

Análisis de conectividad sobre redes de distribución eléctrica

por

Nadia Mayor

Trabajo final de la carrera de grado de
Ingeniería de Sistemas de la
Universidad Nacional del Centro de la Provincia de Buenos Aires



Director: Dr. Gustavo Boroni
Co-Director: Ing. Matias Antunez

Agradecimientos

Quisiera dedicar este trabajo a todas esas personas que estuvieron presentes durante este largo recorrido brindando todo el apoyo y aliento. En especial a mi familia que desde la distancia siempre estuvieron presente desde los mensajes, llamadas o velitas. A mis amigos, los que estuvieron desde la distancia y los que fueron mi segunda familia en Tandil. En particular, me gustaría dar las gracias a Gustavo Boroni y Matías Antunez que dedicaron su tiempo, ofrecieron sus conocimientos y siempre estuvieron contemplando mis dudas durante el desarrollo de esta tesis. Finalmente, le agradezco al Instituto Pladema que me brindó el espacio para estar cerca de mi director y co-director desarrollando la tesis. ¡A todos ellos muchas gracias!

“Our worst fears lie in anticipation”
Don Draper - Mad Men

Índice general

Agradecimientos.....	1
Resumen.....	8
1. Introducción.....	9
1.1. Motivación.....	9
1.2. Estado del Arte.....	9
1.2.1. Análisis de conectividad de Red (Network Connectivity Analysis-NCA).....	10
1.3. Propuesta.....	11
1.4. Organización.....	12
2. Análisis de conectividad de red	13
2.1. Modelo CIM.....	13
2.2. RDF	14
2.3. JENA.....	16
2.4. RDF/XML.....	17
2.5. Grafos.....	18
2.5.1. Algoritmo de búsqueda DFS (Depth-First Search).....	19
2.6. Estado de los interruptores.....	21
2.6.1. Configuración de los interruptores.....	22
2.7. Topología de la red.....	23
2.7.1. Conceptos básicos.....	23
2.7.1.1. Zonas desenergizadas	23
2.7.1.2. Lazos.....	24
2.7.1.3. Paralelos	26
2.7.2. Las diferentes topologías.....	27
2.7.2.1. Red radial.....	27
2.7.2.2. Red en lazo o malla	28
2.7.2.3. Red Lazo-Radial o mixto.....	29
2.8. Conclusión	30
3. Diseño e implementación.....	32
3.1. Modificación del modelo CIM desde el Editor	32
3.2. Creación de grafos a partir del modelo CIM	33
3.2.1. Creación de grafos basado en el patrón Abstract Factory.....	35
3.2.2. Creación del grafo completo	37
3.2.3. Creación del grafo reducido	38
3.3. Utilización del patrón Strategy para búsquedas sobre el grafo reducido.....	40

3.3.1. Búsqueda de zonas desenergizadas.....	42
3.3.2. Búsqueda de lazos.....	44
3.3.3. Búsqueda de paralelos.....	48
3.4. Conversión de los resultados al grafo original.....	52
3.5. Ejecución de todas las etapas del NCA.....	53
3.6. Conclusión.....	56
4. Resultados experimentales.....	58
4.1. Contexto.....	58
4.2. Casos de estudio.....	60
4.2.1. Caso 1: apertura de los interruptores 22 y 23.....	61
4.2.1.1. Zonas desenergizadas.....	67
4.2.1.2. Lazos.....	68
4.2.1.3. Paralelos.....	80
4.2.2. Caso 2: Apertura de los interruptores 23, 22 y 9.....	95
4.2.2.1. Zonas desenergizadas.....	96
4.2.2.2. Lazos.....	97
4.2.2.3. Paralelos.....	98
4.2.3. Caso 3: Apertura de los interruptores 23, 22, 9 y 2.....	99
4.2.3.1. Zonas desenergizadas.....	100
4.2.3.2. Lazos.....	100
4.2.3.3. Paralelos.....	100
4.3. Conclusión.....	101
5. Conclusiones.....	102
5.1. Trabajos futuros.....	103
Bibliografía.....	105

Índice de Figuras

2.1.1. Paquetes CIM y dependencias.....	14
2.2.1. Diagrama de nodos y arcos RDF.....	16
2.3.1. Ejemplo de instancia de modelo RDF con JENA.....	17
2.3.2. Ejemplo de instancia de línea de transmisión.....	17
2.5.1. Ejemplo de Grafo no dirigido.....	19
2.5.1.1. Ejemplo de grafo para algoritmo DFS.....	20
2.6.1. Circuito ejemplo con interruptores.....	22
2.7.1.1.1. Circuito ejemplo de zonas desenergizadas.....	24
2.7.1.2.1. Circuito ejemplo de lazos con Interruptor1 cerrado e Interruptor2 abierto.....	25
2.7.1.2.2. Circuito ejemplo de lazos con Interruptor1 abierto e Interruptor2 cerrado.....	25
2.7.1.3.1. Circuito ejemplo de paralelos con Interruptor1 cerrado e Interruptor2 abierto.....	26
2.7.1.3.2. Circuito ejemplo de paralelos con Interruptor1 abierto e Interruptor2 cerrado.....	27
2.7.2.1.1. Red radial ejemplo.....	28
2.7.2.2.1. Red en lazo ejemplo.....	29
2.7.2.3.1. Ejemplo de una red mixta.....	30
3.1.1. Interacción entre el Editor, el modelo CIM estático, el Procesador topológico y el NCA.....	32
3.2.1. Clase Java Model.....	33
3.2.2. Clases Java ComponentsDataRepository, ComponentsConsultorData y BarraConsultorData.....	34
3.2.3. Clase Java SPARQLManager	34
3.2.1.1. Diagrama de clases del patrón de diseño Abstract Factory.....	35
3.2.1.2. Diagrama de clases donde se ve reflejado el patrón AbstractFactory en el procesador topológico.....	36
3.2.2.3. Circuito ejemplo para la construcción del grafo completo.....	37
3.2.2.4. Diagrama de clases para el grafo completo.....	38
3.2.3.1. Circuito ejemplo luego de calcular el grafo reducido.....	39
3.2.3.2. Diagrama de clases para la creación del grafo reducido.....	40
3.3.1. Diagrama de clases para el patrón de diseño Strategy	41
3.3.2. Diagrama de clases para los distintos algoritmos de búsqueda en el grafo reducido.....	42
3.3.1.1. Diagrama de actividades del método search para zonas desenergizadas.....	43
3.3.1.2. Diagrama de actividades del método dfs para zonas desenergizadas.....	44

3.3.2.1. Diagrama de actividades del método search para búsqueda de lazos.....	46
3.3.2.2. Diagrama de actividades del método dfs para búsqueda de lazos.....	47
3.3.3.1. Diagrama de actividades del método search para búsqueda de paralelos.....	50
3.3.3.2. Diagrama de actividades del método dfs para búsqueda de paralelos.....	51
3.4.1. Diagrama de clases final para Searcher.....	53
3.5.1. Diagrama de secuencia para la búsqueda de zonas desenergizadas.....	54
3.5.2. Diagrama de secuencia para la búsqueda de lazos.....	55
3.5.3. Diagrama de secuencia para la búsqueda de lazos.....	56
4.1. Georeferencia de la ciudad de Armstrong.....	58
4.2. Sección de la red Eléctrica de Armstrong.....	59
4.2.1.1. Apertura de los interruptores 22 y 23 en un sector de la red de Armstrong.....	61
4.2.1.2. Circuito ejemplo.....	63
4.2.1.1.1. Zonas desenergizadas con apertura de los interruptores 22 y 23.....	67
4.2.1.2.1. Primer recorrido entre Barra1 y Barra9.....	71
4.2.1.2.2. Segundo recorrido entre Barra1 y Barra9.....	71
4.2.1.2.3. Primer recorrido entre Barra1 y Barra8.....	72
4.2.1.2.4. Segundo recorrido entre Barra1 y Barra8.....	72
4.2.1.2.5. Primer recorrido entre Barra1 y Barra.....	73
4.2.1.2.6. Segundo recorrido entre Barra1 y Barra3.....	73
4.2.1.2.7. Primer recorrido entre Barra1 y Barra2.....	74
4.2.1.2.8. Segundo recorrido entre Barra1 y Barra2.....	74
4.2.1.2.9. Primer recorrido entre Barra1 y Barra5.....	75
4.2.1.2.10. Segundo recorrido entre Barra1 y Barra5.....	75
4.2.1.2.11: Primer recorrido entre Barra1 y Barra4.....	76
4.2.1.2.12. Segundo recorrido entre Barra1 y Barra4.....	76
4.2.1.2.13. Primer recorrido entre Barra1 y Barra7.....	77
4.2.1.2.14. Segundo recorrido entre Barra1 y Barra7.....	78
4.2.1.2.15. Primer recorrido entre Barra8 y Barra1.....	79
4.2.1.2.16. Segundo recorrido entre Barra8 y Barra1.....	79
4.2.1.3.1. Primer paralelo, la Barra 9 recibe energía de la Barra1 y Barra8.....	84
4.2.1.3.2. Segundo paralelo, la Barra9 recibe energía de la Barra1 y Barra8.....	85
4.2.1.3.3. Primer paralelo, la Barra8 recibe energía de la Barra 1 y de si misma.....	86

4.2.1.3.4. Segundo paralelo, la Barra8 recibe energía de la Barra1 y de sí misma.....	86
4.2.1.3.5. Primer paralelo, la Barra1 recibe energía de la Barra8 y de sí misma.....	87
4.2.1.3.6. Segundo paralelo, la Barra1 recibe energía de la Barra8 y de sí misma.....	88
4.2.1.3.7. Primer paralelo, la Barra7 recibe energía de la Barra1 y la Barra8.....	89
4.2.1.3.8. Segundo paralelo, la Barra7 recibe energía de la Barra1 y la Barra8.....	90
4.2.1.3.9. Primer paralelo, la Barra3 recibe energía de la Barra1 y la Barra8.....	91
4.2.1.3.10. Segundo paralelo, la Barra3 recibe energía de la Barra1 y la Barra8.....	91
4.2.1.3.11. Primer paralelo, la Barra2 recibe energía de la Barra1 y la Barra8.....	93
4.2.1.3.12. Segundo paralelo, la Barra2 recibe energía de la Barra1 y la Barra8.....	93
4.2.1.3.13. Primer paralelo, la Barra5 recibe energía de la Barra1 y la Barra8.....	94
4.2.1.3.14. Segundo paralelo, la Barra5 recibe energía de la Barra1 y la Barra8.....	95
4.2.2.1. Apertura de los interruptores 22, 23 y 9 en un sector de la red de Armstrong.....	96
4.2.3.1: Apertura de los interruptores 22, 23, 9 y 2 en un sector de la red de Armstrong.....	99

Índice de tablas

2.2.1: Alternativas para representar el modelo de datos.....	15
2.4.1: Comparación de alternativas para persistencia de información.....	18
4.1: Componentes que se pueden utilizar en el Editor.....	60

Resumen

Las interrupciones de los sistemas principales dentro de las redes modernas de energía son casos excepcionales. Sin embargo, cuando un gran evento ocurre, puede implicar la pérdida de muchos componentes del sistema y perjuicios para los usuarios.

Las fallas en cascada son las razones típicas de los apagones en las redes eléctricas. En este sentido la topología de la red desempeña un papel importante en la determinación de la dinámica de las fallas en cascada en las redes eléctricas.

Este trabajo tuvo como objetivo principal implementar una herramienta de análisis de conectividad de la red eléctrica, que permitió determinar su topología basada en la información de las cargas (consumos) y las fuentes (generación), junto con la información de conectividad de los componentes que la constituían.

Para ello, se buscaron las zonas desenergizadas, lazos y paralelos dentro de la red de distribución, utilizando la información de los estados de los interruptores. Los lazos se forman cuando en la red existen rutas alternativas de flujo de energía para una fuente de alimentación única; en cambio, los paralelos son una estructura topológica en el cual se alimenta con más de una fuente de energía.

Además, fue necesario representar la red de energía eléctrica en estructuras de grafo basadas en el estado de los interruptores. Esto permitió aplicar algoritmos de búsqueda sobre variantes del algoritmo DFS (Depth-First Search), y obtener las salidas necesarias de las zonas desenergizadas, lazos y paralelos.

Por último, se realizaron validaciones para tres casos de prueba, abriendo y cerrando interruptores para generar diferentes escenarios, calcular tiempos de ejecución y analizar la concordancia de las salidas.

Capítulo 1

Introducción

1.1. Motivación

Las actuales redes de energía eléctrica están avanzando hacia una generación más eficiente y segura, impulsadas por nuevas fuentes de información y tecnologías de comunicación. Sumado a esto, el conocimiento del estado de la red eléctrica permite mejorar la operación del sistema, pudiendo integrar nuevas funcionalidades que asistan a los operadores en la toma de decisiones [14] [18]. Por ejemplo, la integración de la comunicación digital bidireccional basada en varios sensores a través de la red, permite realizar monitoreo y recuperación de manera casi automática.

En el ámbito de operación de una red de distribución, las funciones más comunes son la visualización de la red, la estimación del estado, el control del sistema y la adquisición de datos. A pesar de su relevancia, hay muchas pequeñas y medianas empresas de distribución que tienen problemas para incorporar estas soluciones, y el motivo más común es el alto costo asociado con la licencia, la instanciación de la red y el mantenimiento.

En este sentido están surgiendo nuevas soluciones acordes a la realidad regional, que posibilitan a las distribuidoras operar sobre los datos adquiridos. Un ejemplo de esto es el proyecto TENERGIA [29], cuyo objetivo es el desarrollo de un Sistema de Gestión de Distribución (Distribution Management System - DMS) basado en tecnología de código abierto (Open Source). La herramienta incluye una colección de aplicaciones para monitorear, controlar y analizar las redes de distribución ubicadas en la Argentina [5], y su diseño permite agregar funciones y módulos de acuerdo a las necesidades de la distribuidora.

Sobre la base de este desarrollo surge la necesidad de incorporar un módulo de análisis de conectividad de la red, el cual asistirá al administrador u operador de la misma en la determinación de las zonas desenergizadas, lazos (múltiples rutas desde una fuente) y paralelos (múltiples fuentes hacia un único destino).

1.2. Estado del Arte

Las interrupciones dentro de las redes modernas de energía son casos excepcionales. Sin embargo, cuando un gran evento ocurre, puede implicar la pérdida de muchos componentes del sistema [17].

Por ejemplo, el 6 de Junio del 2018, una falla en la red de alta tensión provocó un apagón en varios barrios de la Ciudad de Buenos Aires - Argentina que afectó a 290 mil usuarios. El corte masivo se extendió por unos 40 minutos [33]. Desde Edesur

explicaron que se registró una falla en la red de alta tensión en siete subestaciones de sus distribuidoras que afectaron a 290 mil clientes [32].

Las fallas en cascada (como es el caso anterior) son las razones típicas de los apagones en las redes eléctricas, donde la topología de la red desempeña un papel importante en la determinación de la dinámica de las fallas.

En tal sentido, la teoría de grafos se ha utilizado de diferentes maneras para estudiar la topología de los sistemas de energía [15]. Una red eléctrica puede ser esquematizada por un grafo donde los nodos representan la generación, transmisión, buses de distribución y subestaciones, mientras que los arcos modelan las líneas de transmisión y los transformadores.

Otro ejemplo es utilizar teoría de grafos para aplicar algoritmos de flujo. Los mismos sirven para determinar el flujo máximo que se puede entregar desde una fuente a una carga dentro de un grafo dirigido, donde cada arco tiene una capacidad de flujo específica [17].

En todos los casos, los métodos basados en la teoría de grafos pueden proporcionar asistencia a los operadores en una gran variedad de escenarios con diferentes cantidades de información disponible.

1.2.1. Análisis de conectividad de Red (Network Connectivity Analysis-NCA)

La red de distribución generalmente cubre un área grande y proporciona alimentación a diferentes clientes con diferentes niveles de voltaje. Por lo tanto, a menudo es muy difícil ubicar las fuentes y las cargas necesarias en una interfaz de operador/GIS [12].

El análisis de conectividad de red es una funcionalidad específica de los sistemas de gestión que le permite al operador identificar datos relevantes de la red con mucha facilidad. En función del estado de todos los dispositivos de conmutación, como el disyuntor (CB), la unidad principal de anillo (RMU) y / o los aisladores que afectan la topología de la red modelada, el NCA permite determinar la topología de la red predominante. Además, puede asistir al operador sobre el estado operativo de la red de distribución, indicando el modo radial, los bucles y los paralelos [13].

A medida que la red eléctrica aumenta de tamaño y complejidad, es cada vez más importante comprender los comportamientos emergentes que pueden tener lugar en el sistema [24]. Considerando esto, en la actualidad existen diferentes compañías que invierten en tecnología con el fin de desarrollar redes de energía más eficientes. Un ejemplo conocido en el mercado eléctrico es ETAP, y que es considerado un líder mundial en tecnología para el modelado, diseño, análisis, optimización, monitoreo, control y software de automatización para sistemas de energía eléctrica [27].

Esta empresa cuenta con un módulo de análisis de conectividad en tiempo real, que calcula el estado de activación e interactúa con la interfaz de usuario mediante el coloreo de mapas. Los colores dedicados se utilizan para el seguimiento de circuitos

y alimentadores, y también cuando la información disponible no es completa o es inconsistente. Además, esta aplicación permite modificaciones temporales en cualquier punto de la red de distribución para cambiar su configuración, aislar fallas, restaurar servicios o realizar tareas de mantenimiento [20].

Si bien es un caso muy interesante desde el punto de vista de la funcionalidad, es un módulo difícil de integrar en sistemas preexistentes, sumado a los altos costos de adquisición del producto. Esta situación se repite para otros productos del mercado, y es la base sobre la cual resulta factible desarrollar soluciones acordes a la aplicabilidad de la región.

1.3. Propuesta

El objetivo de este trabajo es desarrollar una herramienta de software para determinar la topología de la red basada en la información obtenida sobre las cargas (consumos), las fuentes (generación) y los estados de los interruptores [19].

La función principal de esta herramienta es asistir al operador sobre el estado de funcionamiento de la red de distribución, indicando por ejemplo el modo radial, lazos, y paralelos en la red. Las redes de distribución funcionan normalmente en modo radial, donde los lazos y paralelos pueden ser intencionales o formarse de forma inadvertida. Todos los lazos y paralelos generalmente se grafican sobre el entorno de visualización de la red con diferentes colores, de tal manera que cada uno sea fácilmente identificable dada la importancia de su determinación.

A fin de lograr mayor interoperabilidad en la representación de la red eléctrica se trabajará sobre un modelo CIM de representación (Common Information Model), que además constituye un estándar para el mercado eléctrico [5]. El objetivo del modelo CIM es describir los datos de comunicación entre los sistemas, subsistemas y aplicaciones dentro de la empresa de energía eléctrica [31]. Por ende, se pretende que CIM sea un vocabulario común para los datos intercambiados.

Sobre la base del modelo CIM, se propone además analizar la información de la red eléctrica para determinar la conectividad entre los diferentes elementos de la misma. Básicamente permitirá determinar la situación de todos los dispositivos de conmutación que afectan a la topología de la red.

Considerando estos lineamientos se plantearon los siguientes objetivos específicos: comprender un modelo CIM de una red eléctrica; generar una representación de la red que permita aplicar algoritmos computacionales de detección de topologías [9], zonas desenergizadas, y estado de interruptores [20]; implementar una interfaz de comunicación entre ambas representaciones; plantear y analizar distintos casos de estudios sobre una red eléctrica ejemplo.

Finalmente, se propone el diseño e implementación de un módulo para el NCA como un componente más dentro del proyecto DMS-TENERGIA.

1.4. Organización

La estructura general del presente trabajo está dividida en los siguientes capítulos. El capítulo 2 presenta los fundamentos teóricos. Primero se describe la representación, persistencia y consulta sobre la red considerando el modelo CIM y los frameworks RDF y Jena. Luego, se introduce sobre la aplicación de teoría de grafos y el algoritmo DFS (Depth-First Search) utilizado para recorrer este tipo de estructuras. Después, se presenta los diferentes estados de los interruptores y el papel que juegan en la topología de red. Además, se describe el concepto topología de la red y se define zonas desenergizadas, lazos y paralelos.

En el capítulo 3 se presenta el análisis y las decisiones de diseño. Posteriormente se detalla la implementación de la herramienta utilizando diferentes patrones de diseño.

El capítulo 4 describe los tres casos de prueba realizados, los resultados obtenidos, y los tiempos de ejecución para cada funcionalidad.

Finalmente, en el capítulo 5 se exponen las conclusiones y los posibles trabajos futuros sobre la base de lo desarrollado.

Capítulo 2

Análisis de conectividad de red

Un DMS (Distribution Management System) es un conjunto de componentes de software los cuales monitorean y controlan la red con el objetivo de optimizar el comportamiento de una red eléctrica inteligente, y permitir que las empresas de servicios proporcionen energía de una manera más confiable, eficiente y segura [5]. El NCA forma parte del DMS, y constituye una herramienta computacional encargada de brindar información relevante de la red de energía eléctrica, como por ejemplo zonas desenergizadas, lazos y paralelos.

En este capítulo se describe las herramientas seleccionadas que se tuvieron en cuenta para el desarrollo del NCA, el cual se integró al DMS desarrollado en el proyecto TENERGIA.

Primero se trabajó sobre un modelo común para facilitar el intercambio de información entre los distintos módulos que integran el DMS. Para ese objetivo se optó por un modelo CIM que define un estándar de representación.

Luego se necesitó de una herramienta para operar sobre el modelo y persistirlo. Por un lado se seleccionó los grafos RDF como esquema de modelo de datos, y por otro RDF/XML para persistir el mismo ya que es un estándar reconocido para el intercambio entre compañías.

Al estar DMS desarrollado en Java se eligió el framework Jena para manejar y administrar datos representados con grafos RDF.

Por último para poder aplicar los diferentes algoritmos de búsqueda se debió plasmar el circuito de energía eléctrica en una estructura de Grafo. Estas estructuras permiten estudiar las interrelaciones entre unidades que interactúan unas con otras.

2.1. Modelo CIM

Los sistemas de energía y especialmente los sistemas de distribución, cuentan con numerosos módulos que requieren compartir información de un modelo de datos común. En general, estos módulos son desarrollados por distintos fabricantes y la información utilizada puede requerir de una adaptación. En este contexto, EPRI (Electric Power Research Institute) estableció un modelo de datos estándar para estos sistemas denominado CIM (Common Information Model) [16]. Posteriormente la IEC (International Electro-technical Commission) lo adoptó como el modelo de información internacional estándar para la gestión de los sistemas eléctricos, con lo cual el modelo CIM propone la manera de organizar toda la información que pueda ser necesaria en las aplicaciones dedicadas a la gestión de las redes eléctricas.

El EPRI define CIM dentro de las series de normas IEC 61970 e IEC 61968. Las primeras, se centran en los sistemas de gestión de las redes de transporte, también llamados en la literatura EMS (Energy Management System); y las segundas extienden el paquete para soportar los sistemas de gestión de las redes de distribución o DMS [25]. En la Figura 2.1.1 puede verse un diagrama de la interacción de los paquetes. Además, las normas mencionadas describen una arquitectura de referencia para la integración entre las aplicaciones, los servicios de comunicación de acceso a la información, y formatos para el intercambio de información. Por ejemplo, CIM/XML [8] [16] es un formato conocido para el intercambio de grandes cantidades de información acerca de la topología de las redes eléctricas a gestionar. También se utilizan otros formatos como CIM/XSD [16] o CIM/SVG [7].

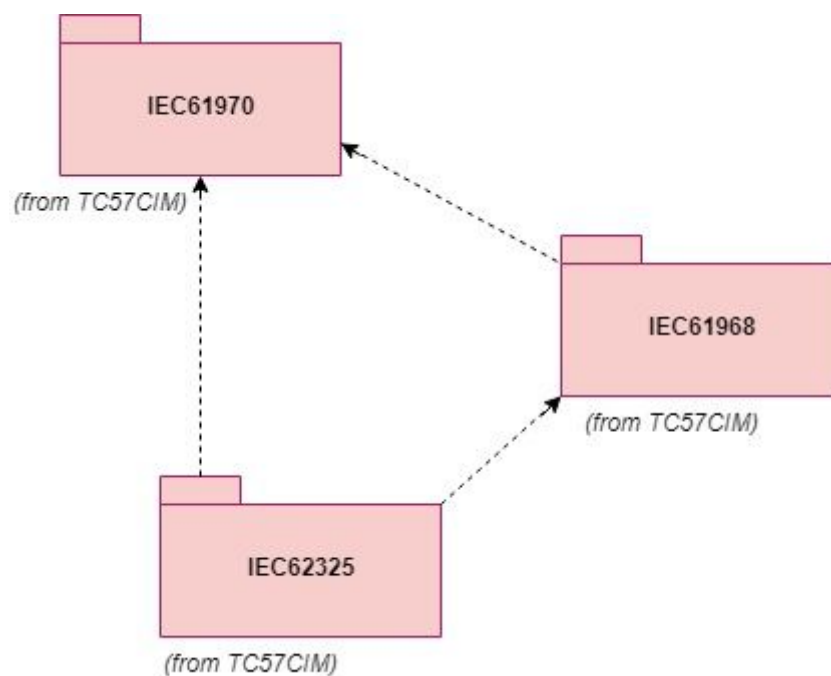


Figura 2.1.1: Paquetes CIM y dependencias [16].

CIM está descrito mediante diagramas de clases y paquetes, cuya composición arquitectónica ha sido ampliamente documentada. No obstante, no hay descripción clara de cómo instanciar el modelo [4].

A partir de la solución propuesta en la referencia anterior, se realizó una instanciación del NCA sobre la base de un modelo RDF, el cual se describe a continuación.

2.2. RDF

Teniendo en cuenta la utilización del modelo CIM como esquema de modelo de datos, para el proyecto TENERGIA se analizaron dos alternativas muy utilizadas

para trabajar con dicho modelo. En la primera, se considera la instanciación de las clases generadas con alguna herramienta partiendo del esquema de paquetes base [8]. Por otra parte, considerando que en la documentación del estándar CIM se propone manejar la persistencia del modelo a través de representaciones RDF. Ésta solución constituye la segunda alternativa y en la Tabla 2.2.1 [4] se muestra la comparación entre ambas.

		Clases	RDF
¿Qué guarda?		Instancia de objetos.	Grafo que representa instancia de objetos.
Complejidad	Entendimiento	Baja. Se utiliza directamente el esquema de clases.	Media/Baja. Se debe determinar el mapeo pero no requiere de mucha complejidad.
	Escalabilidad	Sube. Utilización de estructuras auxiliares.	No varía.
	Guardado	Alta. Debe programar alternativas.	Directo. Utilizando herramienta.
	Consulta de datos	Alta. Utilización de estructuras auxiliares.	Media. A través de lenguaje de consultas.
Herramientas		Cualquier lenguaje orientado a objetos.	Jena para Java. LinqToRdf para C#.

Tabla 2.2.1: Alternativas para representar el modelo de datos [4].

Para el mencionado proyecto se optó por trabajar con un modelo RDF, y es la misma línea que se sigue para la instanciación del NCA. Concretamente se utilizó un estándar sobre la base de RDF definido por World Wide Web Consortium (W3C) para representar modelo de datos y metadatos.

El modelo de datos de RDF se basa en 3 elementos fundamentales:

- Recursos: Representa una entidad de la cual se desean presentar características y relaciones. En CIM se mapea con una instancia de una clase.
- Propiedades: Refleja el nombre de una característica del recurso. En cuanto al uso del CIM, representa un atributo de una clase.
- Literales. Valores de una propiedad de un recursos. En CIM se corresponde con un valor de un atributo de una instancia de una clase.

Se puede observar en la Figura 2.2.1 un diagrama RDF donde los recursos se escriben dentro de una elipse, la propiedad se identifica en el arco del grafo y el

literal se identifica con un rectángulo. También se puede identificar que es un grafo dirigido, es decir que el arco tiene un sentido [3].

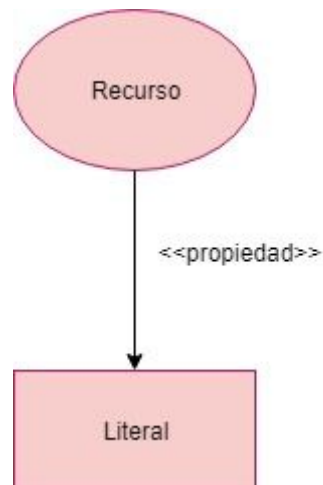


Figura 2.2.1: Diagrama de nodos y arcos RDF.

2.3. JENA

Considerando utilizar RDF se procedió a buscar una herramienta que permita operar sobre el modelo. Teniendo en cuenta que el proyecto TENERGIA fue desarrollado en Java se decidió utilizar JENA. La misma provee un framework para manejar y administrar datos representados con grafos RDF [6].

JENA presenta una API (Application Programming Interface) para poder generar y administrar los recursos, propiedades y literales nombrados en la sección 2.2. Este manejo se realiza a través del armado de un conjunto de declaraciones (Statements). Una declaración es una 3-tupla que representa un hecho en el modelo, la cual consta de los siguientes elementos:

- Sujeto: El recurso descrito.
- Predicado: La propiedad del recurso descrito.
- Objeto: El valor del predicado, y puede ser un literal si se trata de un atributo, un recurso, o una relación entre objetos.

El conjunto de estas 3-tuplas forman un grafo que representa un modelo RDF como se muestra en la Figura 2.3.1 [3].

