM('PM750-4', 'ON')
350         self.Refresh()
351     def OnClickButton_Off4(self, event):
352         OnOff_PM('PM750-4', 'OFF')
353         self.Refresh()
354     def OnClickButton_On5(self, event):

```

```

355     OnOff_PM('PM750-5', 'ON')
356     self.Refresh()
357     def OnClickButton_Off5(self, event):
358         OnOff_PM('PM750-5', 'OFF')
359         self.Refresh()
360     def OnClickButton_On6(self, event):
361         OnOff_PM('PM750-6', 'ON')
362         self.Refresh()
363     def OnClickButton_Off6(self, event):
364         OnOff_PM('PM750-6', 'OFF')
365         self.Refresh()
366     def OnClickButton_On7(self, event):
367         OnOff_PM('PM750-7', 'ON')
368         self.Refresh()
369     def OnClickButton_Off7(self, event):
370         OnOff_PM('PM750-7', 'OFF')
371         self.Refresh()
372     def OnClickButton_On8(self, event):
373         OnOff_PM('PM750-8', 'ON')
374         self.Refresh()
375     def OnClickButton_Off8(self, event):
376         OnOff_PM('PM750-8', 'OFF')
377         self.Refresh()
378     def OnClickButton_On9(self, event):
379         OnOff_PM('PM750-9', 'ON')
380         self.Refresh()
381     def OnClickButton_Off9(self, event):
382         OnOff_PM('PM750-9', 'OFF')
383         self.Refresh()
384
385     def OnClickButton_Actualizar(self, event):
386         ejecutar('', 'False')
387
388         if Salidas[0]==1:
389             self.Led2.SetBackgroundColour(wx.Colour(0,255,0))
390         else:
391             self.Led2.SetBackgroundColour(wx.Colour(255,0,0))
392
393         if Salidas[1]==1:
394             self.Led3.SetBackgroundColour(wx.Colour(0,255,0))
395         else:
396             self.Led3.SetBackgroundColour(wx.Colour(255,0,0))
397
398         if Salidas[2]==1:
399             self.Led4.SetBackgroundColour(wx.Colour(0,255,0))
400         else:
401             self.Led4.SetBackgroundColour(wx.Colour(255,0,0))
402
403         if Salidas[3]==1:
404             self.Led5.SetBackgroundColour(wx.Colour(0,255,0))
405         else:
406             self.Led5.SetBackgroundColour(wx.Colour(255,0,0))
407
408         if Salidas[4]==1:
409             self.Led6.SetBackgroundColour(wx.Colour(0,255,0))
410         else:

```

```

411         self.Led6.SetBackgroundColour(wx.Colour(255,0,0))
412
413     if Salidas[5]==1:
414         self.Led7.SetBackgroundColour(wx.Colour(0,255,0))
415     else:
416         self.Led7.SetBackgroundColour(wx.Colour(255,0,0))
417
418     if Salidas[6]==1:
419         self.Led8.SetBackgroundColour(wx.Colour(0,255,0))
420     else:
421         self.Led8.SetBackgroundColour(wx.Colour(255,0,0))
422
423     if Salidas[7]==1:
424         self.Led9.SetBackgroundColour(wx.Colour(0,255,0))
425     else:
426         self.Led9.SetBackgroundColour(wx.Colour(255,0,0))
427
428     if Salidas[8]=='1':
429         self.Led10.SetBackgroundColour(wx.Colour(0,255,0))
430     else:
431         self.Led10.SetBackgroundColour(wx.Colour(255,0,0))
432
433     if Salidas[9]=='1':
434         self.Led11.SetBackgroundColour(wx.Colour(0,255,0))
435     else:
436         self.Led11.SetBackgroundColour(wx.Colour(255,0,0))
437
438     if Salidas[10]=='1':
439         self.Led12.SetBackgroundColour(wx.Colour(0,255,0))
440     else:
441         self.Led12.SetBackgroundColour(wx.Colour(255,0,0))
442
443     if Salidas[11]=='1':
444         self.Led13.SetBackgroundColour(wx.Colour(0,255,0))
445     else:
446         self.Led13.SetBackgroundColour(wx.Colour(255,0,0))
447
448     if Salidas[12]=='1':
449         self.Led14.SetBackgroundColour(wx.Colour(0,255,0))
450     else:
451         self.Led14.SetBackgroundColour(wx.Colour(255,0,0))
452     self.Refresh()
453
454     def OnClickButton_ActualizarM(self, event):
455         ejecutar('Mediciones', 'False')
456
457     if Salidas[0]==1:
458         self.Led2.SetBackgroundColour(wx.Colour(0,255,0))
459     else:
460         self.Led2.SetBackgroundColour(wx.Colour(255,0,0))
461
462     if Salidas[1]==1:
463         self.Led3.SetBackgroundColour(wx.Colour(0,255,0))
464     else:
465         self.Led3.SetBackgroundColour(wx.Colour(255,0,0))
466

```