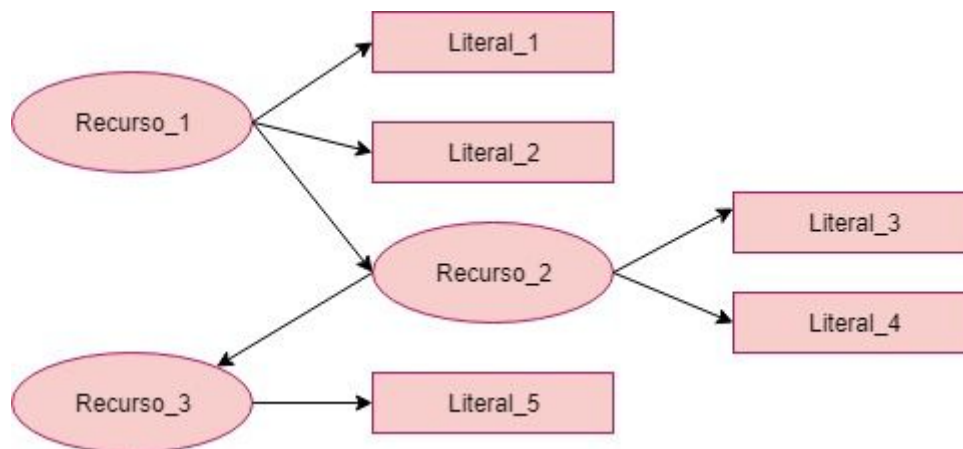


Figura 2.3.1: Ejemplo de instancia de modelo RDF con JENA [4].

A modo de ejemplo, el paquete Wires de la norma IEC 61970 tiene la clase ACLineSegment que representa una línea de transmisión. Esta clase tiene los atributos necesarios para definir este tipo de componente. Entre ellos se pueden nombrar: Susceptancia (bch), Conductancia (gch), Resistencia (r), etc. Una instancia de este tipo de clase puede ser mapeada a través de un recurso JENA que cuenta con literales que representan cada uno de esos parámetros. En la Figura 2.3.2 se representa este caso.

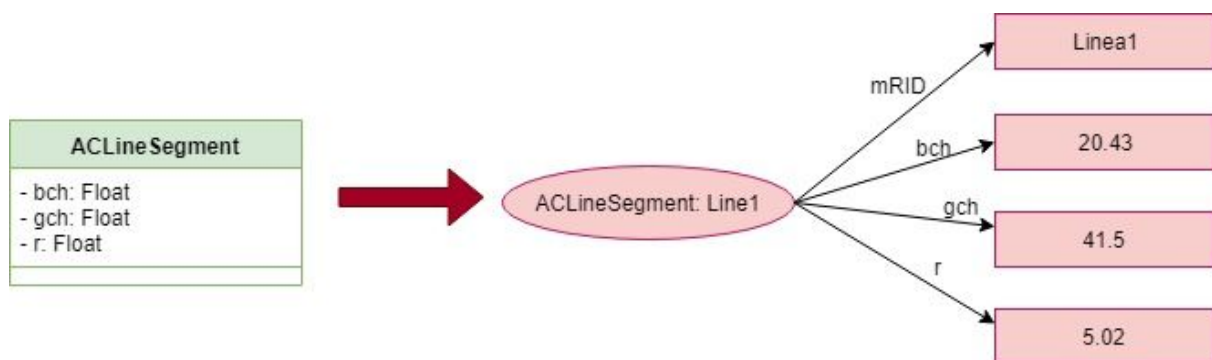


Figura 2.3.2: Ejemplo de instancia de línea de transmisión [4].

2.4. RDF/XML

A partir de la utilización del estándar CIM, otro punto a tener en cuenta es la persistencia de los datos. Para ello, se investigaron diferentes alternativas de guardado y modos de acceso a la información [4]. Los resultados comparativos de las soluciones más utilizadas son presentados la Tabla 2.4.1.

	Serializado	XML	Formato RDF	Json
¿Que guarda?	Objetos en binario. Guarda clases Java.	Estructura de datos con atributos. Guarda información serializada.	Estructura, semántica y asociaciones. Ideal para aguardar modelos de datos.	Sólo estructura.
Complejidad	Baja	Media	Media/Alta	Media/Baja
Legibilidad	Nada	Media	Media	Media
Consultas	-	Si (XGL, XML, QL)	Si (SPARQL)	Si (JSONiq)
Representación	-	Árbol (DOM)	Grafo	Árbol
Herramientas	Java	JAXB(Java)	Jena	JSONSimple (Java)

Tabla 2.4.1: Comparación de alternativas para persistencia de información [4].

Luego de analizar las distintas opciones se concluyó que la más adecuada y directa era utilizar algún formato de representación RDF. Además, la herramienta utilizada para el armado del modelo cuenta con la funcionalidad de guardar y cargar los datos. Sabiendo que existen varias maneras de representar un modelo de este tipo, que permitan persistir la información de manera legible [6], se decidió que RDF/XML es el más estandarizado para el intercambio entre compañías.

Sumado a lo anterior, el esquema RDF cuenta con un lenguaje de consultas sobre los datos, que permite acceder a ellos de manera más ágil. Dicho lenguaje se denomina SPARQL6 (Protocol and RDF Query Language), y es una tecnología clave en el desarrollo de aplicaciones que utilizan este tipo de modelos.

2.5. Grafos

Para poder analizar la red eléctrica según los objetivos del NCA se decidió modelar e implementar la misma en un grafo, ya que este tipo de estructuras permiten estudiar las interrelaciones entre unidades que interactúan unas con otras.

Un Grafo $G=(V,A)$ es una colección de puntos llamados vértices, unidos por líneas llamadas aristas A [10].

En particular se implementó un grafo no dirigido, es decir que sus aristas son bidireccionales. Esto se debe a que en el sistema de energía eléctrica no hay dirección en las distintas conexiones entre las “barras”. A su vez, a los vértices se los consideraron como nodos y a las aristas como arcos [1].

La figura 2.5.1 muestra un grafo no dirigido con cuatro nodos llamados A, B, C y D. Existen cuatro arcos, y se representan como pares no ordenados:

- Arcos: $\{\{A,B\}, \{A,C\}, \{A,D\}, \{B,D\}\}$

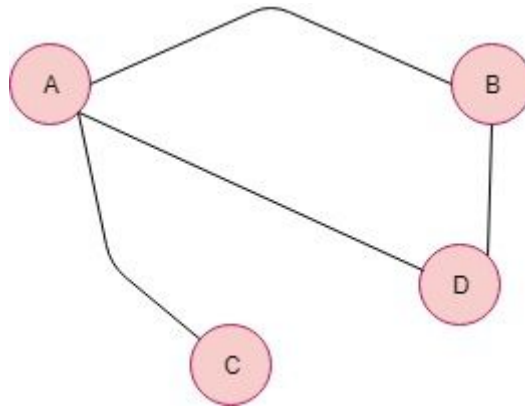


Figura 2.5.1: Ejemplo de Grafo no dirigido.

2.5.1. Algoritmo de búsqueda DFS (Depth-First Search)

Teniendo en cuenta la funcionalidad del NCA, el recorrido de grafo que mejor se ajusta es la búsqueda en profundidad (DFS-Depth-First Search). En la misma primero se explora los arcos partiendo del vértice o nodo v más recientemente descubierto que todavía tiene arcos inexplorados. Una vez que todos los arcos fueron explorados, la búsqueda realiza “backtracking” para explorar los arcos partiendo del vértice o nodo v que fue descubierto. Por ejemplo, se puede observar en la Figura 2.5.1.1 el algoritmo partió del nodo 1 y continuó su recorrido por el arco que existe entre los nodos 1 y 2, y así visitó los nodos 2, 5 y 7. Al llegar al nodo 7 el algoritmo deberá realizar un backtracking para continuar la búsqueda desde el nodo 1 que fue con el que se empezó. Este proceso continúa hasta descubrir todos los vértices o nodos que son alcanzables desde el primer vértice, para el caso del ejemplo se está hablando del vértice 1. De todos modos el DFS va a visitar todos los vértices o nodos del grafo, es decir que el nodo 8 de la Figura 2.5.1.1 si bien no está conectado con el resto de los nodos será visitado en algún momento de la búsqueda [11].

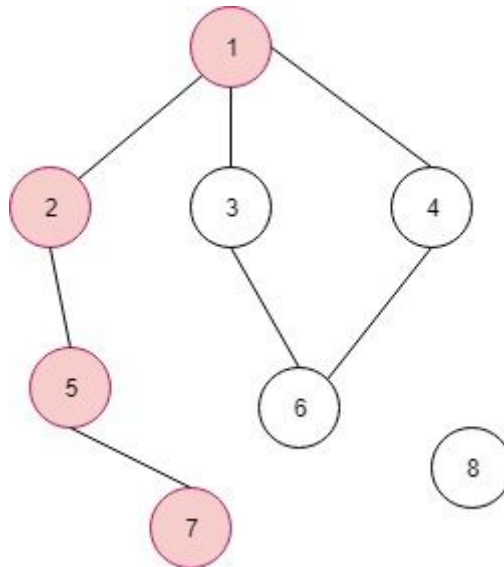


Figura 2.5.1.1: Ejemplo de grafo para algoritmo DFS.

```

DFS(G){
  Para cada vértice  $u \in G.V$ {
     $u.estado = no\_visitado$ ;
     $u.padre = null$ ;
  }
  Para cada vértice  $u \in G.V$ {
    Si  $u.estado$  es  $no\_visitado$ 
      DFS_Visitar(G,u);
  }
}

DFS_Visitar(G,u){
   $u.estado = visitado$ ;
  Para cada  $v \in u.vecinos$ {
    Si  $v.estado$  es  $no\_visitado$ {
       $v.padre = u$ ;
      DFS_Visitar(G,u);
    }
  }
}

```

Para analizar la complejidad temporal del algoritmo DFS se debió tener en cuenta lo siguiente:

Sean $G = (V, E)$, $|E| = e$, $|V| = n$, $m = \max(n, e)$

$$T_{DFS}(n, e) \in O(m)$$

Demostración:

Sea u un vértice y m_u el número de sucesores de u

$DFSvisitar(u)$ procesa a $(1 + m_u)$ vértices

$$\sum_{k=0}^{n-1} m_k = e$$

$$T_{DFS}(n, e) \leq \sum_{k=0}^{n-1} c(1 + m_k) = (nc + ec) = c(n + e)$$

$$T_{DFS}(n, e) \in O(m)$$

Según lo demostrado la complejidad temporal es $O(m)$, siendo m el máximo entre la cantidad de vértices y de arcos [1].

2.6. Estado de los interruptores

Los interruptores son elementos fundamentales en el equipamiento primario de estaciones de distribución de energía eléctrica. A diferencia de los seccionadores, pueden abrir y cerrar circuitos en carga, y forman parte del esquema de protección de las estaciones y subestaciones [2].

Hay que tener en cuenta que los mismos deben ser extremadamente confiables. Estos equipos están generalmente a la intemperie y se ven afectados directamente por el polvo, la humedad y las fluctuaciones de temperatura propias de las condiciones climáticas dónde se encuentran instalados [23].

El interruptor es un dispositivo biestable y puede ser accionado manualmente o desde un elemento externo, también integrado en el sistema de protección (por ejemplo un relé), en respuesta a la aparición o supresión de una falta de la red.

Su función es cortar efectivamente (abrir) la corriente de la línea ante la aparición de una falta o condición de servicio fuera de límites. En términos generales, no es el interruptor el que detecta la falta (aunque sí detecta sus propias faltas) sino que responde a las solicitudes de otros elementos del sistema de protección, como los relés. La mayoría de los interruptores pueden además restablecer el servicio (cerrar) cuando así se les solicite. Es decir que es capaz de establecer, conducir e interrumpir corrientes en condiciones normales del circuito; y también de establecer, conducir por un tiempo determinado, e interrumpir corrientes en determinadas condiciones anormales como las de cortocircuito.

Se puede observar dos interruptores en el circuito de la Figura 2.6.1. El Interruptor1 se encuentra cerrado, lo cual permite el paso de la corriente que viene de la Barra1 hacia la Barra2. Pero el Interruptor2 se encuentra abierto y no permite el paso de la corriente de la Barra2 hacia la Barra3.

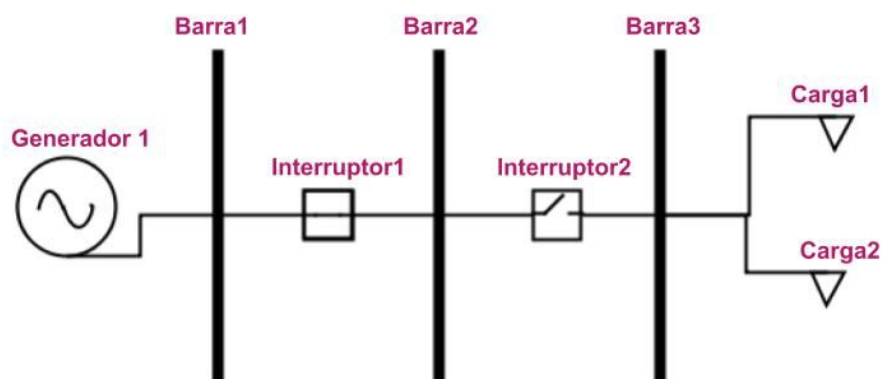


Figura 2.6.1: Circuito ejemplo con interruptores.

2.6.1. Configuración de los interruptores

Dependiendo de la configuración de los interruptores se obtienen los siguientes tipos de red:

- Distribución radial: Son varias líneas en paralelo independientes unas de otras. Si una salta, el resto sigue teniendo corriente pero la parte que ha saltado no recupera la corriente hasta que se repara la avería. Eso no se puede permitir en muchos casos.
- Red mallada: Se colocan los interruptores en una red. Cuando se produce el cortocircuito se abren los dos interruptores adyacentes dejando que el suministro llegue al resto de los puntos.
- Interruptor sencillo barra simple: Permite detectar si un interruptor ha fallado, es decir, si no ha abierto el circuito a pesar de que la intensidad es mayor de la nominal. Esto se hace con interruptores de medida. Al detectarse el fallo se abren los demás interruptores para evitar un fallo generalizado.
- Interruptor sencillo barra simple con enlace: Igual que el caso anterior pero tiene unos dispositivos de enlace que hacen de barrera entre las distintas partes del circuito, con lo cual si uno de los interruptores falla sólo se abrirán los interruptores cercanos, permitiendo que el resto funcione con normalidad.
- Interruptor sencillo, barras principales y de transferencia: Permite detectar el fallo en el interruptor y abrir sólo el interruptor que ha fallado, permitiendo que el resto funcione con normalidad.
- Interruptor doble barra doble: Más completo ya que hay un interruptor por cada barra.

2.7. Topología de la red

Al hablar de la topología de una red de distribución se refiere al esquema o arreglo de la distribución en que se distribuye la energía por medio de la disposición de los segmentos de los circuitos de distribución. En este sentido se enfoca a la forma como se distribuye la energía a partir de la fuente de suministro.

Las redes de distribución manejan ciertos tipos de topología para la conexión dependiendo del país, como por ejemplo red radial, red en lazo o malla y red mixta, que más adelante se dará detalle de cada una de ellas [30].

2.7.1. Conceptos básicos

En esta sección se van a detallar conceptos básicos necesarios para luego comprender las diferentes topologías de red.

2.7.1.1. Zonas desenergizadas

Las zonas desenergizadas son aquellas a las cuales no les llega energía eléctrica de la fuente o de las fuentes de la red, debido algún corte por un desperfecto o mantenimiento.

Como se puede observar en la Figura 2.7.1.1.1 hay dos zonas, una es la que pertenece a la Barra4 y la otra es la que surge de la Barra2. Debido a los estados de los interruptores, una de las zonas está desenergizada y la otra no. El Interruptor1 conecta la Barra 1 con la zona 2, y viendo que el mismo se encuentra cerrado dicha zona está alimentada por la fuente de energía, que en este caso es el Generador1. Pero el Interruptor2, el encargado de conectar la Barra1 con la zona 1 está abierto, es decir que corta el paso de la corriente hacia la misma. Por lo tanto dicha zona se la identifica como desenergizada.

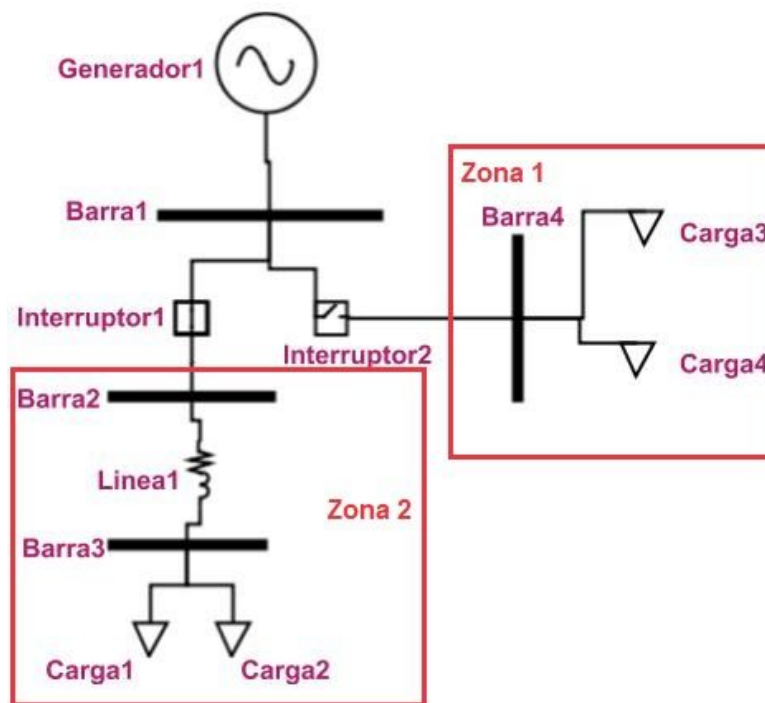


Figura 2.7.1.1.1: Circuito ejemplo de zonas desenergizadas.

2.7.1.2. Lazos

Existen sistemas de energía eléctrica que tienen una fuente de energía que alimenta diferentes zonas por diferentes caminos, es decir que si en un camino hay un corte las zonas continuarán conectadas a la fuente por otro camino sin ser desenergizadas. Para que esto ocurra debe haber lazos dentro del circuito.

Los lazos entonces son un conjunto de ramas que forman una línea cerrada, de tal forma que si se elimina cualquier rama del lazo, el camino queda abierto.

Igualmente para comprender bien el concepto de lazo, se muestra un ejemplo en la Figura 2.7.1.2.1 la cual contiene un lazo. Se puede observar marcado con flechas el camino de la corriente desde la fuente Generador1 hacia las demás barras. También se puede ver que el Interrupitor1 se encuentra cerrado y es el que marca el camino de la corriente. Ahora, si surge algún problema o mantenimiento de la red entre la Barra1 y la Barra2, entonces se puede abrir dicho interruptor y dejar el paso de la corriente cerrando el Interrupitor2. En efecto, la Barra4 seguirá contando con la alimentación de la fuente de energía, en cambio el camino de la corriente será diferente como se puede ver en la Figura 2.7.1.2.2.

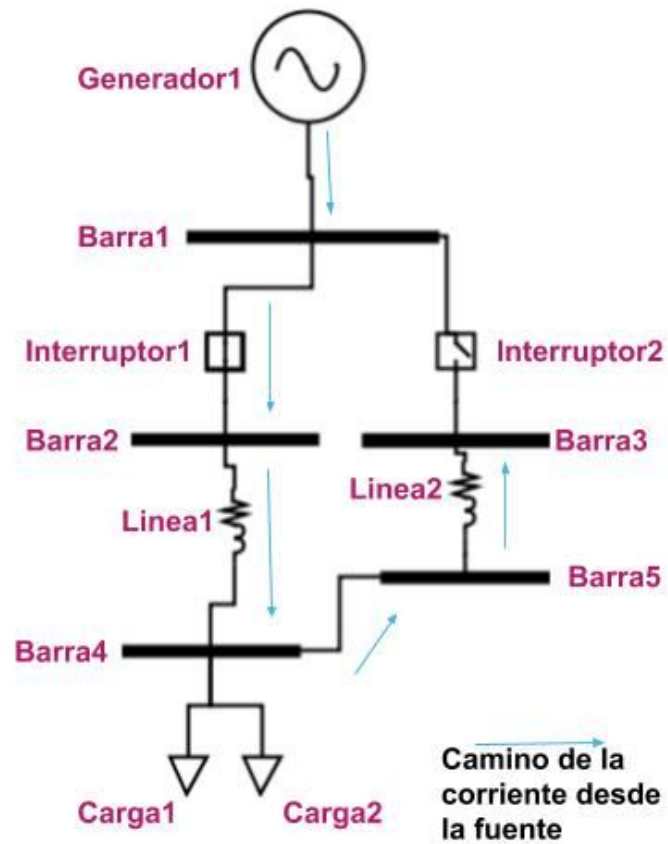


Figura 2.7.1.2.1: Circuito ejemplo de lazos con Interruptor1 cerrado e Interruptor2 abierto.

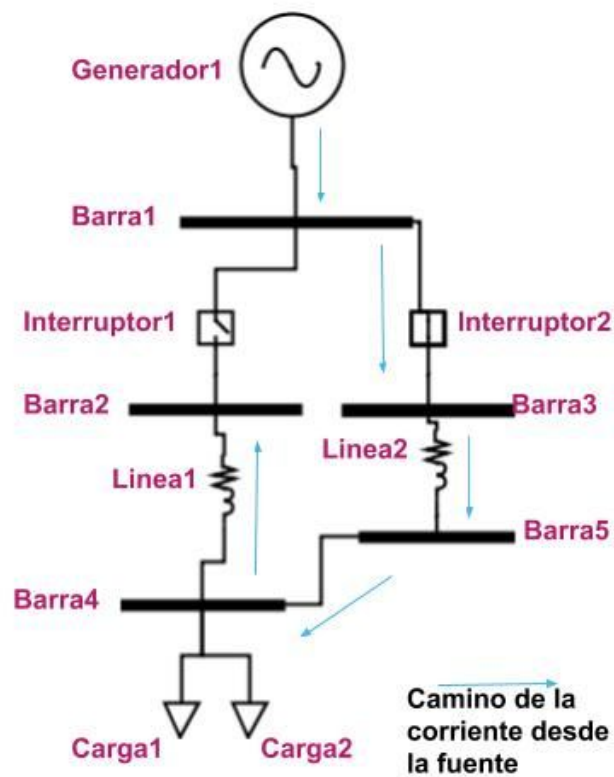


Figura 2.7.1.2.2: Circuito ejemplo de lazos con Interruptor1 abierto e Interruptor2 cerrado.

2.7.1.3. Paralelos

Por otra parte, se puede encontrar sistemas de energía eléctrica donde en vez de haber una sola fuente, hay dos o más fuentes que alimentan la misma zona desde distintos caminos. Así, si ocurre un problema con una de las fuentes o en el camino de distribución de energía, se puede contar con la otra fuente para seguir brindando el servicio a las zonas que alimentaba la fuente que falló.

Entonces, se puede decir que un circuito en paralelo es aquel que tiene más de una fuente de energía y las mismas alimentan a las mismas zonas pero por distintos caminos.

En el caso de la Figura 2.7.1.3.1 hay dos generadores los cuales por distintos caminos alimentan la Barra4. Al estar el Interruptor1 cerrado se marcó con flechas el camino de la corriente desde el Generador1 hasta la Barra4. Suponiendo que surge algún desperfecto en el Generador1 se puede contar con el Generador2 cerrando el Interruptor2 y abriendo el Interruptor1. De esa forma la Barra 4 seguirá teniendo el servicio de energía sin quedar desenergizadas durante la reparación del Generador1. Esta solución se puede contemplar en la Figura 2.7.1.3.2.

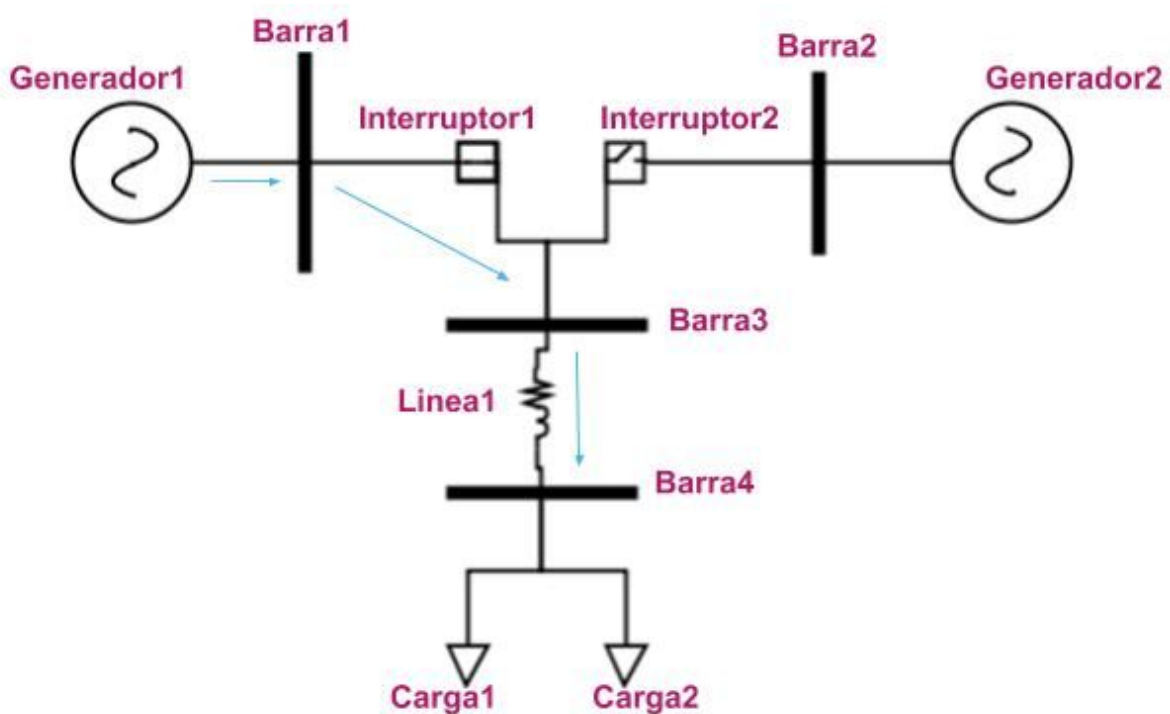


Figura 2.7.1.3.1: Circuito ejemplo de paralelos con Interruptor1 cerrado e Interruptor2 abierto.

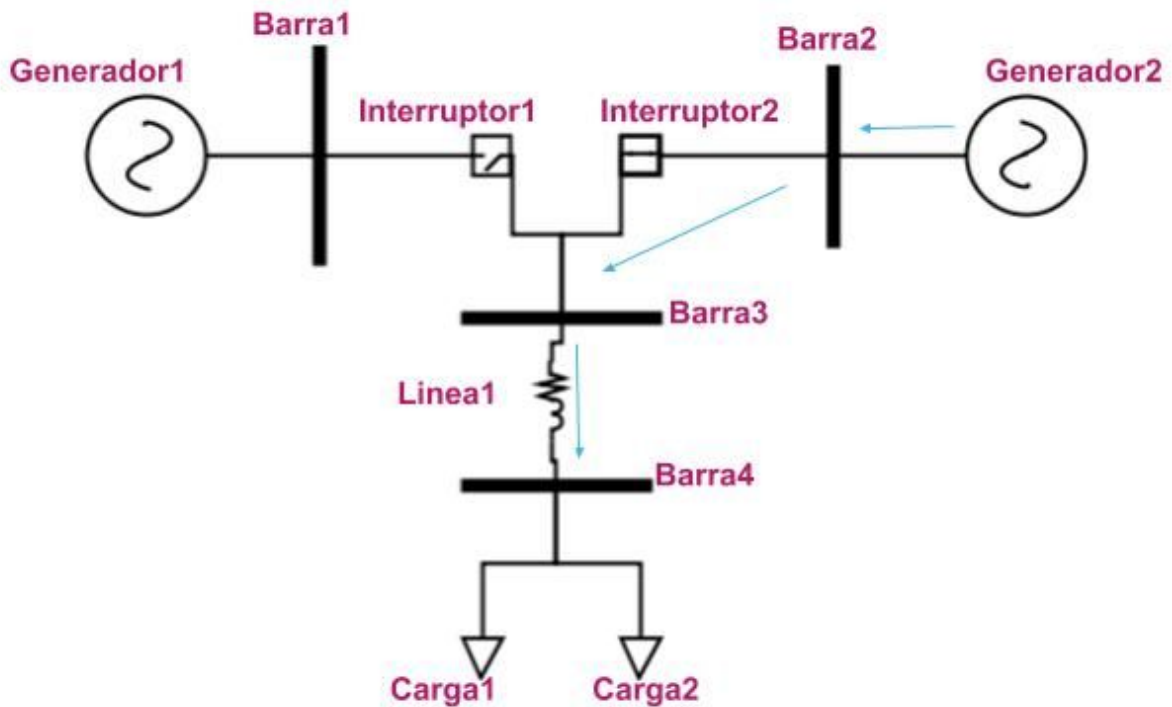


Figura 2.7.1.3.2: Circuito ejemplo de paralelos con Interruptor1 abierto e Interruptor2 cerrado.

2.7.2. Las diferentes topologías

Cada topología presenta ventajas y desventajas frente a modelos similares, ya sea por el tipo de conexión, costes, calidad de servicio, entre otros.

A continuación se detallará cada topología junto a sus ventajas y desventajas.

2.7.2.1. Red radial

Se caracteriza por la alimentación de uno solo de sus extremos transmitiendo la energía en forma radial a los receptores y el emisor. Así, las líneas de distribución se extienden desde la subestación como rayos de una rueda de bicicleta, de donde viene su nombre [30].

La principal ventaja de un arreglo radial es que son simples y económicos. Su desventaja es la falta de garantía de servicio, pudiendo dejar a muchos usuarios fuera de servicio hasta que el problema se resuelva. Estas desventajas pueden ser compensadas en la actualidad con los dispositivos modernos de desconexión automática de la zona en falla llamados "Órganos de Corte de Red" o la utilización de los dispositivos llamados "Reconectores", que desconectan y cierran la zona en falla, procurando de esa manera despejar la zona en falla y volver el servicio sobre la línea completa [26].

La Figura 2.7.2.1.1 muestra un ejemplo de una red radial.

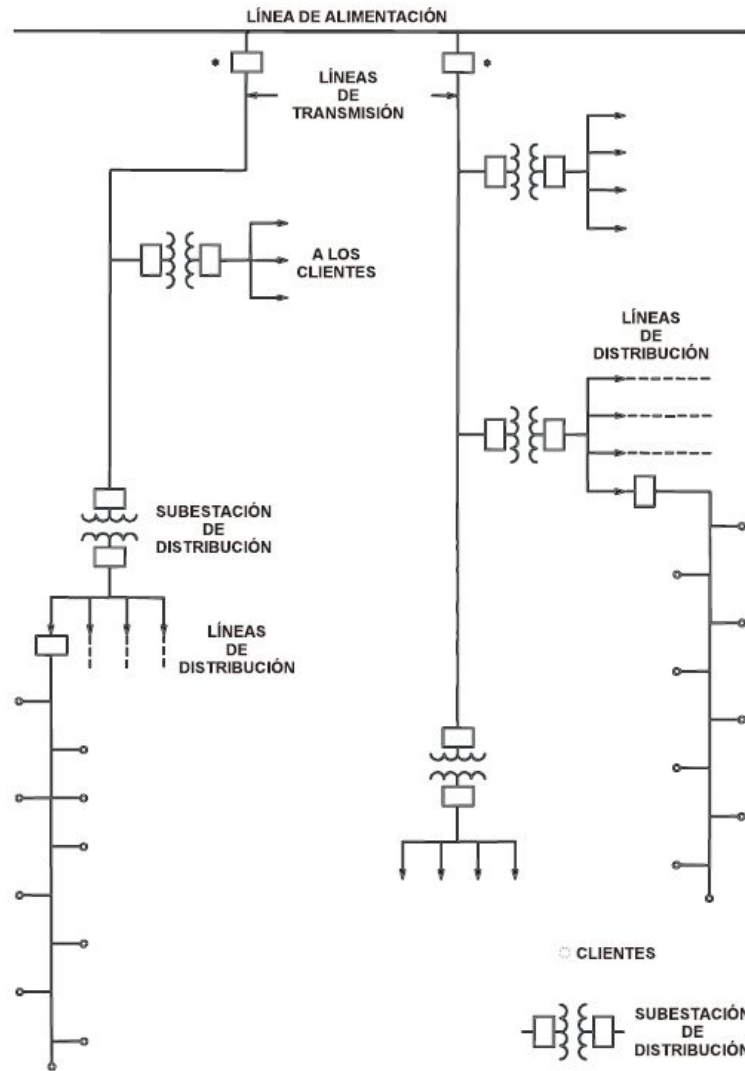


Figura 2.7.2.1.1: Red radial ejemplo [26].

2.7.2.2. Red en lazo o malla

Esta topología se utiliza en situaciones en las que se pretende aumentar la fiabilidad del servicio, ya que si hay una avería en un punto del anillo se puede mantener dicho servicio si se alimenta desde otro punto (se recomienda, por ejemplo, en polígonos industriales) [26]. Por otro lado, esta conexión es más costosa que la de arreglo radial debido a que requiere de más equipos para que de cualquier punto sobre la línea se tenga más de una alimentación [30].

Se puede observar un ejemplo de una red en lazo en la Figura 2.7.2.2.1.

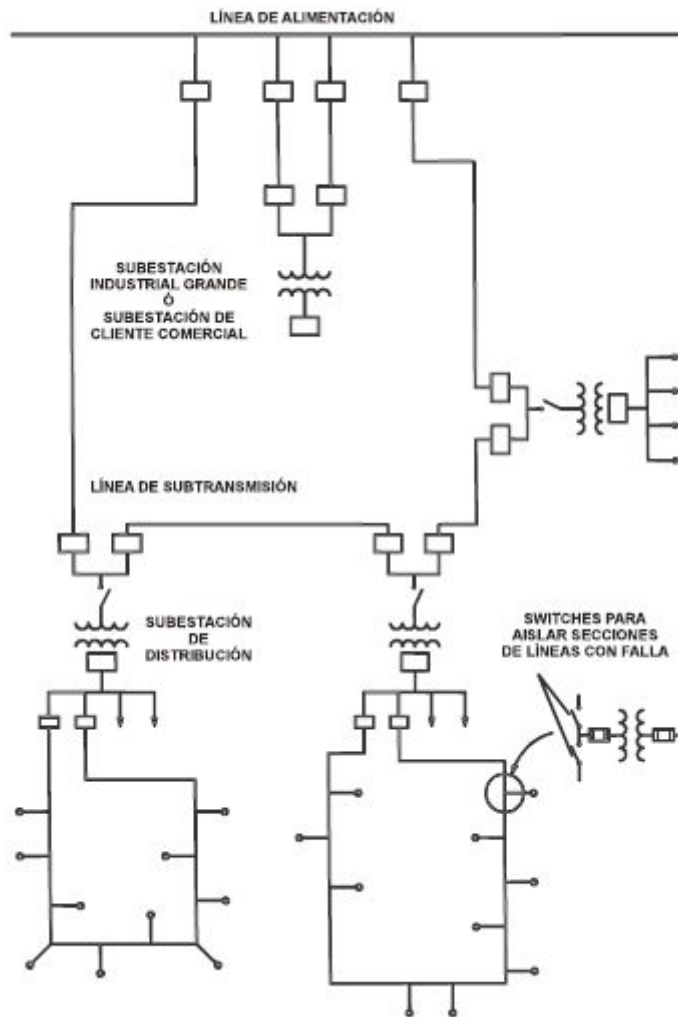


Figura 2.7.2.2.1: Red en lazo ejemplo [30].

2.7.2.3. Red lazo-radial o mixto

Para proporcionar el servicio a los llamados clientes críticos, se puede adoptar una combinación de los llamados sistemas en malla y radial. La parte radial del sistema alimenta a unos pocos clientes residenciales, que pueden quedar fuera de servicio para cualquier condición de falla.

También son utilizados en la red pública donde hay zonas con mayor prioridad ante un corte, como por ejemplo en una zona que hay un hospital y no pueden estar tanto tiempo sin el servicio de energía eléctrica. Por lo tanto la misma está conectada a la red por medio de un lazo para dar soluciones alternativas ante un problema en el suministro de energía.

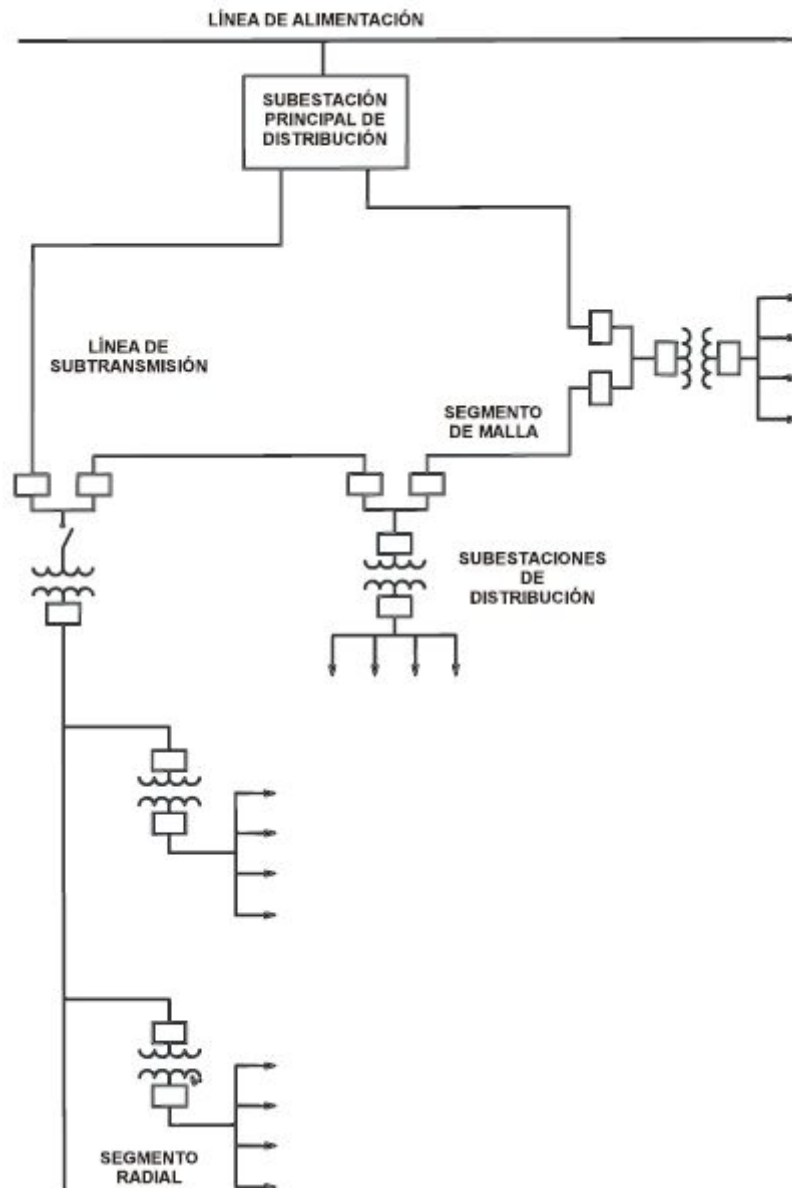


Figura 2.7.2.3.1: Ejemplo de una red mixta [30].

2.8. Conclusión

Durante este capítulo se describió el modelado del NCA y el proceso de instanciación e integración en un DMS. Por un lado al utilizar un modelo CIM (2.1) que impone un estándar, facilita compartir e integrar información de un modelo de datos común entre distintos módulos. Luego, fue necesario instanciar y persistir el modelo, para lo cual se utilizó un grafo RDF (2.2) y RDF/XML (2.4) respectivamente. Posteriormente, para la accesibilidad de los modelos RDF se utilizó JENA (2.3), el cual provee un framework para manejar y administrar datos representados con grafos RDF.

Por último, para la implementación del NCA se decidió utilizar un Grafo (2.5) de representación del sistema eléctrico. Para realizar los recorridos necesarios para ejecutar la funcionalidad del NCA se utilizó el algoritmo de búsqueda DFS(Depth-First Search) (2.5.1) que permite obtener las zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3) de manera intuitiva y con un costo relativamente bajo.

Capítulo 3

Diseño e implementación

En este capítulo se exponen las decisiones de diseño que fueron aplicadas durante el desarrollo del NCA (Network Connectivity Analysis). Para ello fue necesario considerar como entrada el modelo CIM de la red eléctrica (2.1) generado desde un módulo Editor de la red.

Finalmente, se realizó la implementación de la aplicación NCA, que permite al operador conocer situaciones de interés respecto al estado de funcionamiento de la red de distribución, indicando por ejemplo estado de los interruptores (2.6), lazos (2.7.1.2) y paralelos (2.7.1.3).

3.1. Modificación del modelo CIM desde el Editor

Para el modelo CIM de la red se utilizó una herramienta desarrollada en el proyecto TENERGIA. Dicha herramienta es un Editor de la red que permite crear un esquema Unifilar de la misma o abrir uno ya existente. El esquema resultante se guarda en un archivo “.cim” correspondiente al XML/RDF del modelo CIM (2.1).

En este caso el Editor se utilizó para cambiar los estados de los interruptores y guardar los cambios, para luego desde el NCA analizar zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3) dentro de la red eléctrica.

En la Figura 1 se puede ver que por cada cambio en el modelo CIM estático, el procesador topológico debe calcular nuevamente el grafo que representa el modelo, y de manera similar el NCA con las zonas desenergizadas, lazos y paralelos.

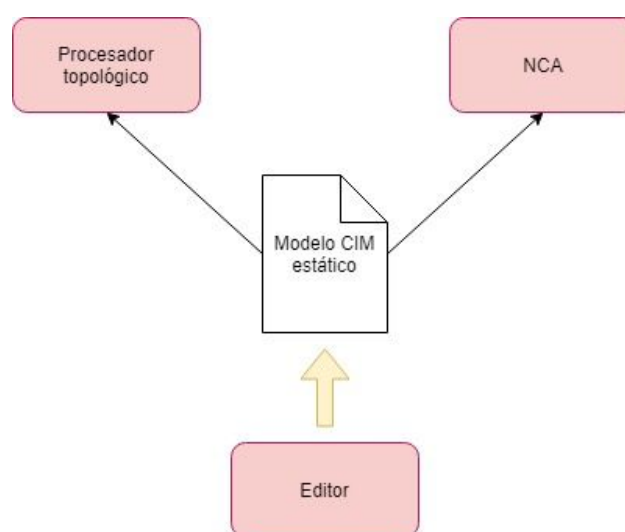


Figura 3.1.1: Interacción entre el Editor, el modelo CIM estático, el Procesador topológico y el NCA.

3.2. Creación de grafos a partir del modelo CIM

Con el fin de instanciar el modelo CIM almacenado en un XML/RDF a una clase java se eligió el framework Jena. En este caso se utilizó una API para eclipse, como se explicó en 2.3, y provee la funcionalidad necesaria para manejar y administrar datos representados en el grafo RDF.

Dentro del proyecto TENERGIA se implementa el modelo sobre la clase Model, la cual contiene los métodos necesarios para instanciar el modelo CIM. En la Figura 3.2.1 está representada la clase Model con todos los atributos y métodos.

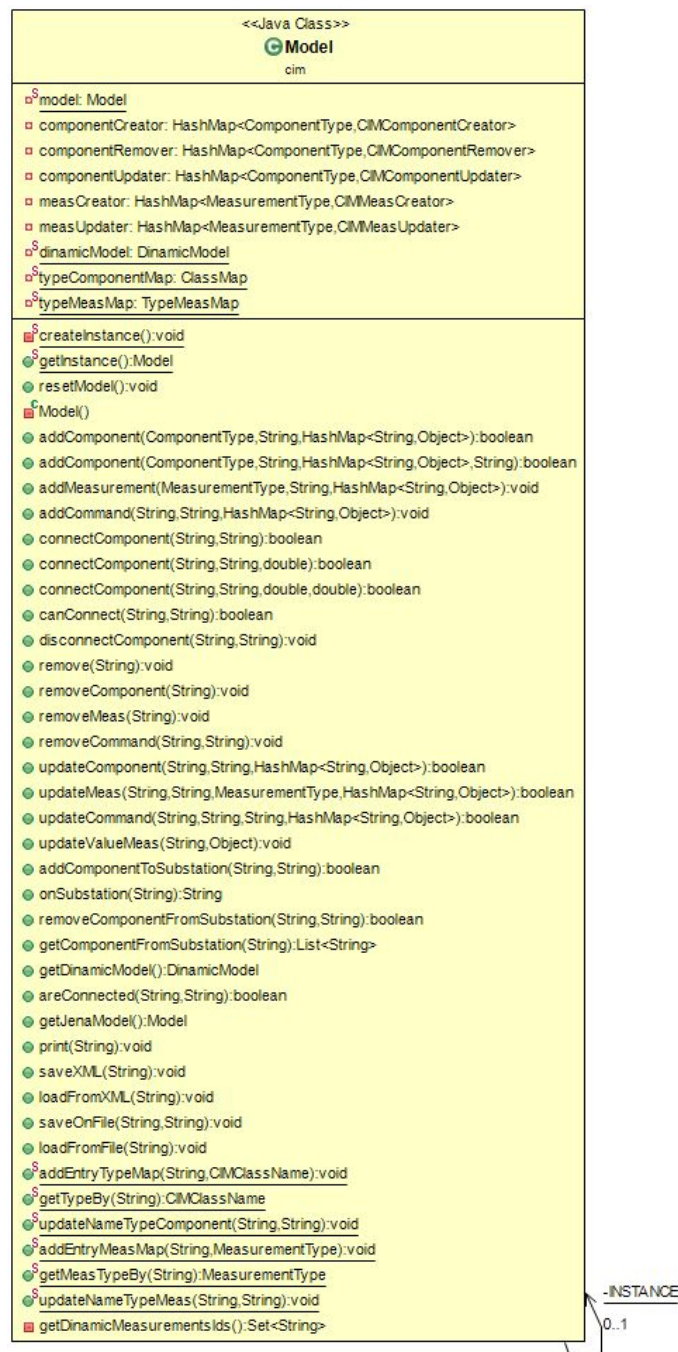


Figura 3.2.1: Clase Java Model.

Además, la clase abstracta `ComponentsConsultorData` define el acceso a los componentes de la red, y que se implementan en las clases `ConsultorData`. Por ejemplo, si se quieren obtener todas las barras del modelo se utiliza una instancia de la clase `BarraConsultorData` que la misma hereda de `ComponentsConsultorData` (Figura 3.2.2). Por último, la clase `BarraConsultorData` utiliza una instancia de la clase `SPARQLManager` para ejecutar las queries necesarias para obtener información relevante en la construcción de los grafos, en particular de todas las barras del sistema de energía eléctrica. En la Figura 3.2.3 está representada la clase `SPARQLManager`.

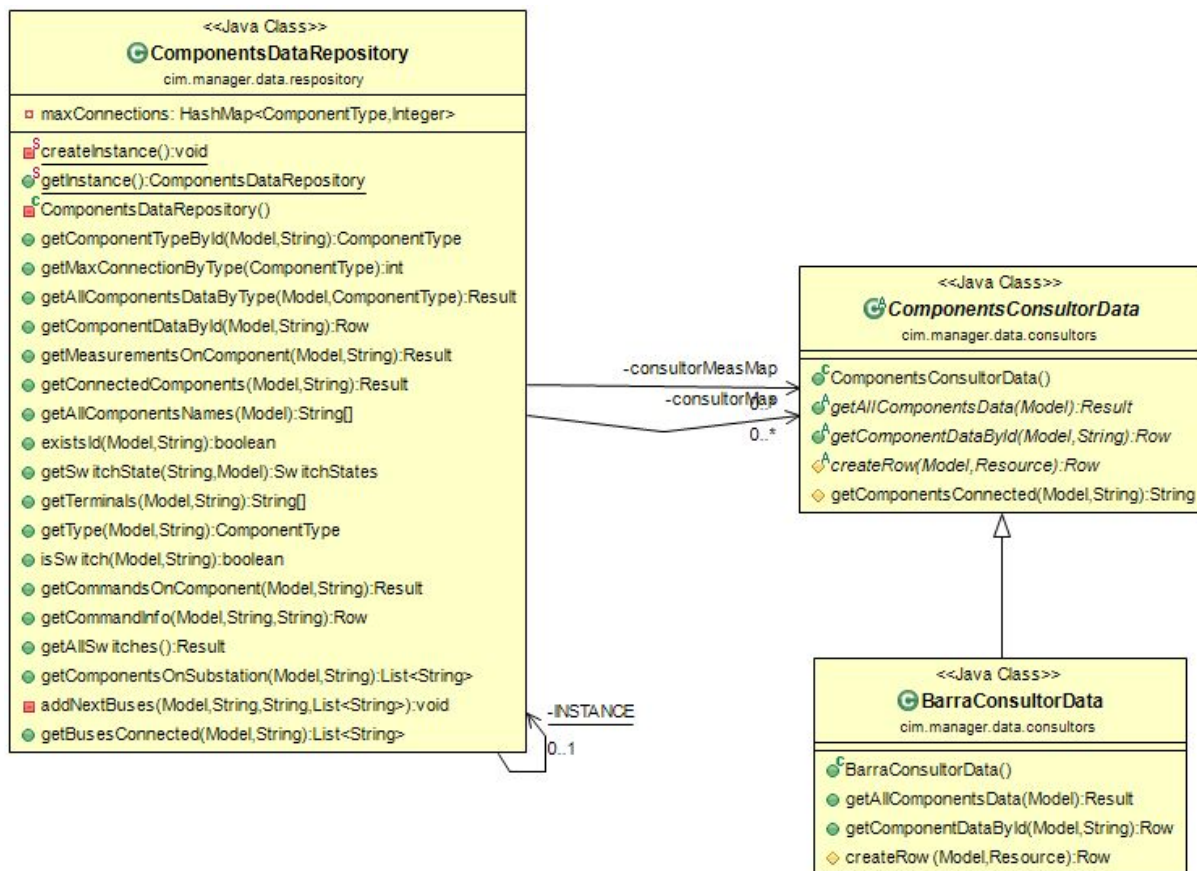


Figura 3.2.2: Clases Java `ComponentsDataRepository`, `ComponentsConsultorData` y `BarraConsultorData`.



Figura 3.2.3: Clase Java `SPARQLManager`.

3.2.1. Creación de grafos basado en el patrón Abstract Factory

Inicialmente, para poder calcular zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3), surgió la necesidad de representar el sistema de energía eléctrica en un grafo no dirigido (2.5), ya que se puede abstraer el problema utilizando este tipo de estructura y brinda facilidad para calcular cada funcionalidad que provee el NCA. Por ejemplo, para el caso de zonas desenergizadas, se puede recorrer el grafo para determinar qué nodos tienen fuentes de energía y los nodos vecinos que la proveen.

Para crear en primer instancia un grafo completo y luego a partir de éste uno reducido se optó por implementar el patrón de diseño Abstract Factory. El mismo agrupa un conjunto de clases que tienen un funcionamiento en común llamadas familias, las cuales son creadas mediante un Factory [21].

Se puede observar en la Figura 3.2.1.1 la estructura de dicho patrón de diseño.

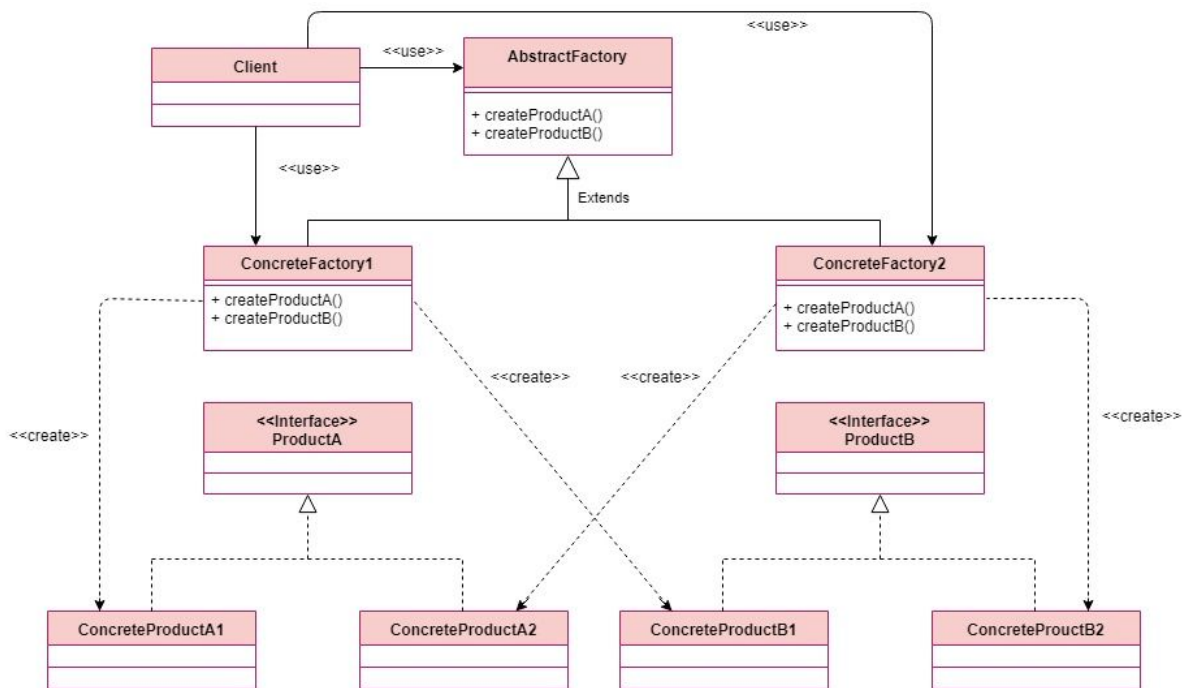


Figura 3.2.1.1: Diagrama de clases del patrón de diseño Abstract Factory.

Por un lado, el mismo brinda flexibilidad al aislar a las clases concretas, y a su vez provee facilidad a la hora de cambiar las familias de productos. Pero por otro lado, si se desea agregar nuevos productos, se deben modificar tanto las fábricas abstractas como las concretas [22].

En particular para el procesador topológico se creó la interfaz **GraphFactory** que contiene el método `createGraph`. Tanto la clase **GraphEPSFactory** como la clase **ReducedGraphEPSFactory** van a implementar la interfaz **GraphFactory**; la primera tiene implementado el método `createGraph` con el fin de crear un grafo completo a partir del modelo CIM, y el objetivo de la segunda es crear un grafo reducido a partir

del grafo completo eliminando interruptores cerrados. El diagrama de clases se puede ver en la Figura 3.2.1.2.

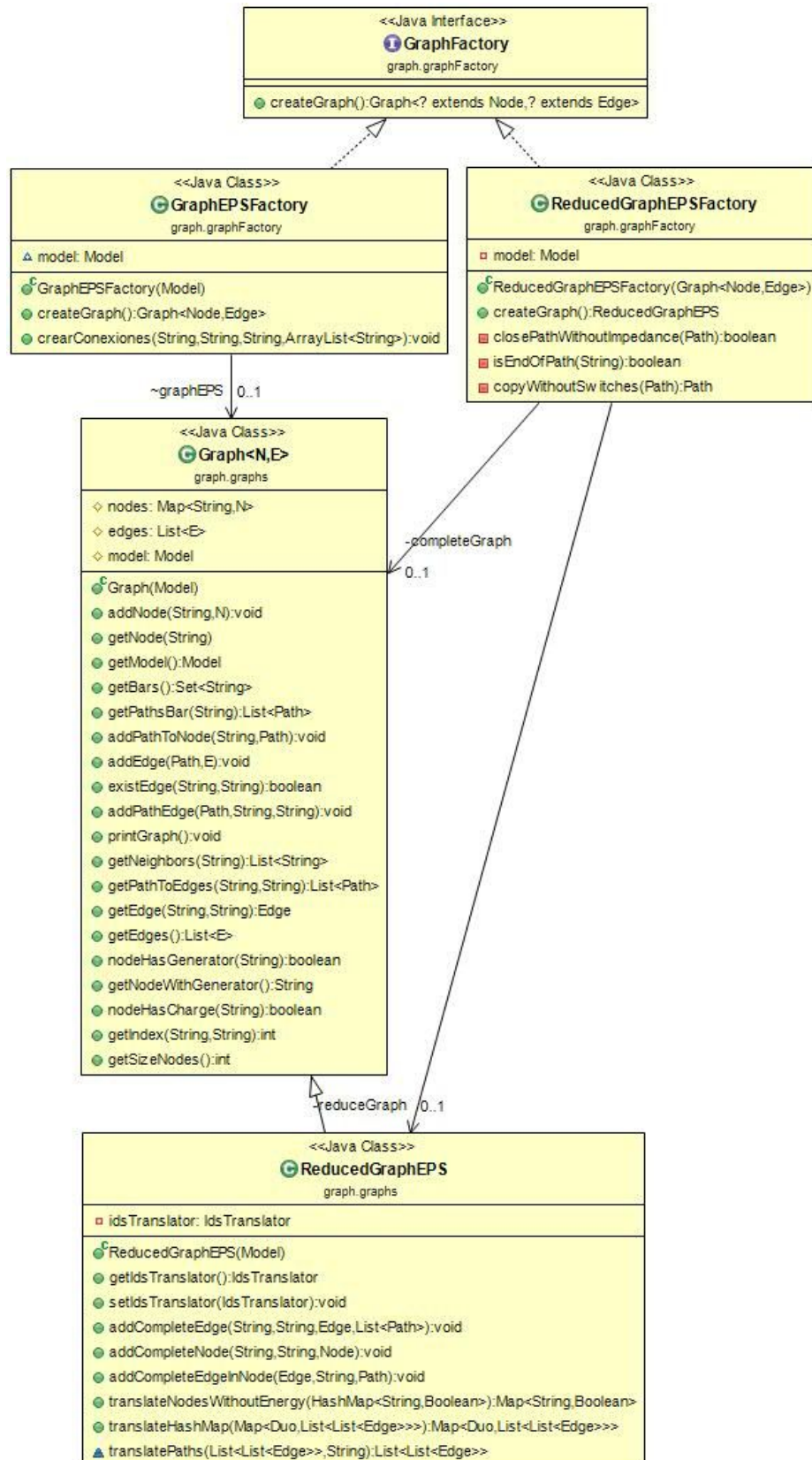


Figura 3.2.1.2: Diagrama de clases donde se ve reflejado el patrón AbstractFactory en el procesador topológico.