```

467     if Salidas [2]==1:
468         self.Led4.SetBackgroundColour (wx.Colour (0,255,0))
469     else:
470         self.Led4.SetBackgroundColour (wx.Colour (255,0,0))
471
472     if Salidas [3]==1:
473         self.Led5.SetBackgroundColour (wx.Colour (0,255,0))
474     else:
475         self.Led5.SetBackgroundColour (wx.Colour (255,0,0))
476
477     if Salidas [4]==1:
478         self.Led6.SetBackgroundColour (wx.Colour (0,255,0))
479     else:
480         self.Led6.SetBackgroundColour (wx.Colour (255,0,0))
481
482     if Salidas [5]==1:
483         self.Led7.SetBackgroundColour (wx.Colour (0,255,0))
484     else:
485         self.Led7.SetBackgroundColour (wx.Colour (255,0,0))
486
487     if Salidas [6]==1:
488         self.Led8.SetBackgroundColour (wx.Colour (0,255,0))
489     else:
490         self.Led8.SetBackgroundColour (wx.Colour (255,0,0))
491
492     if Salidas [7]==1:
493         self.Led9.SetBackgroundColour (wx.Colour (0,255,0))
494     else:
495         self.Led9.SetBackgroundColour (wx.Colour (255,0,0))
496
497     if Salidas [8]== '1':
498         self.Led10.SetBackgroundColour (wx.Colour (0,255,0))
499     else:
500         self.Led10.SetBackgroundColour (wx.Colour (255,0,0))
501
502     if Salidas [9]== '1':
503         self.Led11.SetBackgroundColour (wx.Colour (0,255,0))
504     else:
505         self.Led11.SetBackgroundColour (wx.Colour (255,0,0))
506
507     if Salidas [10]== '1':
508         self.Led12.SetBackgroundColour (wx.Colour (0,255,0))
509     else:
510         self.Led12.SetBackgroundColour (wx.Colour (255,0,0))
511
512     if Salidas [11]== '1':
513         self.Led13.SetBackgroundColour (wx.Colour (0,255,0))
514     else:
515         self.Led13.SetBackgroundColour (wx.Colour (255,0,0))
516
517     if Salidas [12]== '1':
518         self.Led14.SetBackgroundColour (wx.Colour (0,255,0))
519     else:
520         self.Led14.SetBackgroundColour (wx.Colour (255,0,0))
521     self.Refresh()
522

```

```

523
524
525
526     def OnClickButton1(self, event):
527         CambiaTop('', 'True', 3)
528
529     def OnClickButton2(self, event):
530         ejecutar('Mediciones', 'True', 3)
531
532     def cerrar(self, event):
533         self.Close()
534
535
536
537 if __name__ == "__main__":
538
539     app = wx.App()
540     fr = MiFrame(None, -1, u"Contactos y Ejecuci n", size=(320,495), style
541 =wx.STAY_ON_TOP)
542     app.MainLoop()
543
544 opc.close()

```

Código 7.1: Programa principal

7.3. Código de SQLite

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Sep 25 18:01:00 2019
4
5 @author: Encargado
6 """
7
8 import sqlite3
9 from sqlite3 import Error
10
11 def sql_connection():
12     try:
13         con=sqlite3.connect('Microrred.db')
14         return con
15     except Error:
16         print(Error)
17
18 def sql_table(con):
19     cursorObj=con.cursor()
20     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM2(Medicion text, Valor
21 real, Estado text, FechaHora text)")
22     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM3(Medicion text, Valor
23 real, Estado text, FechaHora text)")
24     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM4(Medicion text, Valor
25 real, Estado text, FechaHora text)")
26     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM5(Medicion text, Valor
27 real, Estado text, FechaHora text)")
28     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM6(Medicion text, Valor

```

```

25     real, Estado text, FechaHora text)")
26     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM7(Medicion text, Valor
27     real, Estado text, FechaHora text)")
28     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM8(Medicion text, Valor
29     real, Estado text, FechaHora text)")
30     cursorObj.execute("CREATE TABLE IF NOT EXISTS PM9(Medicion text, Valor
31     real, Estado text, FechaHora text)")
32     cursorObj.execute("CREATE TABLE IF NOT EXISTS ION8600(Medicion text,
33     Valor real, Estado text, FechaHora text)")
34     cursorObj.execute("CREATE TABLE IF NOT EXISTS F35(Medicion text, Valor
35     long, Estado text, FechaHora text)")
36     con.commit()
37
38 def sql_insert(con,entities ,table):
39     cursorObj= con.cursor()
40     cursorObj.execute('INSERT INTO '+table+'(Mediccion,Valor,Estado,
41     FechaHora) VALUES(?,?,?,?)',entities)
42     con.commit()

```

Código 7.2: Programa de almacenamiento en SQL