3.2.2. Creación del grafo completo

Como se mencionó anteriormente, la creación del grafo completo se realiza a través de GraphEPSFactory. En primer lugar cada nodo del grafo quedó definido por una barra perteneciente a la red con todos sus elementos conectados. Para ilustrar esto en la Figura 3.2.2.3 se tiene la Barra1 con sus elementos conectados Interruptor1 y Generador1.

Luego, para definir los arcos entre los nodos se creó un camino de componentes. Así, en el ejemplo de la Figura 3.2.2.3 el arco que existe entre la Barra 1 y la Barra 2 va a estar formado por la Linea1 y el Interruptor5.

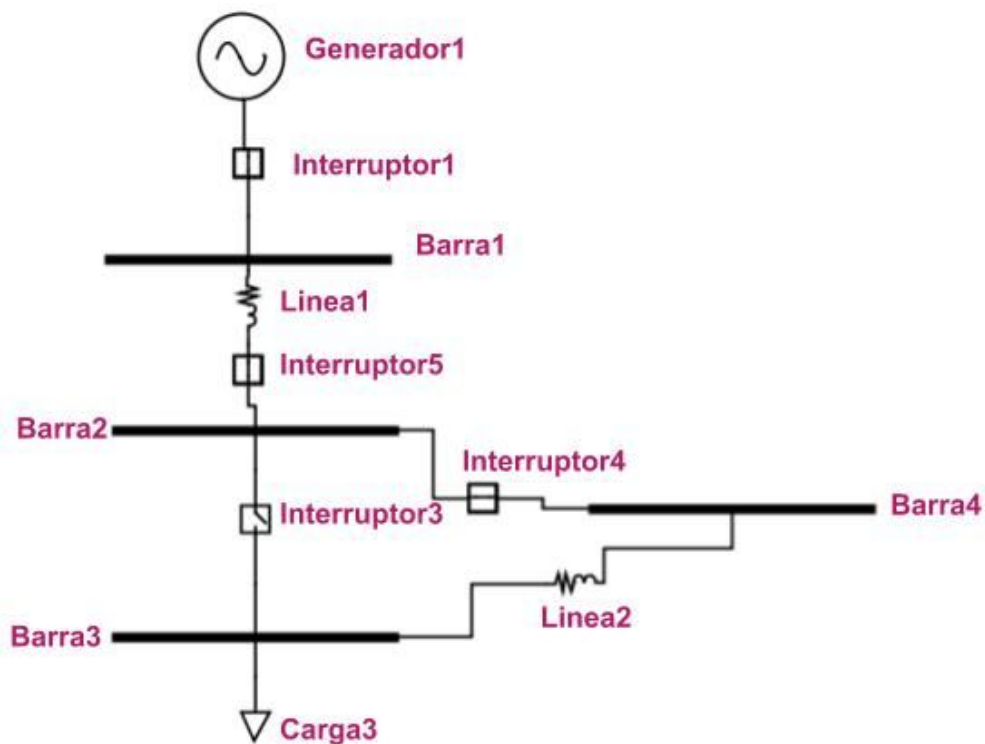


Figura 3.2.2.3: Circuito ejemplo para la construcción del grafo completo.

La clase Node contiene una lista de caminos. Para éstos caminos se definió la clase Path que está formada por una lista de Strings que hacen referencia a los componentes, como por ejemplo para la Figura 3.2.2.3 existe un nodo para la Barra1 que tiene un Path formado por el Interruptor1 y el Generador1.

Por otro lado se definió la clase Edge, la cual hace referencia al arco entre dos nodos. La misma está formada por dos Strings, que son el id de cada Barra que conecta el arco, y a su vez una lista de Path que son los distintos caminos de componentes que se conectan las barras.

En la Figura 3.2.2.4 se muestra el diagrama de clases para la construcción de un grafo completo.

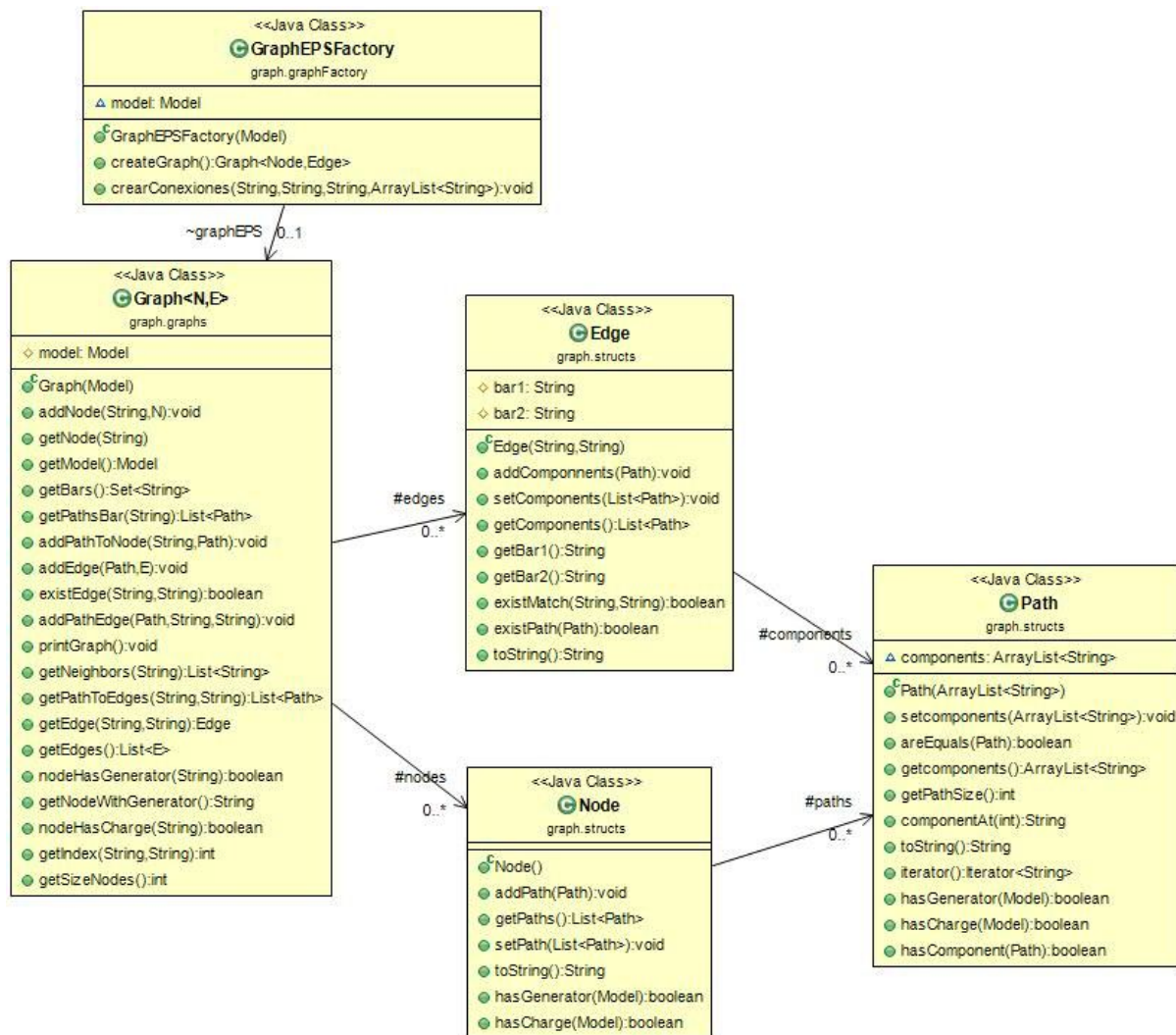


Figura 3.2.2.4: Diagrama de clases para el grafo completo.

3.2.3. Creación del grafo reducido

El siguiente punto es crear un grafo reducido a partir del completo. El objetivo de la construcción del mismo es reflejar el estado de los interruptores, y para ello se definieron las siguientes situaciones.

En primer lugar cuando se tienen dos Barras conectadas por un interruptor cerrado, en el grafo reducido estas dos barras se convierte en una sola y el interruptor desaparece. Por ejemplo en el circuito de la Figura 3.2.2.3, como la Barra2 y la Barra4 están conectadas por el Interruptor4 que se encuentra cerrado, las mismas se reemplazan por la Barra2_Fic y el interruptor desaparece del circuito.

En segundo lugar, por cada interruptor cerrado que haya en algún camino desaparece para el grafo reducido. Es decir que en el ejemplo de la Figura 3.2.2.3, los interruptores 1 y 5 desaparecen luego de construir el grafo reducido.

Por último, si hay un interruptor abierto entre dos barras desaparece el arco que existe entre ellas. Para el caso de la Figura 3.2.2.3 cuando se construye el grafo

reducido deja de existir el arco formado por el Interruptor3 que conecta las barras 2 y 3.

Entonces, teniendo en cuenta el ejemplo de la Figura 3.2.2.3 luego de calcular el grafo reducido, el circuito queda como el que se ve en la Figura 3.2.3.1.

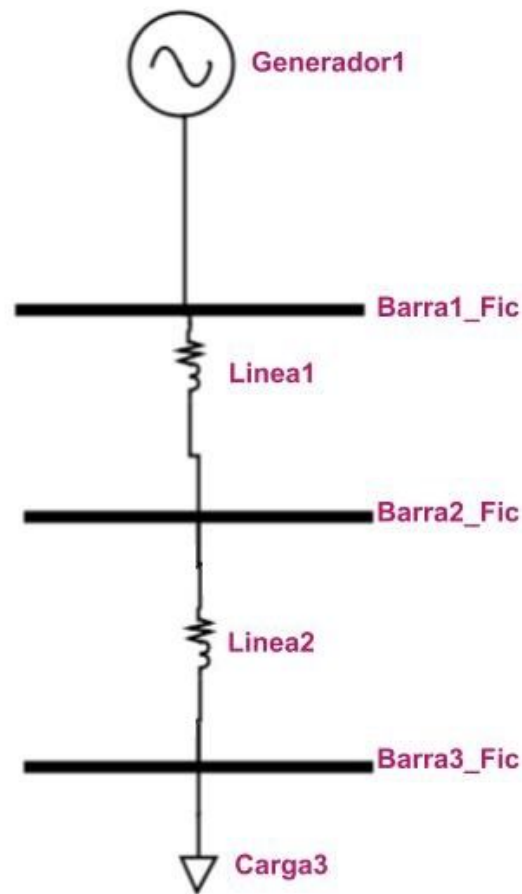


Figura 3.2.3.1: Circuito ejemplo luego de calcular el grafo reducido.

Finalmente, para la construcción del grafo reducido y la aplicación de los resultados sobre el grafo completo se decidió crear la clase `ReducedGraphEPS`, compuesta por `ReducedNode` y `ReducedEdge`. El propósito de estas dos últimas es almacenar los datos necesarios para volver al circuito original del cual se partió. Ambas clases heredan de `Node` y `Edge` respectivamente.

Como se puede observar en la Figura 3.2.3.2 dentro del `ReducedNode` hay una instancia de `Graph`, ya que para el caso que se unan dos barras por estar conectadas por un interruptor cerrado, se debe almacenar a que nodos originales pertenecían y por cuales arcos estaban conectados.

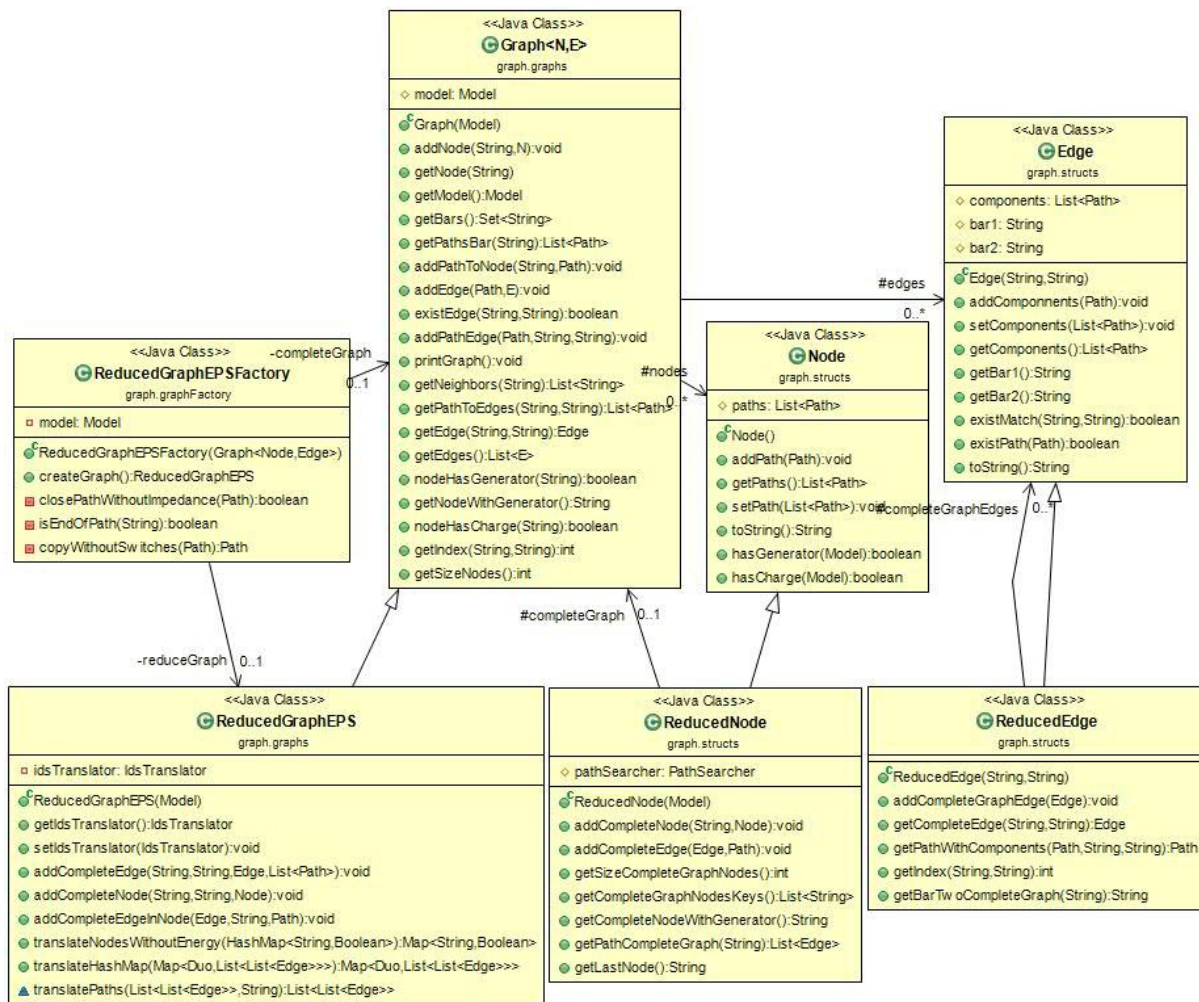


Figura 3.2.3.2: Diagrama de clases para la creación del grafo reducido.

3.3. Utilización del patrón Strategy para búsquedas sobre el grafo reducido

Una vez obtenido el grafo reducido el siguiente objetivo fue buscar zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3) sobre la red de energía eléctrica. Para dicha solución se utilizó el patrón de diseño Strategy.

Se eligió dicho patrón debido a que define un conjunto de algoritmos, encapsula cada uno de ellos y los hace intercambiables. Además, permite que el algoritmo pueda variar independientemente de los clientes que lo utilicen [28]. Las ventajas que ofrece son:

- El contexto es más fácil de entender, modificar y mantener.
- Evita la duplicación de código y la explosión de subclases.
- Cumple con el principio de responsabilidad única, es decir que un objeto debería tener una única responsabilidad.
- También cumple con el principio de abierto/cerrado, significa que las “entidades de software deben estar abiertas para su extensión, pero cerradas para su modificación”.

Se puede observar el diagrama de clases para dicho patrón en la Figura 3.3.1.

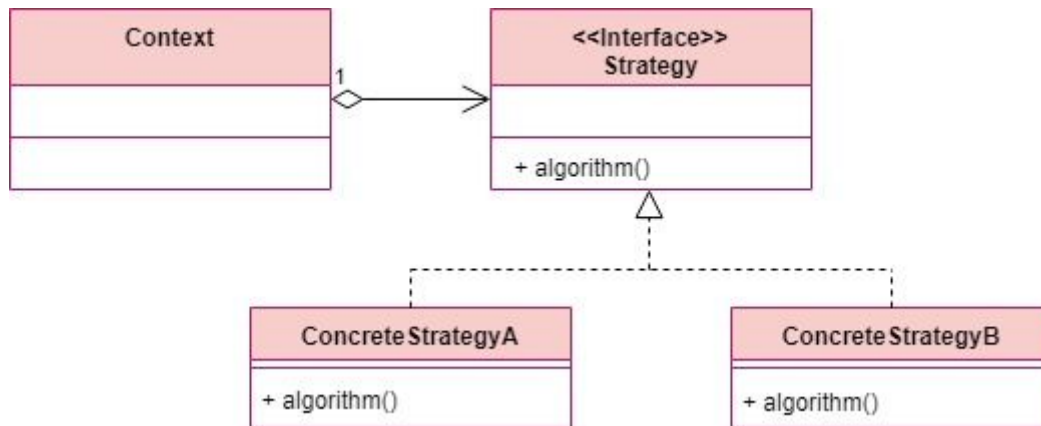


Figura 3.3.1: Diagrama de clases para el patrón de diseño Strategy.

Como ya se mencionó, para poder buscar zonas desenergizadas, lazos y paralelos se trabaja sobre el grafo reducido. Para esto se optó por realizar una búsqueda en profundidad (Depth First Search) (2.5.1), considerando en cada caso de detección distintos criterios de búsqueda.

En consecuencia se creó una clase abstracta llamada **Searcher** la cual contiene los métodos **search()** y **dfs()** abstractos. Estos métodos los implementan las clases que heredan de la misma según el objetivo de la búsqueda. A su vez, ésta clase tiene la variable “**graph**” sobre la cual se van a realizar las búsquedas, y una variable “**visited**” encargada de tener registrado los nodos que se visitan durante el algoritmo de búsqueda. De la misma heredan las clases **ZoneWhithoutEnergySearcher**, **LoopSearcher** y **ParalelSearcher**, las cuales tienen implementados los métodos abstractos mencionados. Se puede observar dichas clases en el diagrama de la Figura 3.3.2.

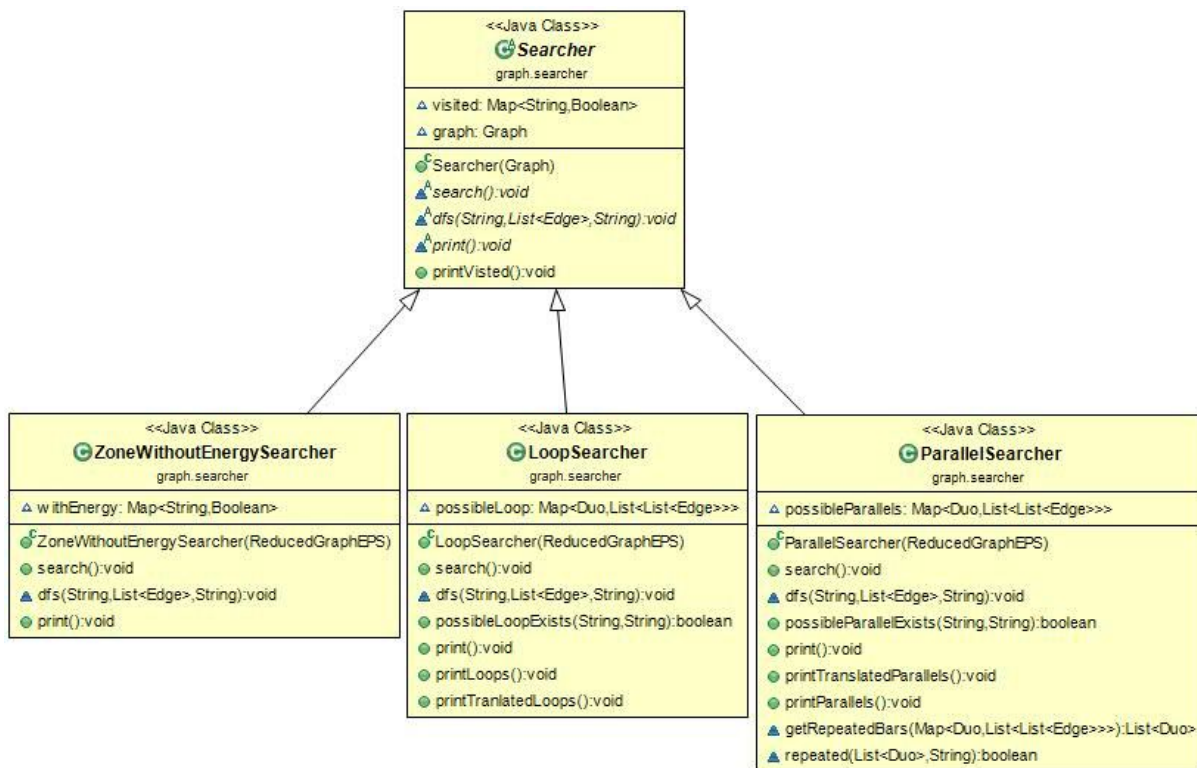


Figura 3.3.2: Diagrama de clases para los distintos algoritmos de búsqueda en el grafo reducido.

3.3.1. Búsqueda de zonas desenergizadas

Uno de los objetivos principales del NCA es poder encontrar zonas desenergizadas (2.7.1.1); para ello la clase encargada de realizar la búsqueda es ZoneWhithoutEnergySearcher.

Dentro de ésta clase se tiene un Map encargado de almacenar como clave un string que hace referencia al id de la barra/nodo, y un valor booleano para determinar si esa barra/nodo tiene energía. Esta estructura se actualiza durante el recorrido en el grafo.

Sabiendo que la misma hereda de Searcher, implementa los métodos search() y dfs() que son los necesarios para recorrer el grafo y poder calcular las zonas desenergizadas.

```

search( ){
    Por cada nodo v del grafo {
        si no esta visitado y tiene un generador asociado{
            dfs(v);
        }
    }
}

dfs(Node v){

```

```

    Marcar como visitado a v;
    Marcar que tiene energía v;
    Por cada nodo w vecino de v{
        Si no esta visitado{
            dfs(w);
        }
    }
}

```

Para visualizar el funcionamiento de los métodos search y dfs se crearon los diagramas de actividades correspondientes (Figuras 3.3.1.1 y 3.3.1.2 respectivamente). El objetivo de dichos métodos es recorrer el grafo de manera recursiva, y así ir marcando aquellos nodos que cuentan con energía. Pero aquellos nodos que no son alcanzados por ningún camino, jamás llegarán a ser visitados y por lo tanto formarán parte del conjunto de nodos que hacen referencia a las zonas desenergizadas.

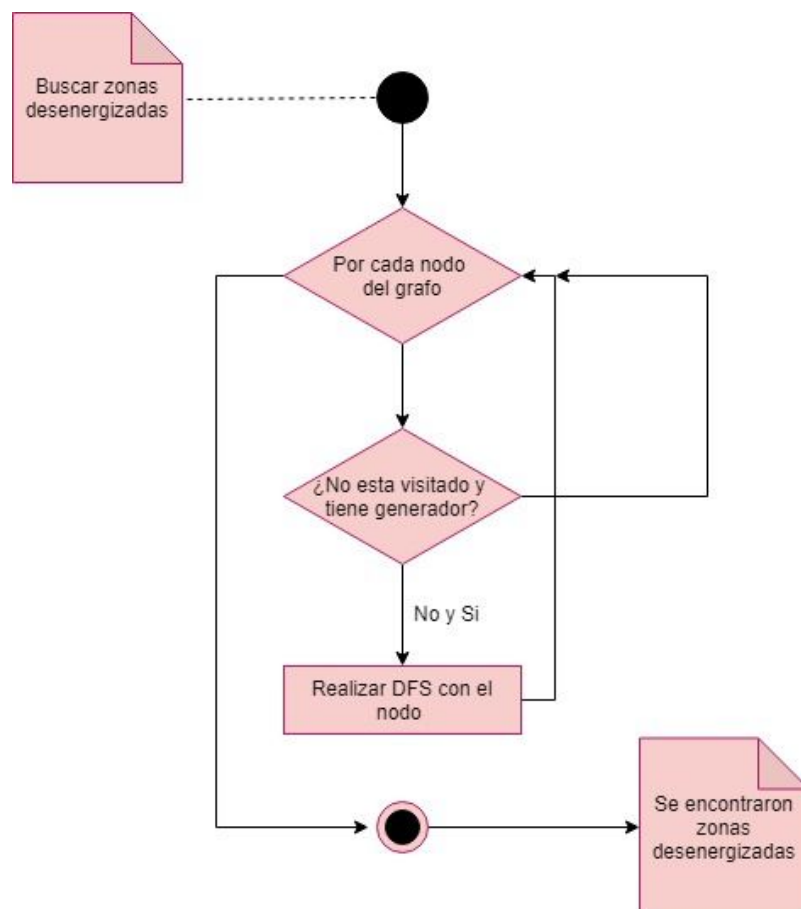


Figura 3.3.1.1: Diagrama de actividades del método search para zonas desenergizadas.

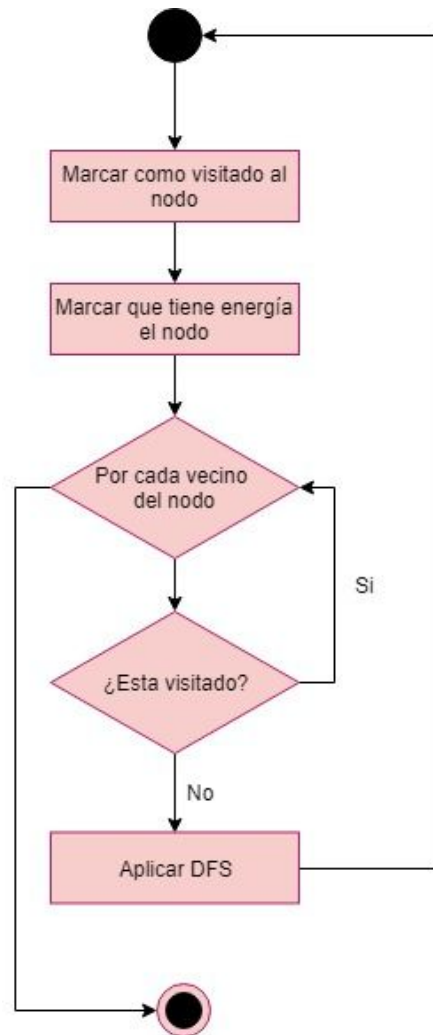


Figura 3.3.1.2: Diagrama de actividades del método dfs para zonas desenergizadas.

En este caso para la búsqueda de zonas desenergizadas la complejidad es $O(m)$, siendo m el máximo entre el número de nodos y el número de arcos. La demostración se encuentra en el Capítulo 2 (2.5.1) cuando se describe el algoritmo DFS.

3.3.2. Búsqueda de lazos

Con el propósito de encontrar los lazos (2.7.1.2) dentro de la red de energía eléctrica se inventó la clase LoopSearcher. Como ya se mencionó la misma hereda de Searcher (Figura 3.3.2), por lo tanto debe implementar los métodos search y dfs. Para calcular los lazos es necesario almacenar todos los caminos entre una barra que tenga un generador asociado y las demás barras. Entonces dentro de la clase LoopSearcher se define la variable possibleLoops. La misma es un Map que utiliza como clave una variable de tipo Duo y almacena una lista de una lista de Edge. La clase Duo almacena el id de la barra que tiene el generador y el id de la barra que está comunicada con la primera por algún camino. Por otro lado, se tiene una lista

de Edge, que permite acceder a los arcos que hay en el camino entre el nodo que tiene un generador asociado y un nodo al cual le llega energía del primero. A su vez, como puede haber varios caminos se implementó una lista de los posibles caminos.

```
search( ){
    Por cada nodo v del grafo {
        Inicializar la lista camino;
        Si no esta visitado y tiene un generador asociado{
            dfs(v, camino, v);
        }
    }
}

dfs(Node v, Path camino, Node vConGenerador){
    Marcar como visitado el nodo v;
    Si el nodo v no es igual a vConGenerador{
        Crar el Duo d con vConGenerador y v;
        Agregar al Map el camino de d;
    }
    Por cada nodo w vecino de v{
        Si no esta visitado w{
            Por cada arco entre v y w{
                Agregar al camino el arco de v con w;
                dfs(w, camino, vConGenerador);
                Eliminar del camino el arco de v con w;
            }
        }
    }
    Marcar como no visitado el nodo v;
}
```

Los diagramas de actividades para los métodos search y dfs representados en las figuras 3.3.2.1 y 3.3.2.2 respectivamente.

Hay que tener en cuenta que dichos métodos encuentran los posibles lazos, es decir que se deben buscar dentro de los “duos” del Map aquellos que tengan más de una lista de arcos. Considerar que un lazo se forma cuando un nodo es alimentado por la misma fuente de energía a través de distintos caminos.

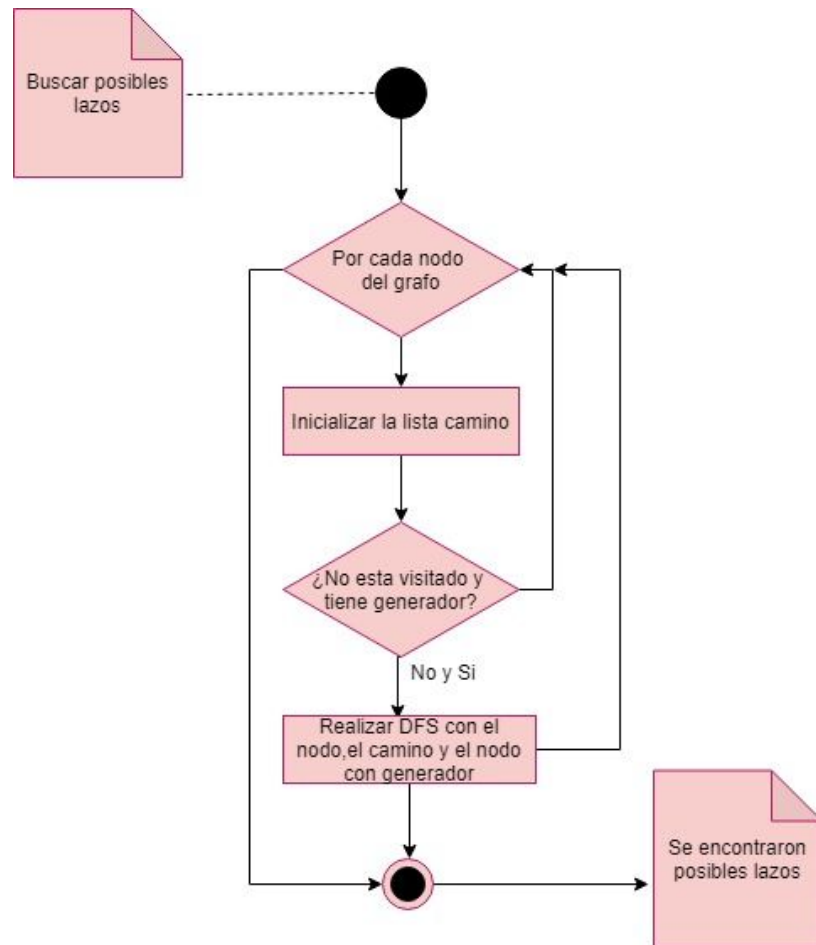


Figura 3.3.2.1: Diagrama de actividades del método search para búsqueda de lazos.

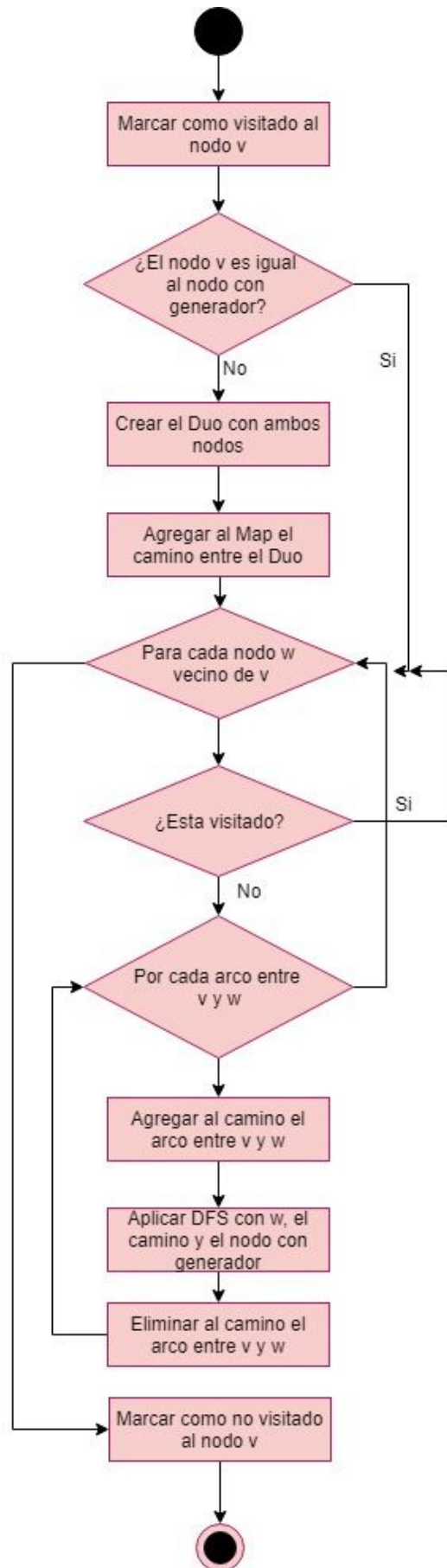


Figura 3.3.2.2: Diagrama de actividades del método dfs para búsqueda de lazos.

Con el propósito de calcular la complejidad temporal se tuvo en cuenta lo siguiente:

Sean $G = (V, E)$, $|E| = e$, $|V| = n$, $m = \max(n, e)$, $g = \text{número de nodos con generador}$

Suponiendo el peor de los casos que g sea igual al número de nodos

$$T_{DFS}(g) \in O(nm)$$

Demostración:

$$\text{Sea } T_{DFS}(n, e) \in O(m)$$

Entonces

$$T_{DFS}(g) \leq \sum_{k=0}^{n-1} c(DFS(g)) = c(nm)$$

$$T_{DFS}(g) \in O(nm)$$

Por lo tanto la complejidad temporal para la búsqueda de lazos es $O(nm)$, siendo n el total de nodos, y m el máximo entre la cantidad de nodos y cantidad de arcos.

3.3.3. Búsqueda de paralelos

Para la búsqueda de paralelos (2.7.1.3) sobre la red se creó la clase `ParallelSearcher`, que hereda de `Searcher` (Figura 3.3.2), y por lo tanto implementa los métodos `search` y `dfs`.

De forma similar a la búsqueda de lazos, para encontrar paralelos se necesita almacenar todos los caminos entre una barra que tenga un generador asociado y las barras restantes. Por tal motivo, dentro de la clase `ParallelSearcher` se encuentra la variable `possibleParallels`. Esta variable tiene las mismas características que `possibleLoop` (3.3.2); es decir, la estructura es un `Map` que utiliza como clave una variable de tipo `Duo` y almacena una lista de una lista de `Edge`. La clase `Duo` almacena el id de la barra que tiene el generador asociado y el id de la barra que está comunicada con la primera por algún camino.

A su vez la lista de `Edge` representa todos los arcos que hay en el camino entre el nodo que tiene un generador asociado y un nodo al cual le llega energía del primero.

```
search( ){
    Por cada nodo v del grafo {
        Inicializar la lista camino;
        Si no esta visitado y tiene un generador asociado{
            dfs(v, camino, v);
        }
    }
}
```

```

dfs(Node v, Path camino, Node vConGenerador){
    Marcar como visitado el nodo v;
    Crar el Duo d con vConGenerador y v;
    Agregar al Map el camino de d;
    Por cada nodo w vecino de v{
        Si no esta visitado w{
            Por cada arco entre v y w{
                Agregar al camino el arco de v con w;
                dfs(w,camino,vConGenerador);
                Eliminar del camino el arco de v con w;
            }
        }
    }
    Marcar como no visitado el nodo v;
}

```

En comparación con la búsqueda de lazos, los métodos search son iguales en ambos casos. No obstante los métodos dfs tienen una pequeña diferencia. Para este caso no es necesario verificar que el nodo actual sea distinto que el nodo con generador; esto es debido a que se necesita almacenar todas las soluciones para determinar los paralelos. Los diagramas de actividades para los métodos mencionados se esquematizan en las figuras 3.3.3.1 y 3.3.3.2 respectivamente.

Dado que el método search encuentra posibles paralelos, luego de realizar dicha búsqueda se recorre el Map y se determina cuáles son los paralelos. Para esto se tiene en cuenta que haya dos “duos” que tengan como segundo nodo el mismo id de barra y como primer nodo distinto id de barra, es decir que el primer nodo es la barra que tiene una fuente de energía asociado y el segundo nodo es al que le proveen energía. En conclusión se obtienen los paralelos representados por tres barras, dos que hacen referencia a las barras con la fuente de energía asociado, y la tercer barra a la cual le llega energía de las dos primeras por algún camino de componentes.

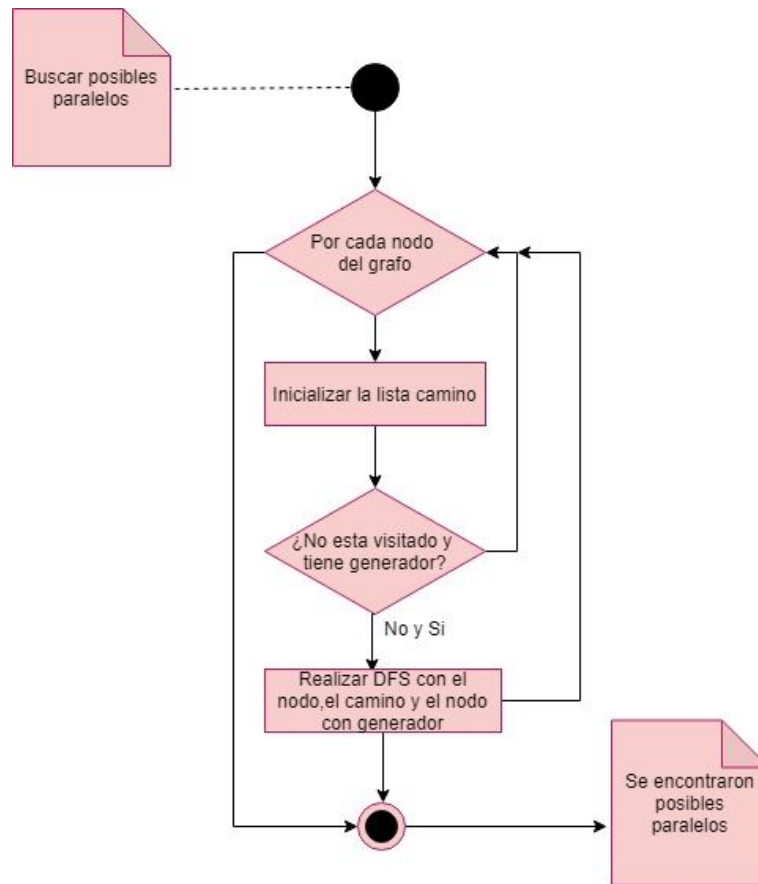


Figura 3.3.3.1: Diagrama de actividades del método search para búsqueda de paralelos.

3.4. Conversión de los resultados al grafo original

Considerando que los resultados obtenidos están contruidos a partir del grafo reducido, y el mismo está formado por barras ficticias y sin interruptores cerrados, el paso siguiente es convertir las barras ficticias en reales y agregar nuevamente los interruptores a los resultados obtenidos en búsqueda de zonas desenergizadas, lazos y paralelos.

Para esto fue necesario guardar cierta información en el grafo reducido y luego utilizarla en la reconstrucción de los resultados al grafo original.

Primero dentro de los nodos del grafo reducido se guardó un pequeño grafo que contiene las barras reales y cómo estas estaban conectadas. Este es el caso, por ejemplo, de que dos barras estén comunicadas por medio de un interruptor cerrado y se deban unir en una única barra ficticia.

En segundo lugar, dentro de los arcos del reducido se tuvo que guardar la composición de los arcos del grafo original. Por ejemplo, si en el arco había un interruptor cerrado y un transformador, para el grafo reducido no existe ese interruptor cerrado pero en el resultado debe figurar el mismo.

La clase encargada de implementar la conversión al grafo original es `ReducedGraph`. Dentro de la misma hay tres métodos, `translateHashMap`, `translatePaths` y `translateNodesWithoutEnergy` que realizan toda la conversión de los resultados.

Además se creó una clase que extiende de `Searcher` y se llama `PathSearcher`, encargada de buscar los caminos originales a partir del pequeño grafo almacenado dentro del nodo reducido.

En la Figura 3.4.1 se puede ver la herencia de clases asociada a la clase `Searcher`, `ZoneWhithoutEnergySearcher`, `LoopSearcher`, `ParallelSearcher` y `PathSearcher`.

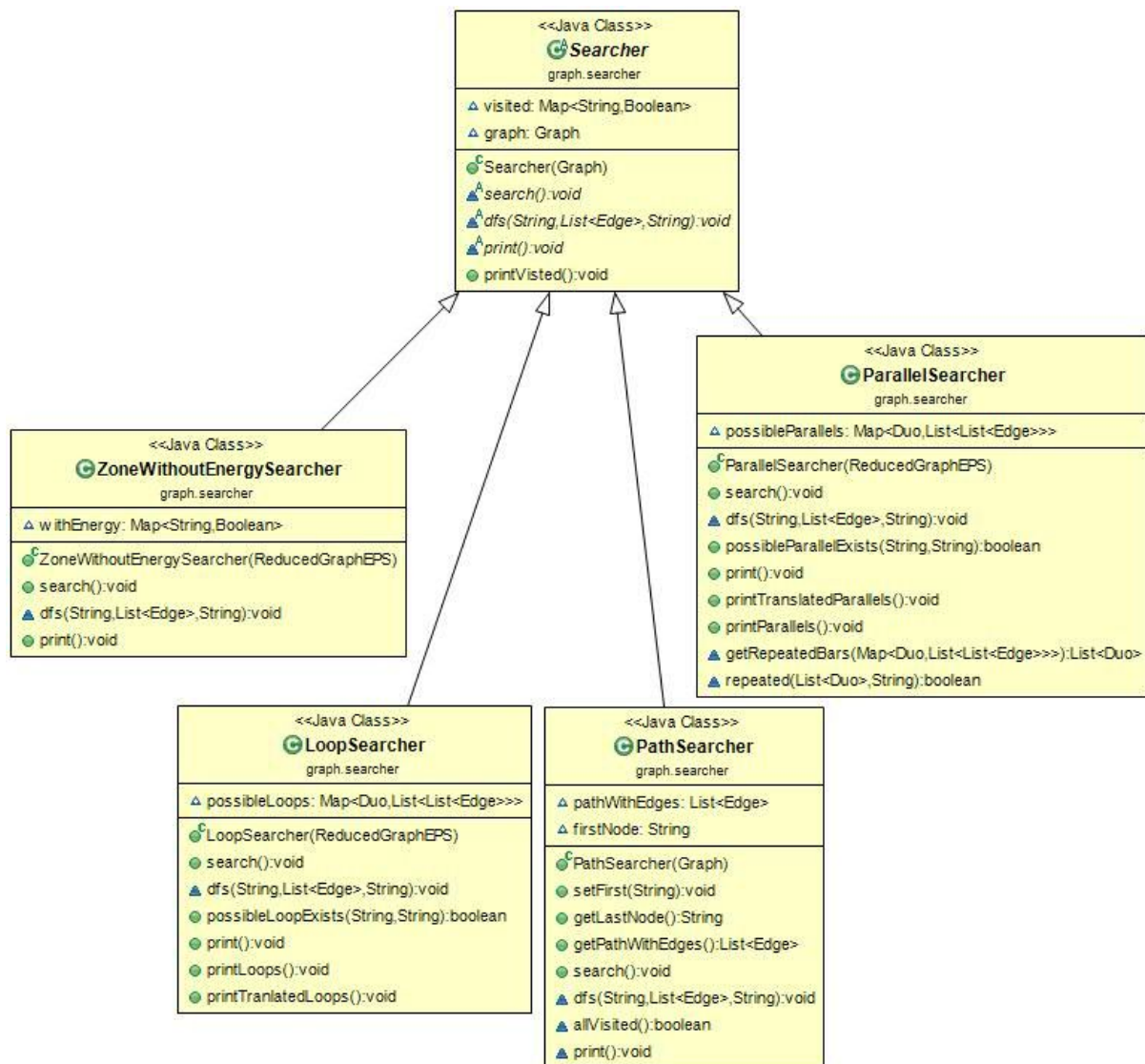


Figura 3.4.1: Diagrama de clases final para Searcher.

3.5. Ejecución de todas las etapas del NCA

En los ítems anteriores se describió cada etapa del NCA, y en este apartado se expone cómo interactúan en tiempo de ejecución.

Se puede observar en las Figuras 3.5.1, 3.5.2 y 3.5.3 el diagrama de secuencia para la búsqueda de zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3) respectivamente.

Primero se instanció cada clase que interviene en las distintas búsquedas. Por un lado se tiene Creador del grafo que es una instancia de la clase GraphEPSFactory, que se encarga de crear una instancia del Grafo según los datos que le provee el Modelo. Por lo tanto, hay un Grafo que es una instancia de la clase Graph y un Modelo que es una instancia de la clase Model. El Modelo tiene todos los datos que le provee el XML/RDF.

Luego de tener la instancia del Grafo creada, se debe crear una instancia de ReducedGraphEPS que en los diagramas de secuencia se denomina Creador del grafo reducido.

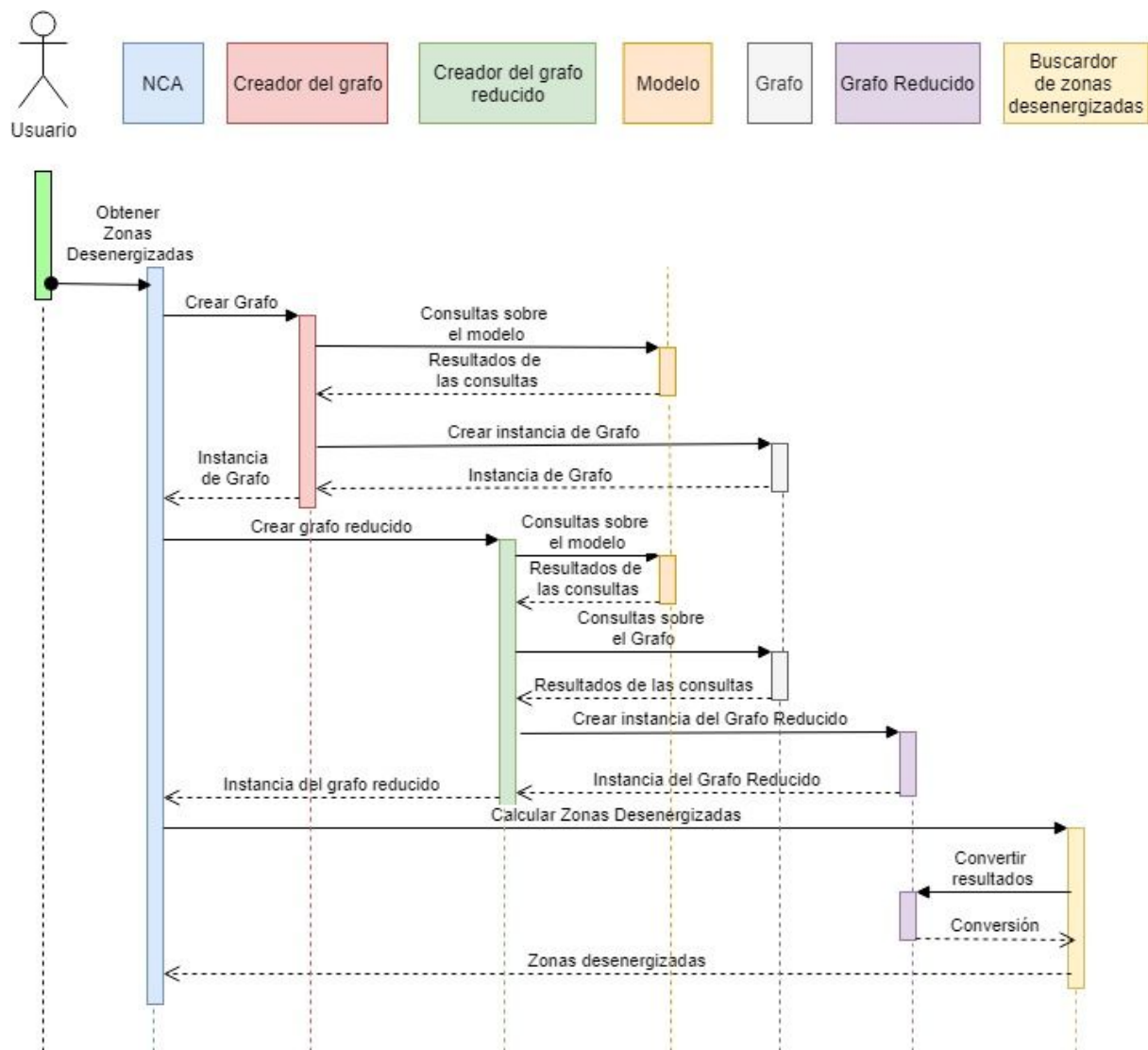


Figura 3.5.1: Diagrama de secuencia para la búsqueda de zonas desenergizadas.

Dicha clase se encarga de la construcción del Grafo reducido realizando consultas al Modelo y al Grafo. Como ya se explicó en la sección 3.2.3, la construcción del grafo reducido consiste en eliminar del sistema de energía eléctrica aquellos interruptores que están cerrados y unir las barras que están comunicadas por medio de interruptores cerrados.

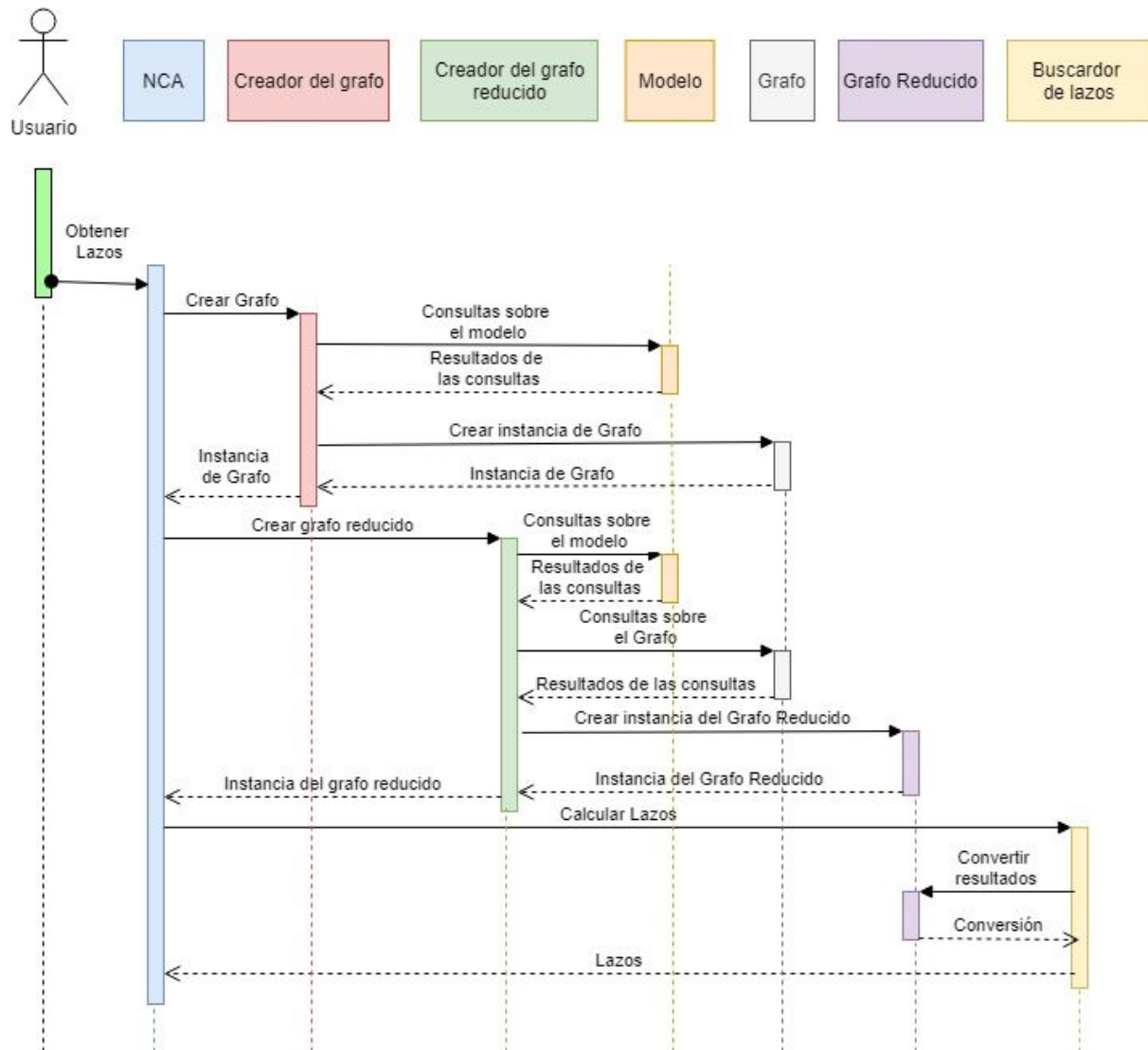


Figura 3.5.2: Diagrama de secuencia para la búsqueda de lazos.

Luego de obtener el grafo reducido se crea una instancia de un buscador, es decir que si se desea buscar zonas desenergizadas se instancia la clase `ZoneWhithoutEnergySearcher`. Por ejemplo en la Figura 3.5.1 la instancia se llama Buscador de zonas desenergizadas; pero si se quiere buscar lazos se creará una instancia de `LoopSearcher` (Figura 3.5.2) denominada Buscador de lazos; y si se pretende buscar paralelos debemos crear una instancia de `ParallelSearcher` (Figura 3.5.3) llamada Buscador de paralelos.

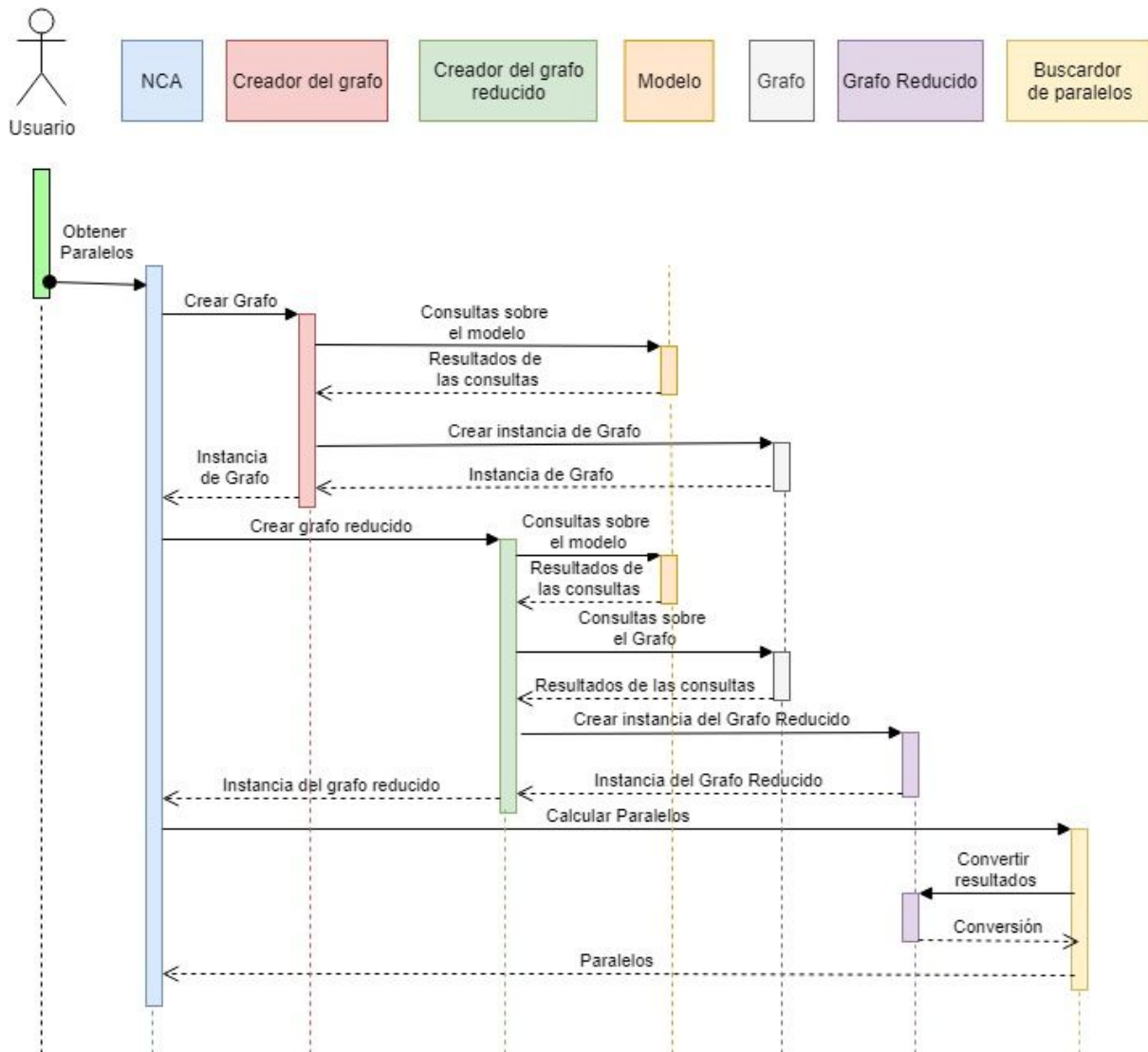


Figura 3.5.3: Diagrama de secuencia para la búsqueda de lazos.

Finalmente estos buscadores utilizarán el grafo reducido para calcular los resultados deseados. Como el grafo reducido se construyó con barras ficticias y se eliminaron los interruptores, el paso final es convertir los resultados obtenidos. El grafo reducido es el encargado de dicha tarea para que luego el buscador retorne los resultados deseados.

3.6. Conclusión

El NCA provee al administrador u operador de la red de energía eléctrica la información de partes relevantes de la misma para facilitar su trabajo; en este caso la determinación de zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3). Para ello se utilizaron patrones de diseño orientados al modelado e implementación de las funcionalidades mencionadas, como por ejemplo Abstract Factory (3.2.1) y Strategy (3.3). Finalmente se construyeron diagramas de clases, de actividades y de secuencia, que permite esquematizar las etapas de desarrollo e

implementación que van desde el modelo original, pasando por las etapas de detección, y finalizando en la reconstrucción de la red junto con la información generada en el proceso de detección.

Capítulo 4

Resultados experimentales

En este capítulo se presentan los experimentos realizados con el fin de dar evidencia empírica sobre las búsquedas que realiza el NCA. El objetivo de los experimentos fue evaluar el rendimiento general de la aplicación e identificar potenciales problemas de la misma.

Durante el análisis se muestran los distintos pasos necesarios para buscar zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3).

4.1. Contexto

Para definir los casos de estudio se trabajó en un sector de la red eléctrica de Armstrong de 13.2kv (Figura 4.1). El acceso a los detalles de la red se logró mediante un convenio que hay entre la cooperativa eléctrica de Armstrong y Pladema-Unicen. El esquema unifilar equivalente de la red referenciada se puede ver en la Figura 4.2.

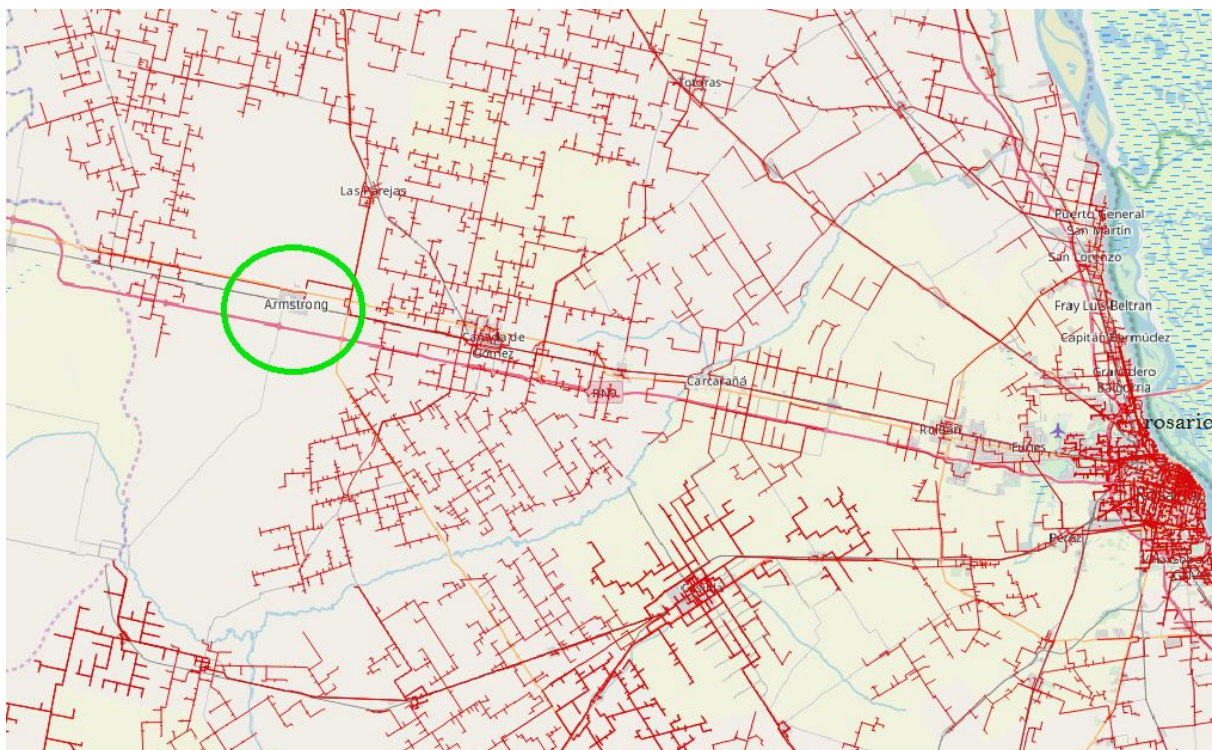


Figura 4.1: Georeferencia de la ciudad de Amstrong.

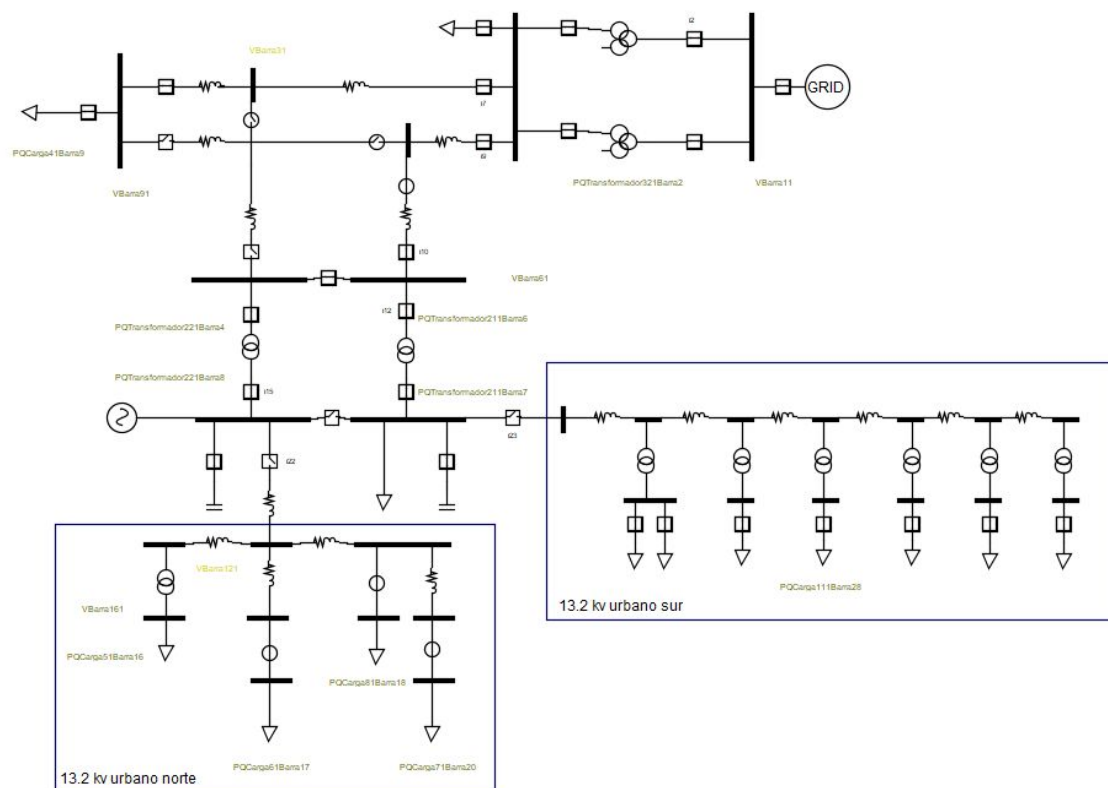






Figura 4.2: Sección de la red Eléctrica de Armstrong.

Dicho esquema unifilar se realizó sobre un Editor desarrollado en el instituto Pladema a través del proyecto TENERGIA. Dentro de las funcionalidades que posee el editor se encuentra la posibilidad de exportar la red eléctrica en un archivo XML. Este archivo tiene como objetivo persistir el modelo CIM de la red eléctrica representado en un grafo RDF (2.2).

La red representada en la Figura 4.2 está compuesta por los siguientes elementos conectados: barras, switches, generadores, cargas, líneas, capacitores y transformadores. Estos se pueden identificar en la Tabla 4.1.

	Componentes
	Barra
	Seccionador
	Interruptor
	Generador

	Generador Slack
	Línea
	Transformador
	Capacitor
	Carga
	Transformador

Tabla 4.1: Componentes que se pueden utilizar en el Editor.

4.2. Casos de estudio

A partir de la red esquematizada, se realizaron pruebas vinculadas al cambio de estado de los interruptores. Para realizar dichas pruebas sobre el editor los pasos necesarios son:

1. Primero se debe Abrir con un Editor el esquema unifilar de la red eléctrica.
2. Luego se cambia de estado los interruptores deseados para un caso de estudio.
3. A partir del paso anterior se guardan dichos cambios para que éstos queden persistidos en el XML correspondiente.
4. Después desde la aplicación desarrollada se construye el Grafo que representa la red eléctrica (3.2).
5. A continuación se crea un grafo reducido a partir del grafo del paso anterior, eliminando interruptores cerrados dentro del circuito y creando barras ficticias (3.2.3).
6. Teniendo el grafo reducido se aplican los diferentes algoritmos de búsqueda (2.5.1).
7. Finalmente, se realiza la conversión de los resultados obtenidos (3.4) en el paso anterior para poder plasmarlos en el Editor. Esto implica que las barras ficticias que pertenecen al grafo reducido son transformadas en barras reales, agregando los interruptores eliminados en el paso 5 para volver al grafo original.

Como salida se generan la conformación de las zonas desenergizadas (ids de componentes), los lazos y paralelos, en un formato simple de procesar y

comprensible para el editor, donde el resultado se pueda visualizar de manera gráfica.

Además se estudiaron los tiempos asociados al proceso de 7 pasos mencionado, que va desde Editor - Grafo - Grafo Reducido - Algoritmo de búsqueda - Grafo - Editor.

Concretamente, los tiempos medibles están asociados a:

- Construcción del grafo asociado al circuito de energía eléctrica.
- Construcción del grafo reducido.
- Ejecución de los algoritmos de búsqueda para recorrer el grafo reducido con el fin de obtener zonas desenergizadas, lazos y paralelos.
- Conversión de los resultados al grafo original.

4.2.1. Caso 1: apertura de los interruptores 22 y 23

En este caso se procedió a la apertura de los interruptores 22 y 23 sobre el entorno del editor, como se puede observar en la Figura 4.2.1.1.

Primero se analizaron las zonas desenergizadas luego de abrir los interruptores 22 y 23. Posteriormente se realizó la búsqueda de lazos y paralelos como consecuencia de la apertura de dichos interruptores.

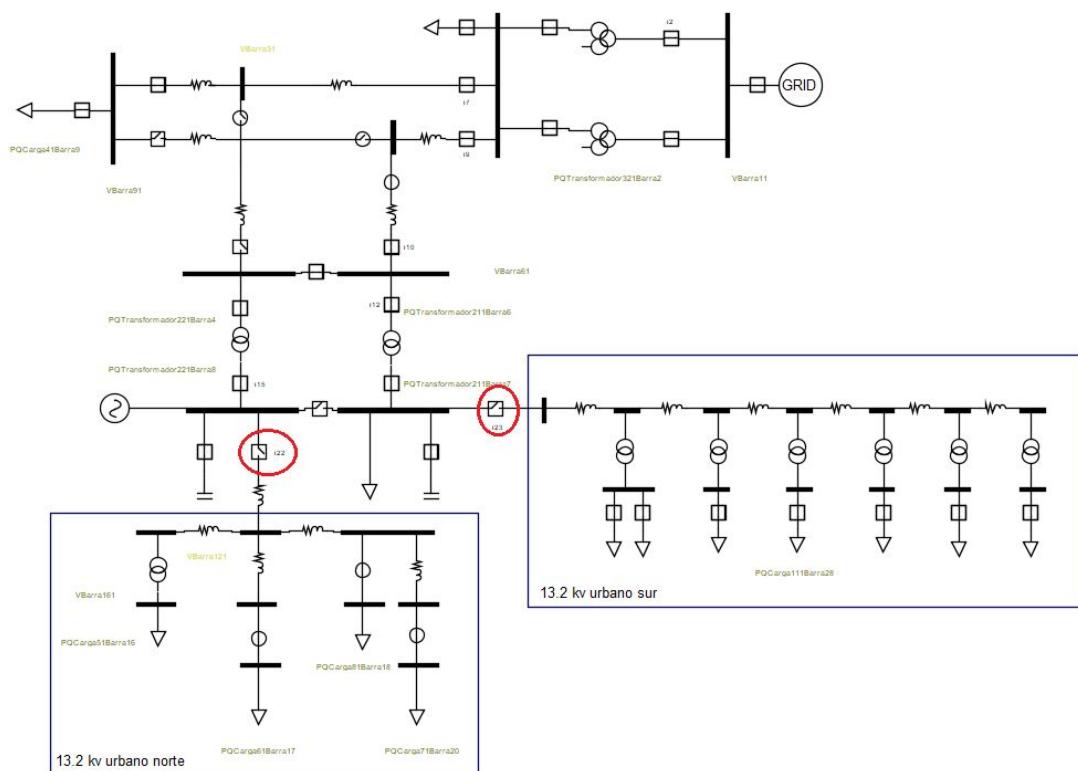


Figura 4.2.1.1: Apertura de los interruptores 22 y 23 en un sector de la red de Armstrong.

Teniendo en cuenta la red con los cambios de estado realizados se construye un grafo asociado (3.2.2).

Luego de aplicado el cambio el grafo queda representado sobre el siguiente formato:

Nodos:

- BarraX: Interruptor1, GeneradorSlack1
- BarraY:
 - Interruptor17, Capacitor1
 - Carga2
- BarraZ

Arcos:

- BarraX, BarraY: Seccionador7
- BarraX, BarraZ:
 - Interruptor4, Transformador31, Interruptor2
 - Interruptor5, Transformador32, Interruptor3

Cada nodo hace referencia a una barra y puede haber componentes conectados a ella. La salida anterior significa que la BarraX tiene el camino de componentes conectados Interruptor1, GeneradorSlack1; la BarraY tiene dos caminos de componentes conectados por un lado Interruptor17, Capacitor1 y por el otro la Carga2; y por último la BarraZ no tiene ningún camino asociado.

A su vez los arcos quedan representados por dos barras y los caminos de componentes que las conectan. Para el formato ejemplo la BarraX y la BarraY están conectadas por medio del Seccionador7. En cambio la BarraX y la BarraZ están conectadas por dos caminos de componentes, uno está formado por el Interruptor4, el Transformador31 y el Interruptor2 y el otro por el Interruptor5, el Transformador32 y el Interruptor3. En la Figura 4.2.1.2 se ve representado gráficamente el circuito utilizado para dicho formato.

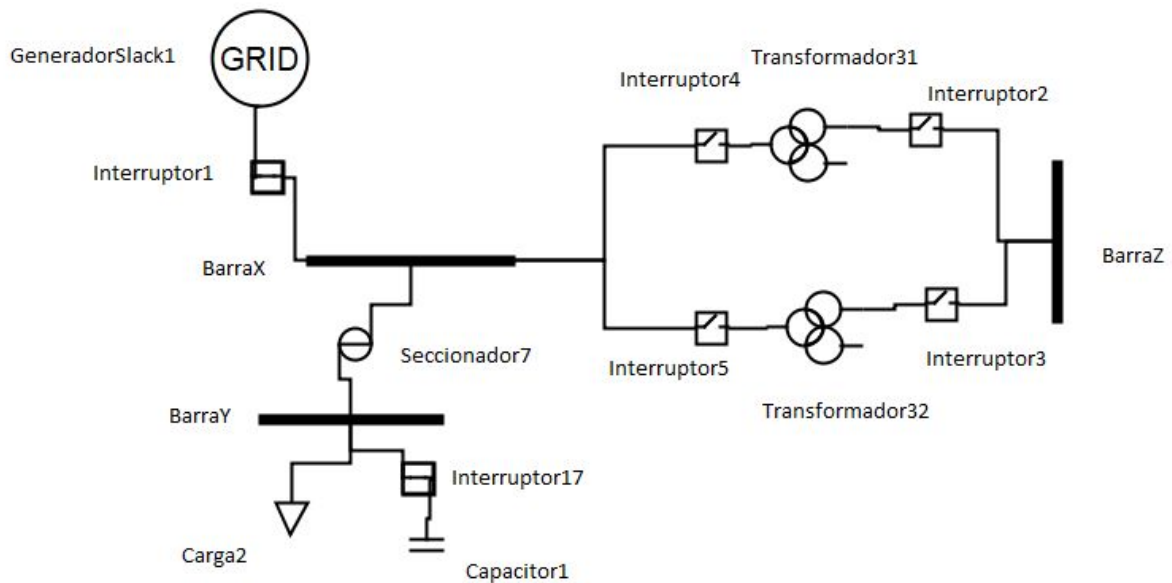


Figura 4.2.1.2: Circuito ejemplo.

Teniendo en cuenta el formato anterior, luego de haber realizado la apertura de los interruptores 22 y 23 se obtuvo el siguiente grafo:

Nodos:

- Barra1: Interruptor1, GeneradorSlack1
- Barra2: Interruptor6, Carga1
- Barra3
- Barra4
- Barra6
- Barra5
- Barra7:
 - Interruptor17, Capacitor1
 - Carga2
- Barra8:
 - Interruptor18, Capacitor2
 - Generador2
- Barra9: Interruptor21, Carga4
- Barra11
- Barra12
- Barra13
- Barra14
- Barra16: Carga5
- Barra17: Carga6
- Barra18: Carga8
- Barra19

- Barra20: Carga7
- Barra21
- Barra22
- Barra23
- Barra24
- Barra25
- Barra26
- Barra27
- Barra28: Interruptor26, Carga11
- Barra29: Interruptor25, Carga10
- Barra30:
 - Interruptor24, Carga9
 - Interruptor30, Carga15
- Barra31: Interruptor27, Carga12
- Barra32: Interruptor28, Carga13
- Barra33: Interruptor29, Carga14

Arcos:

- Barra20, Barra19: Seccionador7
- Barra33, Barra27: Transformador29
- Barra9, Barra3: Interruptor19, Linea5
- Barra9, Barra5: Interruptor20, Linea6, Seccionador3
- Barra31, Barra25: Transformador27
- Barra13, Barra11: Linea8
- Barra13, Barra17: Seccionador4
- Barra2, Barra5: Interruptor9, Linea3
- Barra2, Barra1:
 - Interruptor4, Transformador31, Interruptor2
 - Interruptor5, Transformador32, Interruptor3
- Barra2, Barra3: Interruptor7, Linea1
- Barra4, Barra6: Interruptor11
- Barra4, Barra3: Interruptor8, Linea2, Seccionador1
- Barra4, Barra8: Interruptor13, Transformador22, Interruptor15
- Barra30, Barra22: Transformador24
- Barra29, Barra23: Transformador25
- Barra32, Barra26: Transformador28
- Barra14, Barra19: Linea11
- Barra14, Barra18: Seccionador5
- Barra14, Barra11: Linea10
- Barra27, Barra26: Linea17
- Barra21, Barra7: Interruptor23
- Barra21, Barra22: Linea12

- Barra12, Barra16: Transformador23
- Barra12, Barra11: Linea9
- Barra26, Barra25: Linea16
- Barra24, Barra28: Transformador26
- Barra24, Barra23: Linea14
- Barra24, Barra25: Linea15
- Barra8, Barra7: Interruptor16
- Barra8, Barra11: Interruptor22, Linea7
- Barra6, Barra7: Interruptor12, Transformador21, Interruptor14
- Barra6, Barra5: Interruptor10, Linea4, Seccionador2
- Barra23, Barra22: Linea13

A partir del grafo anterior el paso siguiente es crear un grafo reducido (3.2.3) sobre el cual se van aplicar los distintos algoritmos de recorrido (3.3) para calcular zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3). Cada vez que se realizan cambios de estados de los interruptores hay que reconstruir el grafo reducido ya que el cambio de estado de un interruptor implica que aparezca un nuevo arco entre dos nodos o desaparezca dicha relación. Cabe recordar que la idea del grafo reducido sirve para disminuir los costos de búsqueda debido a que su construcción implica, por un lado, fusionar barras que estén conectadas por medio de switches cerrados y por el otro eliminar switches cerrados y abiertos.

El grafo reducido queda representado de la siguiente forma:

Nodos:

- Barra1_Fic: GeneradorSlack1
- Barra2_Fic: Carga1
- Barra3_Fic
- Barra4_Fic
- Barra5_Fic
- Barra7_Fic:
 - Capacitor1
 - Carga2
- Barra8_Fic:
 - Capacitor2
 - Generador2
- Barra9_Fic: Carga4
- Barra11_Fic
- Barra12_Fic
- Barra13_Fic: Carga6
- Barra14_Fic: Carga8
- Barra15_Fic
- Barra16_Fic: Carga5

- Barra20_Fic: Carga7
- Barra21_Fic
- Barra22_Fic
- Barra23_Fic
- Barra24_Fic
- Barra25_Fic
- Barra26_Fic
- Barra27_Fic
- Barra28_Fic: Carga11
- Barra29_Fic: Carga10
- Barra30_Fic:
 - Carga9
 - Carga15
- Barra31_Fic: Carga12
- Barra32_Fic: Carga13
- Barra33_Fic: Carga14

Arcos:

- Barra33_Fic, Barra27_Fic: Transformador29
- Barra9_Fic, Barra3_Fic: Linea5
- Barra31_Fic, Barra25_Fic: Transformador27
- Barra13_Fic, Barra11_Fic: Linea8
- Barra2_Fic, Barra5_Fic: Linea3
- Barra2_Fic, Barra1_Fic:
 - Transformador31
 - Transformador32
- Barra2_Fic, Barra3_Fic: Linea1
- Barra4_Fic, Barra8_Fic: Transformador22
- Barra30_Fic, Barra22_Fic: Transformador24
- Barra29_Fic, Barra23_Fic: Transformador25
- Barra32_Fic, Barra26_Fic: Transformador28
- Barra14_Fic, Barra20_Fic: Linea11
- Barra14_Fic, Barra11_Fic: Linea10
- Barra27_Fic, Barra26_Fic: Linea17
- Barra21_Fic, Barra22_Fic: Linea12
- Barra12_Fic, Barra16_Fic: Transformador23
- Barra12_Fic, Barra11_Fic: Linea9
- Barra26_Fic, Barra25_Fic: Linea16
- Barra24_Fic, Barra28_Fic: Transformador26
- Barra24_Fic, Barra23_Fic: Linea14
- Barra24_Fic, Barra25_Fic: Linea15
- Barra4_Fic, Barra7_Fic: Transformador21

- Barra4_Fic, Barra5_Fic: Linea4
- Barra23_Fic, Barra22_Fic: Linea13

4.2.1.1. Zonas desenergizadas

Luego de ejecutar el algoritmo de búsqueda (3.3.1) sobre el grafo reducido se obtuvo como resultado 23 barras desenergizadas: Barra11, Barra12, Barra13, Barra14, Barra15, Barra16, Barra17, Barra18, Barra19, Barra20, Barra21, Barra22, Barra23, Barra24, Barra25, Barra26, Barra27, Barra28, Barra29, Barra30, Barra31, Barra32, Barra33. Las mismas corresponde a las zonas Urbano Norte y Urbano Sur (Figura 4.2.1.1.1).

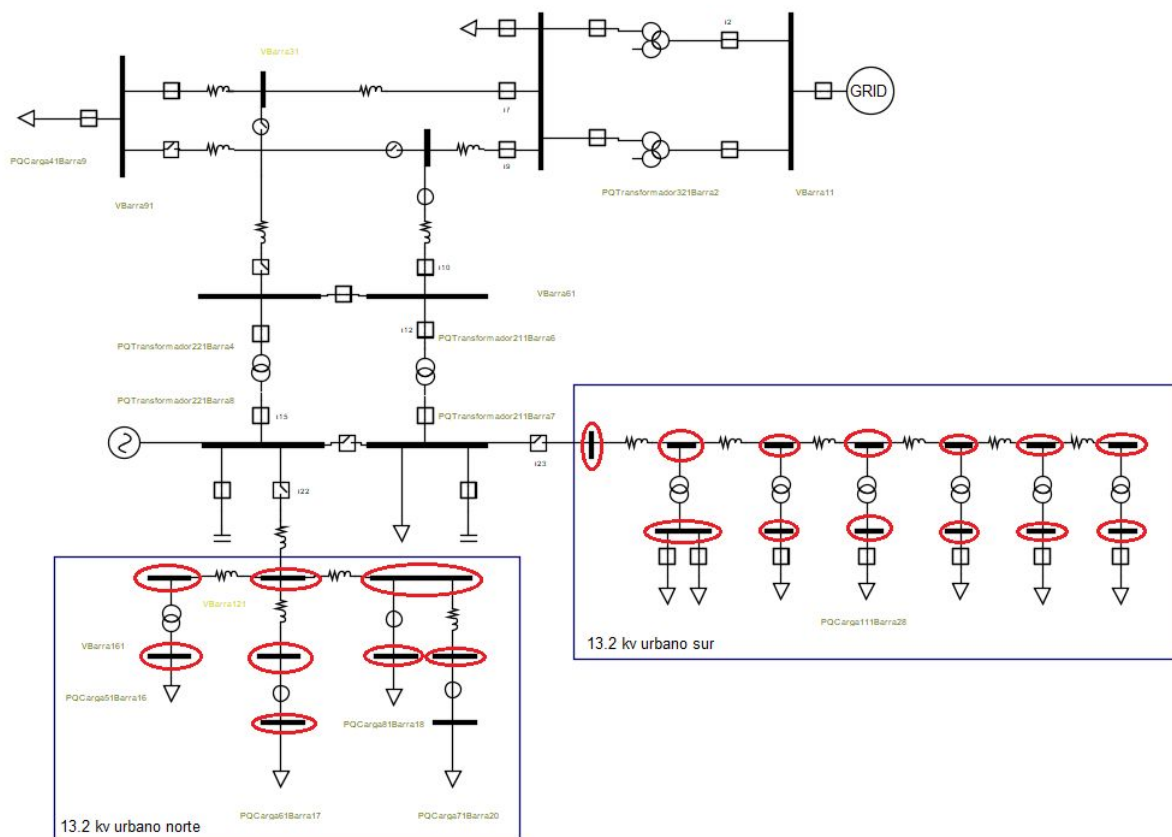


Figura 4.2.1.1.1: Zonas desenergizadas con apertura de los interruptores 22 y 23.

Como se puede ver en la figura 4.2.1.1.1 el resultado obtenido es correcto ya que los interruptores 22 y 23 son los únicos componentes que vinculan las barras desenergizadas con las fuentes de la red eléctrica. En este caso las fuentes son el GeneradorSlack1 y Generador1.

Considerando que para todas las pruebas se utilizó una PC con las siguientes especificaciones técnicas: procesador Intel Core i7-4710HQ CPU 2.50GHz ; 8GB de memoria RAM; SSD 250GB, el tiempo total de ejecución asociado al proceso de detección desde el grafo inicial es 1520.78 ms en promedio. El promedio se obtuvo

a partir de diez ejecuciones tomando la misma red eléctrica como entrada. Este tiempo es resultado de la suma de los tiempos de los siguientes pasos:

- Construcción del grafo a partir del modelo CIM del circuito (3.2.2): 1032.40 ms.
- Construcción del grafo reducido a partir del grafo anterior (3.2.3): 486.05 ms.
- Ejecución del algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener zonas desenergizadas (3.3.1): 1.13 ms.
- Transformación de los resultados del grafo reducido al grafo real (3.4): 1.96 ms.

Sobre la prueba anterior se puede destacar que construir el grafo a partir del modelo es más costoso que la construcción del grafo reducido debido a que para el primero se debe recorrer toda la red por completo para crear todos los nodos y las conexiones entre éstos.

Además, se puede observar que los tiempos del algoritmo de búsqueda y de la transformación los de resultados son poco significativos respecto del total, a diferencia de las construcción de los grafos que abarcar aproximadamente el 99%.

4.2.1.2. Lazos

Luego de analizar las zonas desenergizadas se procedió a buscar los lazos (2.7.1.2) dentro de la red eléctrica (Figura 4.2.1.1). Estos se dan cuando una barra recibe energía de la misma fuente pero por distintos caminos.

Para la detección de lazos se aplica el algoritmo de búsqueda (3.3.2). Como resultado se genera una salida con el siguiente formato:

Lazos:

Barra 1: BarraX, Barra 2: BarraY

Recorrido 0:

(BarraX, BarraE) : {[Interruptor4, Transformador31, Interruptor2]}

(BarraE, BarraF) : {[Interruptor7, Linea1]}

(BarraF, Barra Y) : {[Interruptor19, Linea5]}

Recorrido 1:

(Barra X, BarraE) : {[Interruptor5, Transformador32, Interruptor3]}

(BarraE, BarraF) : {[Interruptor7, Linea1]}

(BarraF, BarraY) : {[Interruptor19, Linea5]}

En este caso cada lazo está representado por dos barras, la BarraX que tiene asociado un generador y la BarraY a la cual le llega energía de la BarraX por diferentes caminos. Cada recorrido representa el camino por el cual la BarraX le da energía a la BarraY.

Concretamente sobre el caso de estudio aplicando la detección de lazos se obtuvo:

Lazos:

Barra 1:Barra1, Barra 2:Barra9

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Barra 1:Barra1, Barra 2:Barra8

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

(Barra6, Barra4) : {[Interruptor11]}

(Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

(Barra6, Barra4) : {[Interruptor11]}

(Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}

Barra 1:Barra1, Barra 2:Barra3

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

Barra 1:Barra1, Barra 2:Barra2

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

Barra 1:Barra1, Barra 2:Barra5

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

Barra 1:Barra1, Barra 2:Barra4

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]]]
(Barra2, Barra5) : {[Interruptor9, Linea3]]]
(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]]]
(Barra6, Barra4) : {[Interruptor11]]]

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]]]
(Barra2, Barra5) : {[Interruptor9, Linea3]]]
(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]]]
(Barra6, Barra4) : {[Interruptor11]]]

Barra 1:Barra1, Barra 2:Barra7

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]]]
(Barra2, Barra5) : {[Interruptor9, Linea3]]]
(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]]]
(Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]]]

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]]]
(Barra2, Barra5) : {[Interruptor9, Linea3]]]
(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]]]
(Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]]]

Barra 1:Barra8, Barra 2:Barra1

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]]
(Barra4, Barra6) : {[Interruptor11]]]
(Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]]
(Barra5, Barra2) : {[Interruptor9, Linea3]]]
(Barra2, Barra1) : {[Interruptor4, Transformador31, Interruptor2]]]

Recorrido 1:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]]
(Barra4, Barra6) : {[Interruptor11]]]
(Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]]
(Barra5, Barra2) : {[Interruptor9, Linea3]]]
(Barra2, Barra1) : {[Interruptor5, Transformador32, Interruptor3]]]

Por un lado para la Barra1 que tiene asociado el Generador Slack1 le distribuye energía a la Barra9 por los siguientes caminos:

- Primer recorrido 1
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]]]
 - (Barra2, Barra3) : {[Interruptor7, Linea1]]]

- (Barra3, Barra9) : {[Interruptor19, Linea5]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
 - (Barra3, Barra9) : {[Interruptor19, Linea5]}

Esto se puede ver gráficamente representado en la Figuras 4.2.1.2.1 y 4.2.1.2.2.

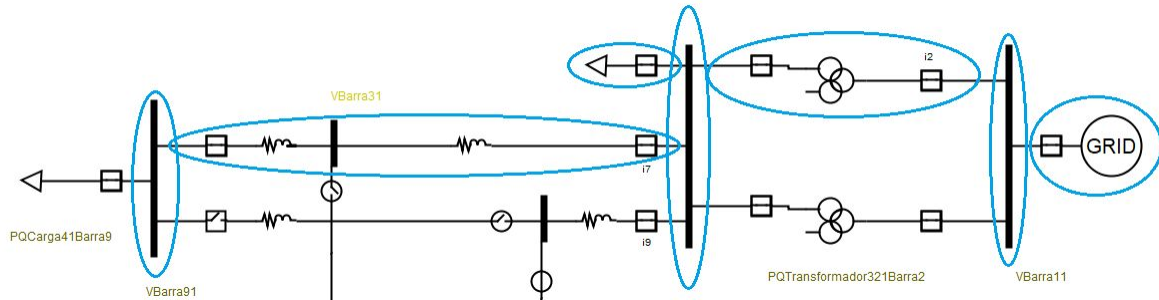


Figura 4.2.1.2.1: Primer recorrido entre Barra1 y Barra9.

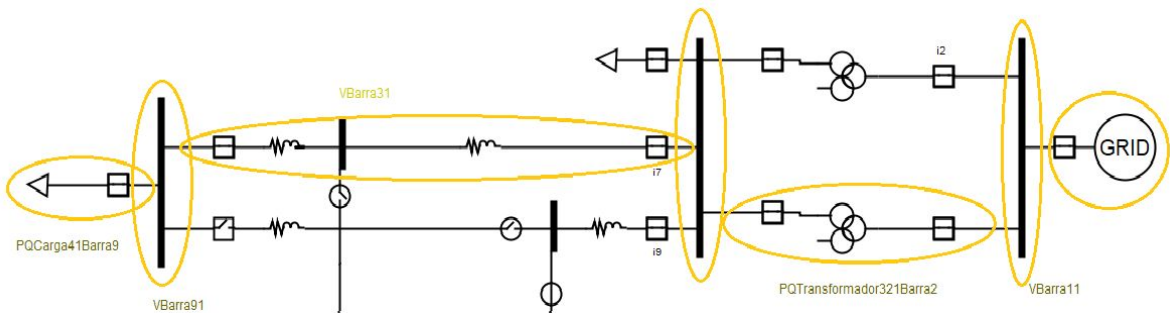


Figura 4.2.1.2.2: Segundo recorrido entre Barra1 y Barra9.

Por otro lado la Barra1 también le distribuye energía a la Barra8 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra6, Barra4) : {[Interruptor11]}
 - (Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

- (Barra6, Barra4) : {[Interrupor11]}
- (Barra4, Barra8) : {[Interrupor13, Transformador22, Interrupor15]}

En las Figuras 4.2.1.2.3 y 4.2.1.2.4 se puede ver esquematizado el lazo anterior.

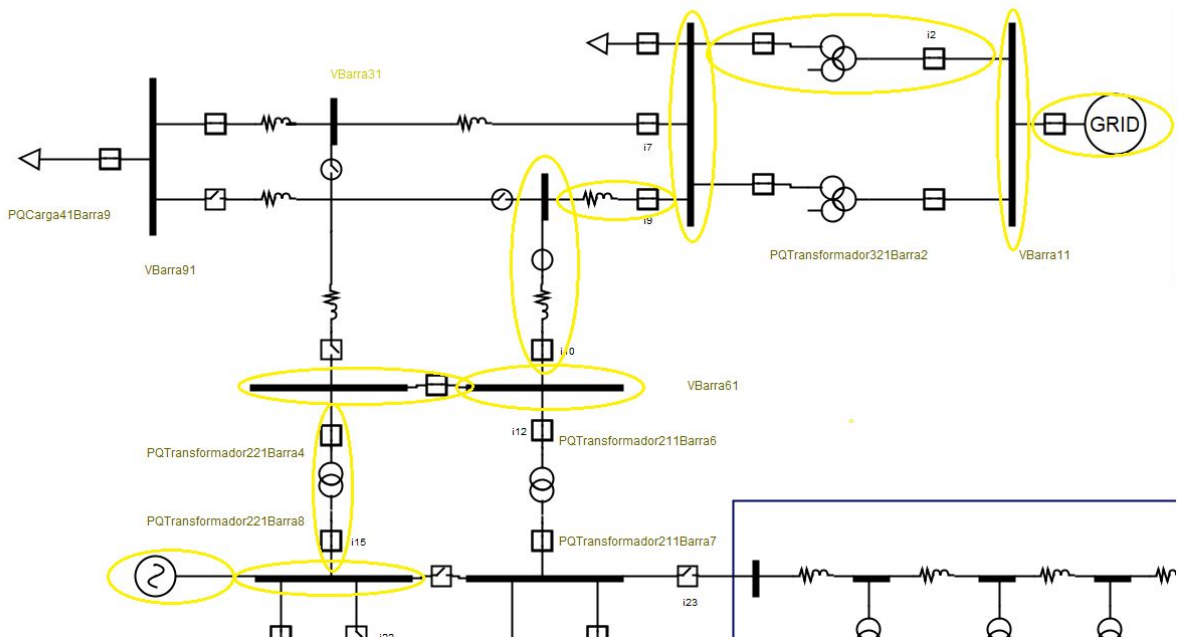


Figura 4.2.1.2.3: Primer recorrido entre Barra1 y Barra8.

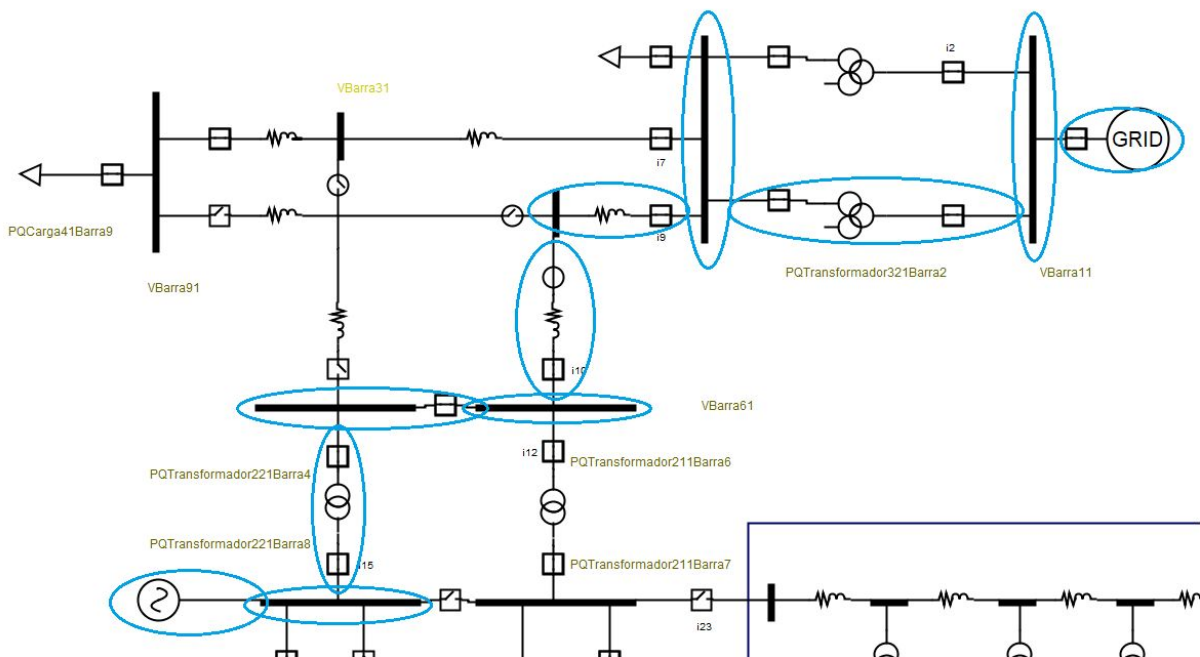


Figura 4.2.1.2.4: Segundo recorrido entre Barra1 y Barra8.

A su vez la Barra1 le distribuye energía a la Barra3 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}

El lazo obtenido anteriormente se puede ver representado en las Figuras 4.2.1.2.5 y 4.2.1.2.6.

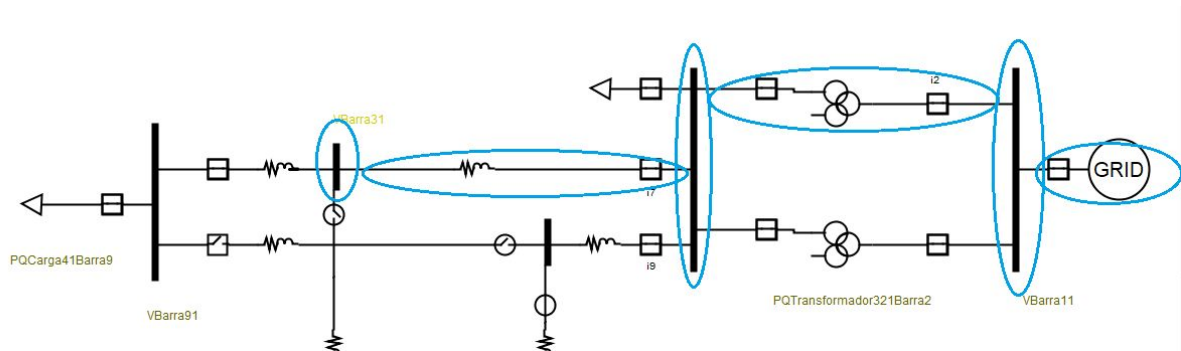


Figura 4.2.1.2.5: Primer recorrido entre Barra1 y Barra3.

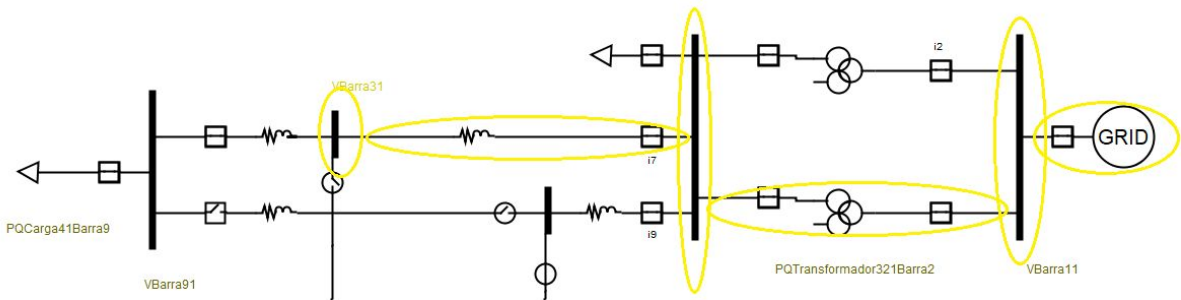


Figura 4.2.1.2.6: Segundo recorrido entre Barra1 y Barra3.

La Barra1 también le distribuye energía a la Barra2 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

Este caso se puede observar en las Figura 4.2.1.2.7 y 4.2.1.2.8.

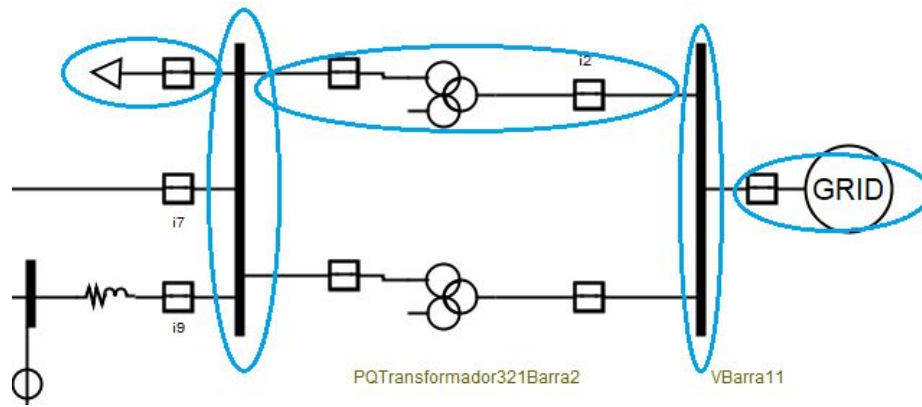


Figura 4.2.1.2.7: Primer recorrido entre Barra1 y Barra2.

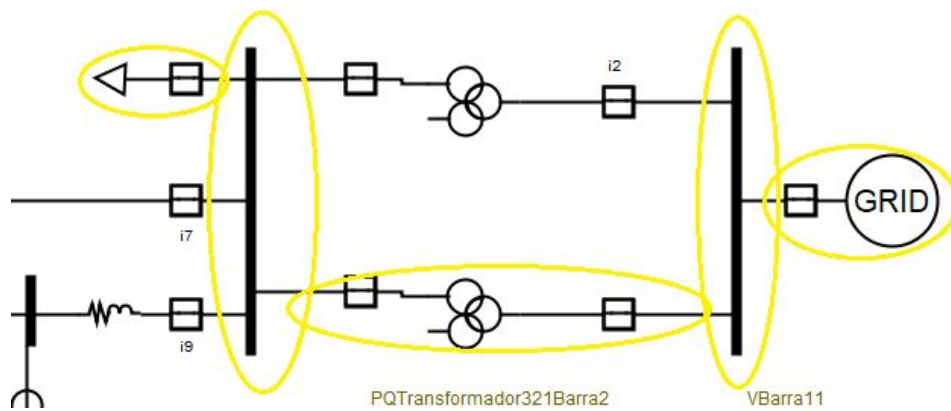


Figura 4.2.1.2.8: Segundo recorrido entre Barra1 y Barra2.

Además, la Barra1 le distribuye energía a la Barra5 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}

El lazo obtenido se puede ver en las Figura 4.2.1.2.9 y 4.2.1.2.10.

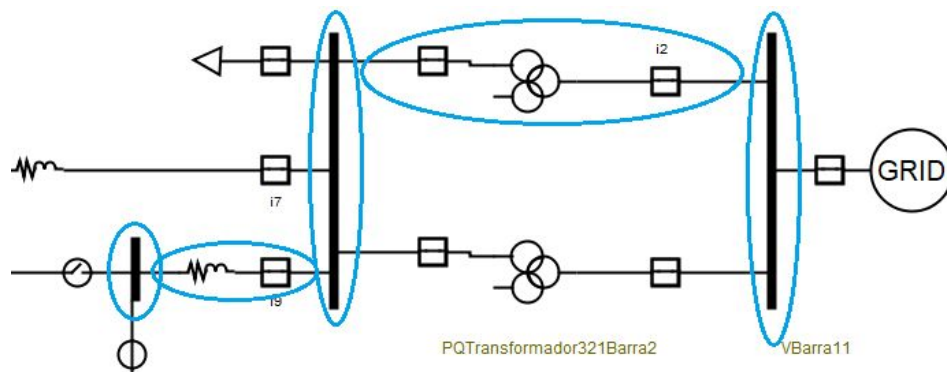


Figura 4.2.1.2.9: Primer recorrido entre Barra1 y Barra5.

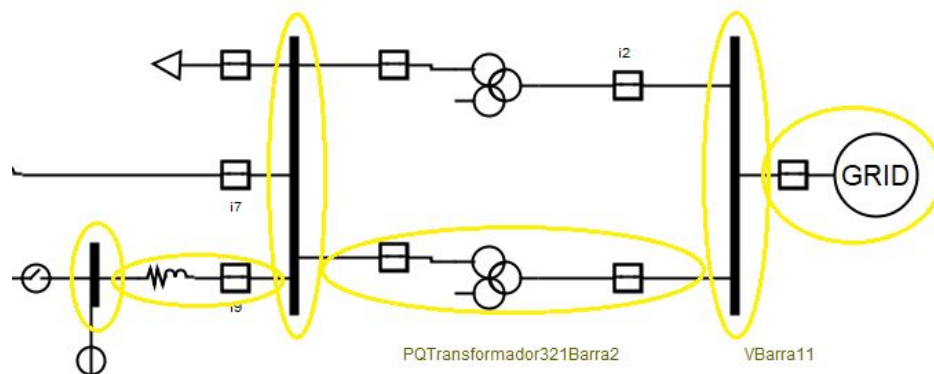


Figura 4.2.1.2.10: Segundo recorrido entre Barra1 y Barra5.

También se obtiene que la Barra1 le distribuye energía a la Barra4 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra6, Barra4) : {[Interruptor11]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra6, Barra4) : {[Interruptor11]}

Las Figuras 4.2.1.2.11 y 4.2.1.2.12 muestran el lazo que se genera entre la Barra1 y la Barra4.

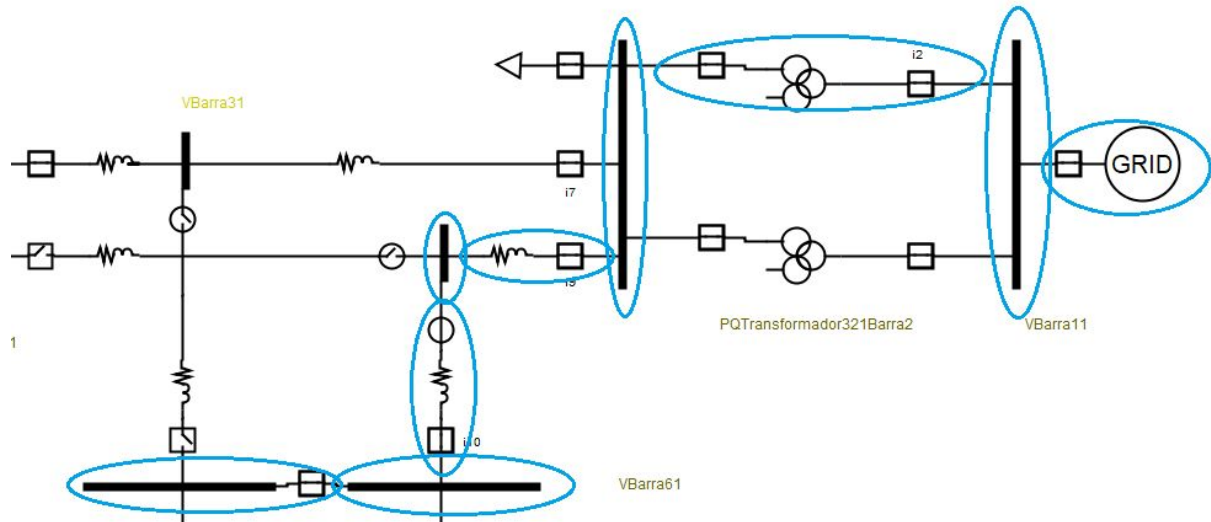


Figura 4.2.1.2.11: Primer recorrido entre Barra1 y Barra4.

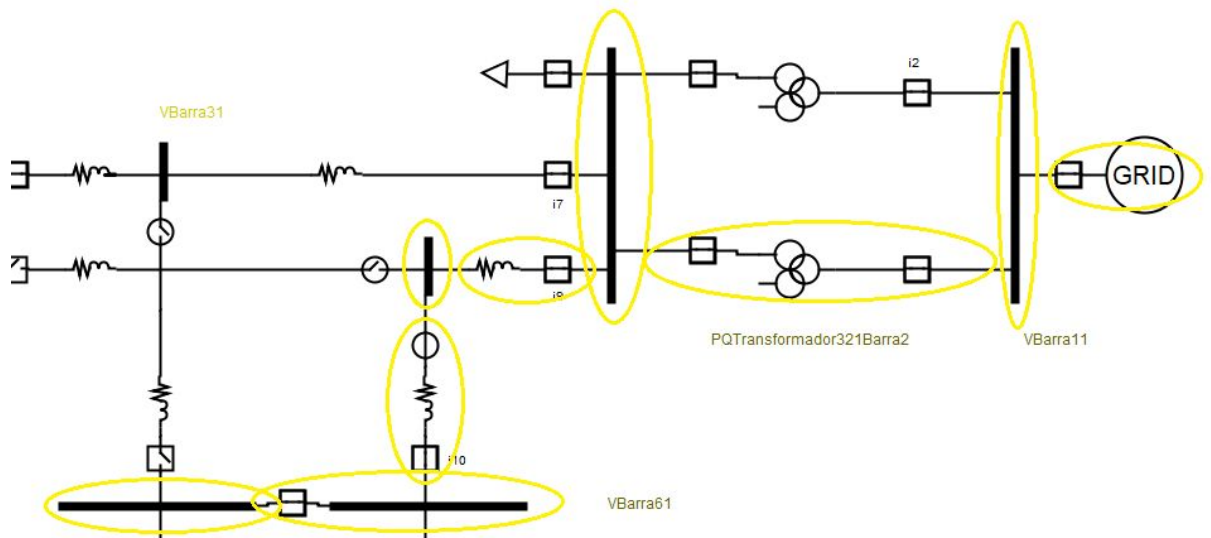


Figura 4.2.1.2.12: Segundo recorrido entre Barra1 y Barra4.

A su vez, la Barra1 le distribuye energía a la Barra7 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]}
- Segundo recorrido:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

- (Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]}

Este lazo se puede ver representado en la Figura 4.2.1.2.13 y en la Figura 4.2.1.2.14.

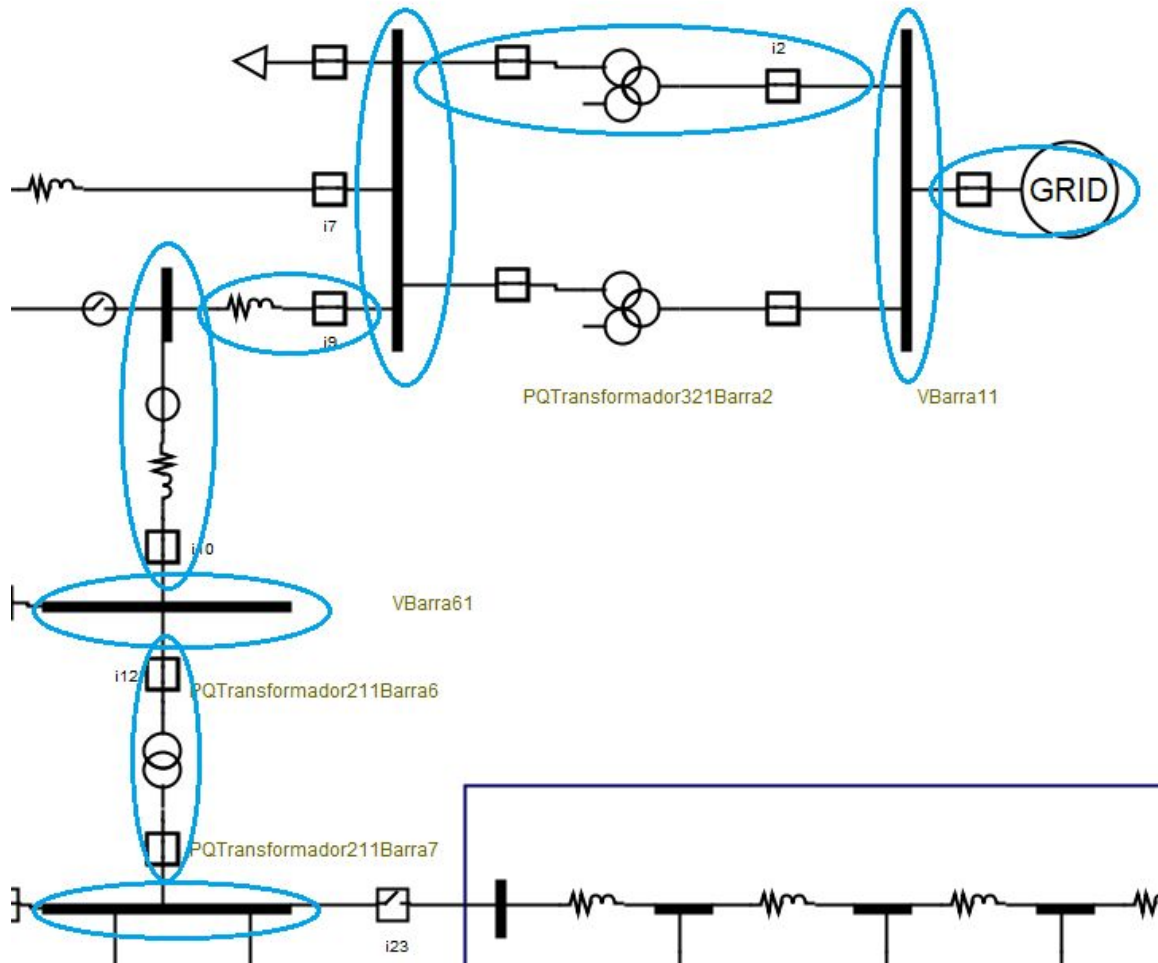


Figura 4.2.1.2.13: Primer recorrido entre Barra1 y Barra7.

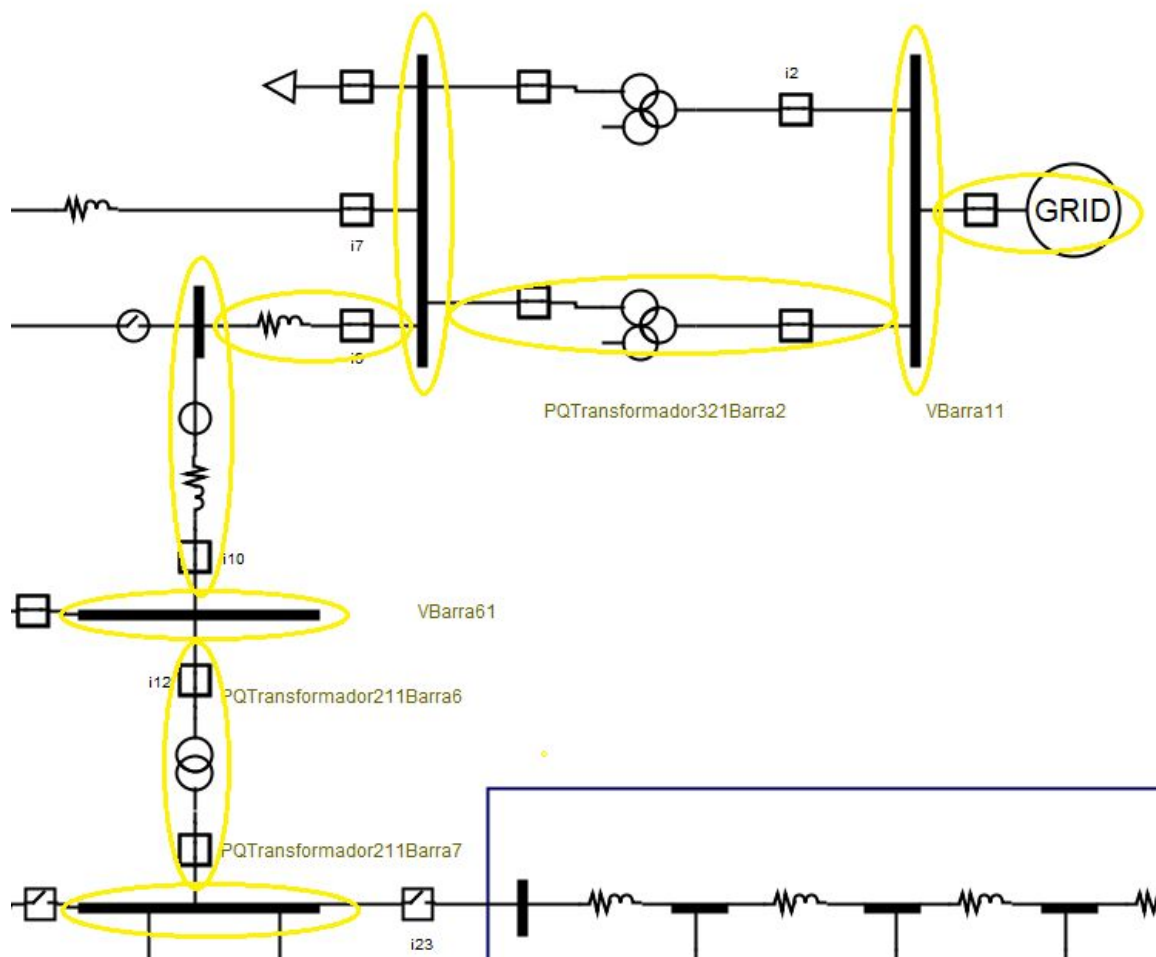


Figura 4.2.1.2.14: Segundo recorrido entre Barra1 y Barra7.

Mediante la Barra8, que tiene asociado el Generador2, se distribuye energía a la Barra1 por los siguientes caminos:

- Primer recorrido 1:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
 - (Barra2, Barra1) : {[Interruptor4, Transformador31, Interruptor2]}
- Segundo recorrido:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
 - (Barra2, Barra1) : {[Interruptor5, Transformador32, Interruptor3]}

En las Figuras 4.2.1.2.15 y 4.2.1.2.16 se puede ver representado el lazo obtenido anteriormente.

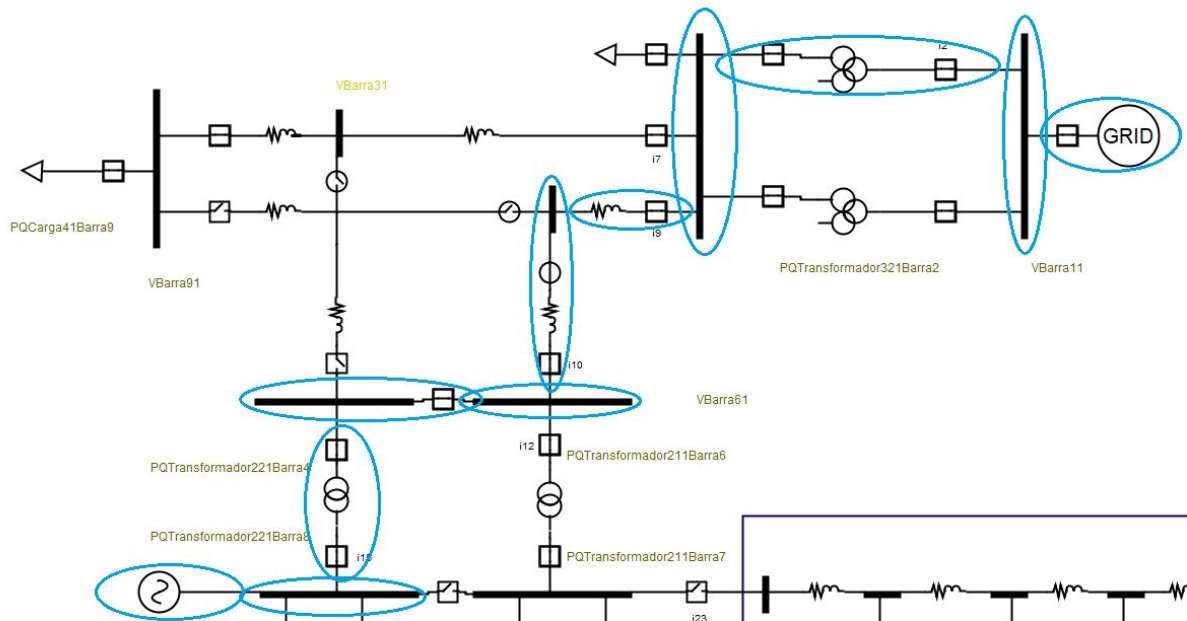


Figura 4.2.1.2.15: Primer recorrido entre Barra8 y Barra1.

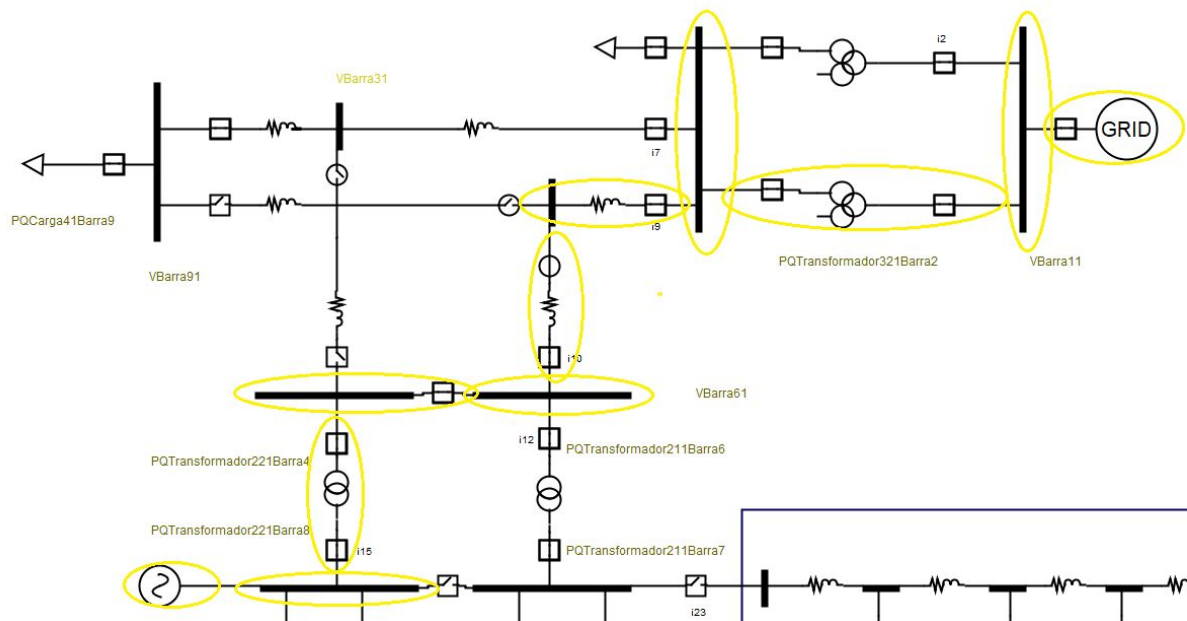


Figura 4.2.1.2.16: Segundo recorrido entre Barra8 y Barra1.

Luego de analizar los resultados se puede determinar que la apertura de los interruptores 22 y 23 no impiden el paso de la corriente de las fuentes al resto del circuito, lo cual permite afirmar que el resultado obtenido es correcto. Por lo tanto

con el GeneradorSlack1 y Generador1 se forman lazos y paralelos en la red eléctrica.

Los tiempos de construcción promedio del primer grafo (3.2.2) y del reducido (3.2.3) son los mismos que se vieron para el estudio de zonas desenergizadas. Pero para el caso de los lazos cambian los tiempos promedio del algoritmo de búsqueda (3.3) y de la conversión de los resultados (3.4). Estos tiempos en promedio fueron de 1.97 ms y 2.99 ms respectivamente (luego de realizar diez ejecuciones de la aplicación).

Finalmente, resulta más costoso la transformación de los resultados del grafo reducido al grafo real. De todos modos no existe gran diferencia entre dichos tiempos de ejecución, teniendo en cuenta que para el primer caso se recorre el grafo reducido y para el segundo se recorre una hash que almacena una lista de elementos y el grafo original.

4.2.1.3. Paralelos

Luego de buscar lazos se buscan los paralelos (2.7.1.3) dentro del circuito (Figura 4.2.1.1). Estas circunstancias se dan cuando una barra recibe energía de dos fuentes de energía distintas.

Para la detección de paralelos se aplica el algoritmo de búsqueda (3.3.3). Como resultado se genera una salida con el siguiente formato:

Paralelos:

Barra 1:BarraX, Barra 2:BarraY

Recorrido 0:

(BarraX, BarraQ) : {[Interruptor4, Transformador31, Interruptor2]}

(BarraQ, BarraW) : {[Interruptor7, Linea1]}

(BarraW, BarraY) : {[Interruptor19, Linea5]}

Recorrido 1:

(BarraX, BarraQ) : {[Interruptor5, Transformador32, Interruptor3]}

(BarraQ, BarraW) : {[Interruptor7, Linea1]}

(BarraW, BarraY) : {[Interruptor19, Linea5]}

Barra 1:BarraZ, Barra 2:BarraY

Recorrido 0:

(BarraZ, BarraK) : {[Interruptor13, Transformador22, Interruptor15]}

(BarraK, BarraM) : {[Interruptor11]}

(BarraM, BarraN) : {[Interruptor10, Linea4, Seccionador2]}

(BarraN, BarraP) : {[Interruptor9, Linea3]}

(BarraP, BarraW) : {[Interruptor7, Linea1]}

(BarraW, BarraY) : {[Interruptor19, Linea5]}

Cada paralelo está representado por tres barras, la BarraX que tiene asociado un generador, la BarraZ que tiene otro generador y la BarraY, a la cual le llega energía de la BarraX y la BarraZ por los distintos caminos. Cada recorrido representa el camino por el cuál la BarraX y la BarraZ le brindan energía a la BarraY .

Concretamente sobre el caso de estudio aplicando la detección de lazos se obtuvo:

Paralelos:

Barra 1:Barra1, Barra 2:Barra9

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Barra 1:Barra1, Barra 2:Barra8

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

(Barra6, Barra4) : {[Interruptor11]}

(Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

(Barra6, Barra4) : {[Interruptor11]}

(Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}

Barra 1:Barra8, Barra 2:Barra8

Recorrido 0:

Barra 1:Barra1, Barra 2:Barra1

Recorrido 0:

Barra 1:Barra8, Barra 2:Barra7

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}

(Barra4, Barra6) : {[Interruptor11]}

(Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]}

Barra 1:Barra1, Barra 2:Barra3

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

Barra 1:Barra8, Barra 2:Barra9

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}

(Barra4, Barra6) : {[Interruptor11]}

(Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}

(Barra5, Barra2) : {[Interruptor9, Linea3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Barra 1:Barra1, Barra 2:Barra2

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

Barra 1:Barra1, Barra 2:Barra5

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

Barra 1:Barra1, Barra 2:Barra7

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

(Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra5) : {[Interruptor9, Linea3]}

(Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}

(Barra6, Barra7) : {[Interruptor12, Transformador21, Interruptor14]}

Barra 1:Barra8, Barra 2:Barra2

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}

(Barra4, Barra6) : {[Interruptor11]}

(Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}

(Barra5, Barra2) : {[Interruptor9, Linea3]}

Barra 1:Barra8, Barra 2:Barra1

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]
 (Barra4, Barra6) : {[Interruptor11]]
 (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]
 (Barra5, Barra2) : {[Interruptor9, Linea3]]
 (Barra2, Barra1) : {[Interruptor4, Transformador31, Interruptor2]]

Recorrido 1:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]
 (Barra4, Barra6) : {[Interruptor11]]
 (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]
 (Barra5, Barra2) : {[Interruptor9, Linea3]]
 (Barra2, Barra1) : {[Interruptor5, Transformador32, Interruptor3]]

Barra 1:Barra8, Barra 2:Barra3

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]
 (Barra4, Barra6) : {[Interruptor11]]
 (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]
 (Barra5, Barra2) : {[Interruptor9, Linea3]]
 (Barra2, Barra3) : {[Interruptor7, Linea1]]

Barra 1:Barra8, Barra 2:Barra5

Recorrido 0:

(Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]
 (Barra4, Barra6) : {[Interruptor11]]
 (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]

La Barra9 recibe energía de la Barra1 y la Barra8. La Barra1 tiene asociado el Generador Slack1 y la Barra8 tiene asociado el Generador2. Los caminos por los cuales se forman los paralelos son:

- Primer paralelo:
 - Barra1 a Barra9:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]]
 - (Barra2, Barra3) : {[Interruptor7, Linea1]]
 - (Barra3, Barra9) : {[Interruptor19, Linea5]]
 - Barra8 a Barra9:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]]
 - (Barra4, Barra6) : {[Interruptor11]]
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]]
 - (Barra5, Barra2) : {[Interruptor9, Linea3]]
 - (Barra2, Barra3) : {[Interruptor7, Linea1]]
 - (Barra3, Barra9) : {[Interruptor19, Linea5]]

- Segundo paralelo:
 - Barra1 a Barra9:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
 - (Barra3, Barra9) : {[Interruptor19, Linea5]}
 - Barra8 a Barra9:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
 - (Barra3, Barra9) : {[Interruptor19, Linea5]}

Se puede ver representado los paralelos obtenidos anteriormente en las Figuras 4.2.1.3.1 y 4.2.1.3.2.

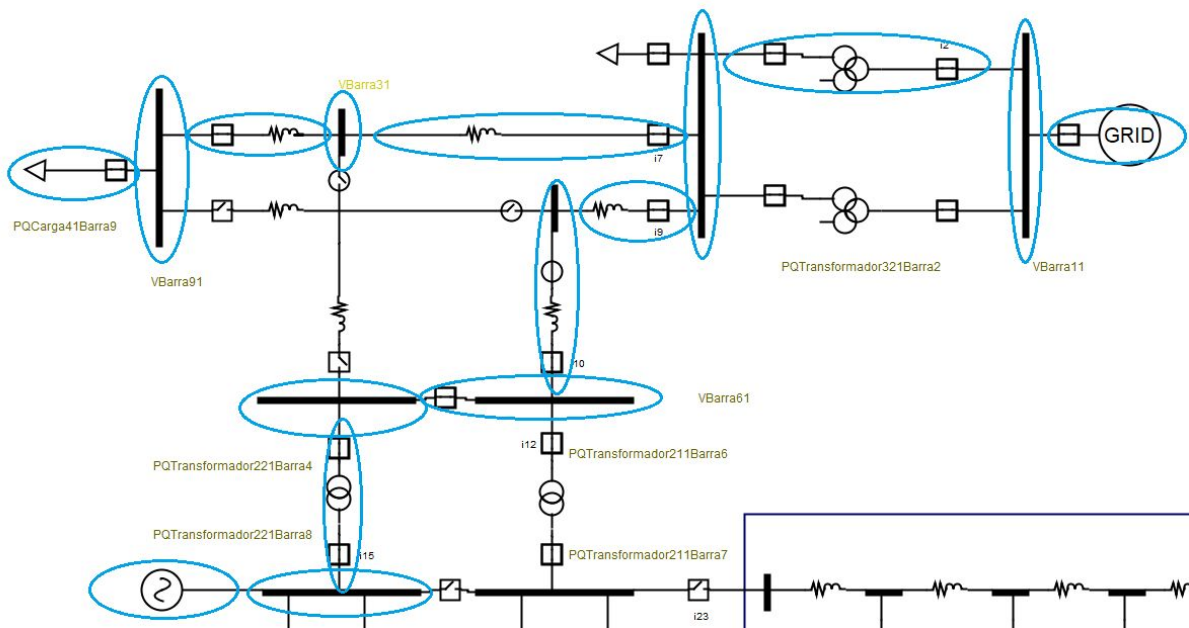


Figura 4.2.1.3.1: Primer paralelo, la Barra 9 recibe energía de la Barra1 y Barra8.

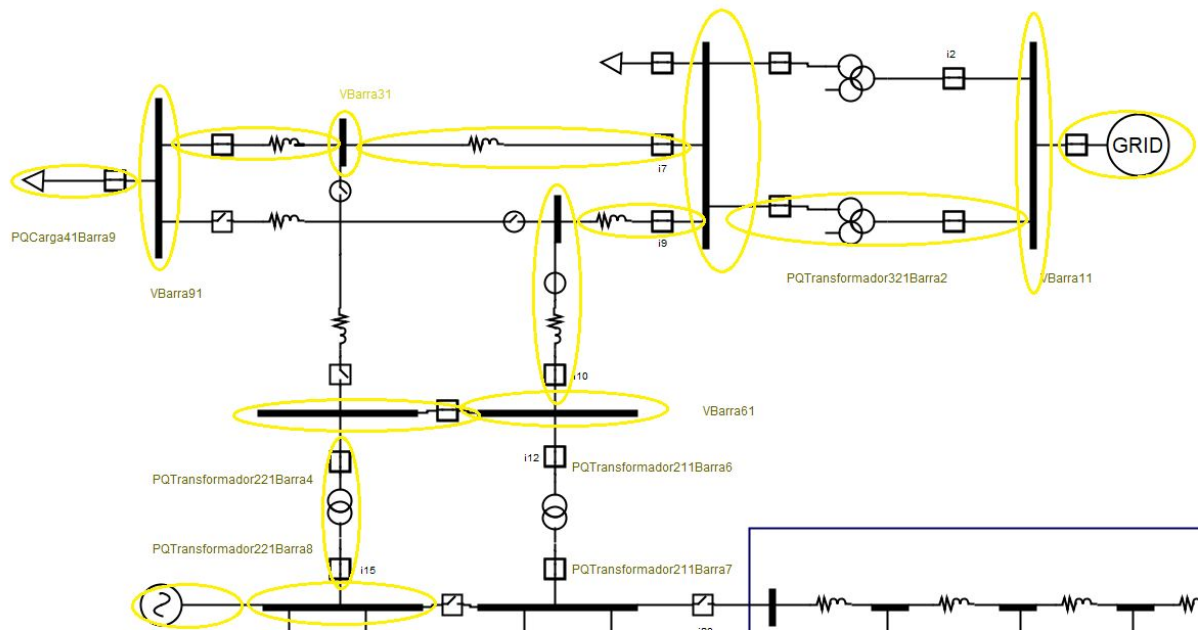


Figura 4.2.1.3.2: Segundo paralelo, la Barra9 recibe energía de la Barra1 y Barra8.

Por otro lado, la Barra8 recibe energía de la Barra1 y de ella misma. Como se mencionó anteriormente, la Barra1 tiene asociado el Generador Slack1 y la Barra8 tiene asociado el Generador2. Los caminos por los cuales se forman los paralelos son:

- Primer paralelo:
 - Barra1 a Barra8:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra6, Barra4) : {[Interruptor11]}
 - (Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}
 - Barra8 a Barra8
- Segundo paralelo:
 - Barra1 a Barra8:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - (Barra5, Barra6) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra6, Barra4) : {[Interruptor11]}
 - (Barra4, Barra8) : {[Interruptor13, Transformador22, Interruptor15]}

- Barra8 a Barra8:

Los paralelos obtenidos anteriormente se pueden observar en las Figuras 4.2.1.3.3 y 4.2.1.3.4.

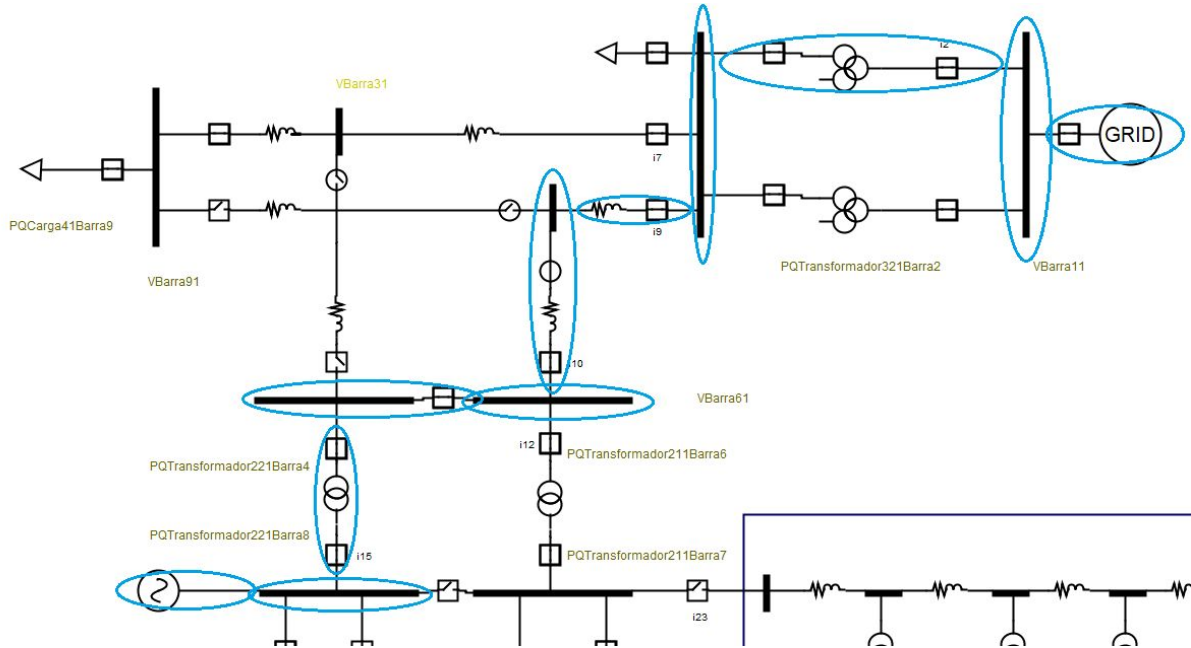


Figura 4.2.1.3.3: Primer paralelo, la Barra8 recibe energía de la Barra 1 y de si misma.

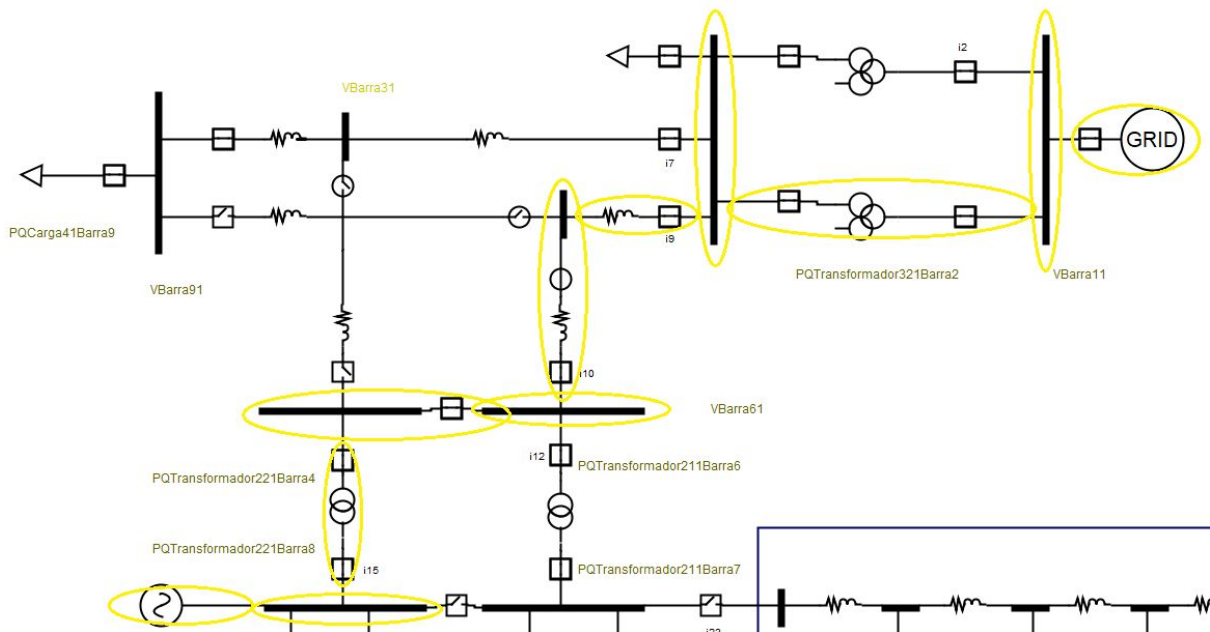


Figura 4.2.1.3.4: Segundo paralelo, la Barra8 recibe energía de la Barra1 y de sí misma.

Además, la Barra1 recibe energía de la Barra8 y de ella misma. Los caminos por los cuales se forman los paralelos son:

- Primer paralelo:
 - Barra8 a Barra1:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
 - (Barra2, Barra1) : {[Interruptor4, Transformador31, Interruptor2]}
 - Barra1 a Barra1
- Segundo paralelo:
 - Barra8 a Barra1:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
 - (Barra2, Barra1) : {[Interruptor5, Transformador32, Interruptor3]}
 - Barra1 a Barra1:

En las Figuras 4.2.1.3.5 y 4.2.1.3.6 se pueden ver los paralelos mencionados.

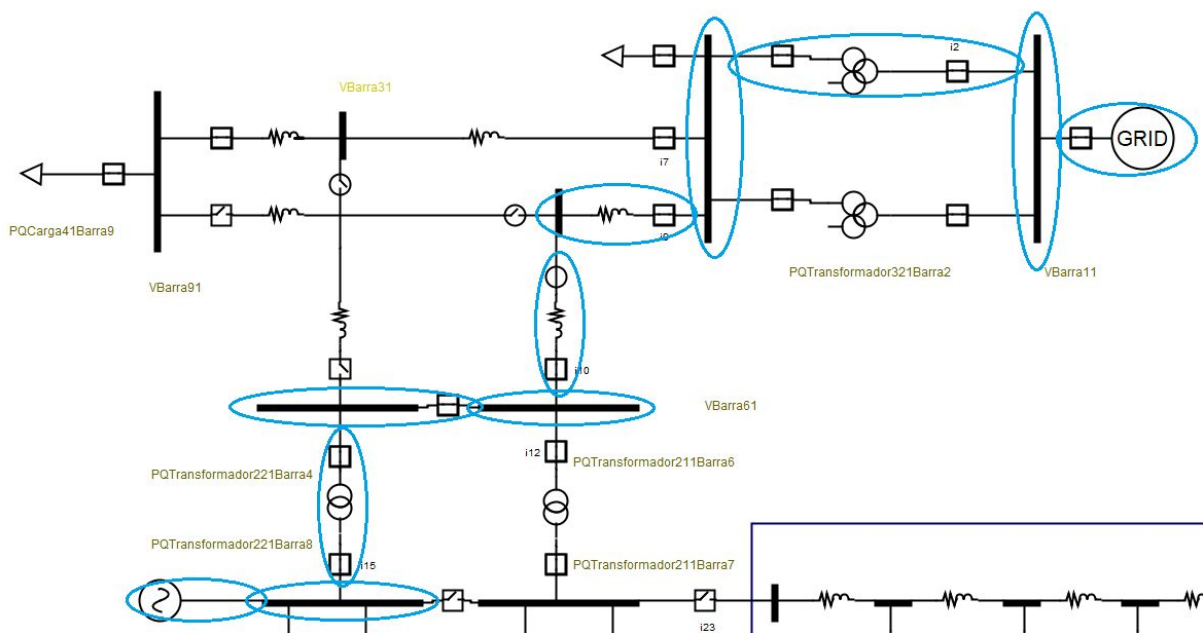


Figura 4.2.1.3.5: Primer paralelo, la Barra1 recibe energía de la Barra8 y de sí misma.

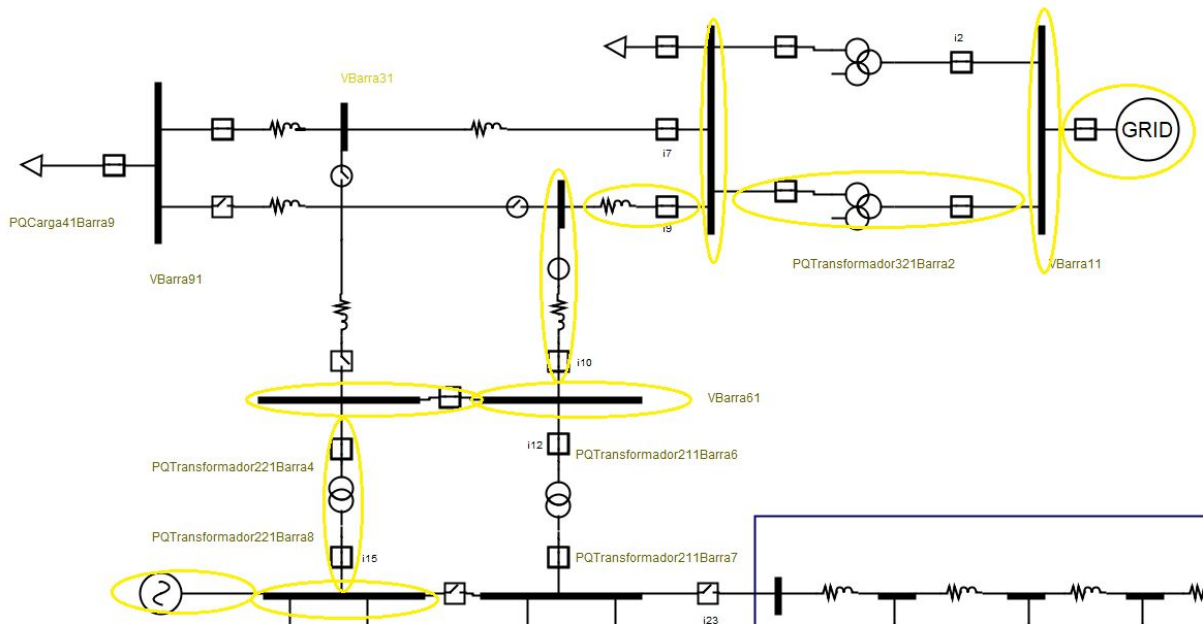


Figura 4.2.1.3.6: Segundo paralelo, la Barra1 recibe energía de la Barra8 y de sí misma.

A su vez, la Barra7 recibe energía de la Barra1 y de la Barra8. Recordemos que la Barra1 tiene asociado el Generador Slack1 y la Barra8 tiene asociado el Generador2. Los caminos por los cuales se forman los paralelos son:

- Primer paralelo:
 - Barra1 a Barra7:
 - (Barra1, Barra2) : [[Interruptor4, Transformador31, Interruptor2]]
 - (Barra2, Barra5) : [[Interruptor9, Linea3]]
 - (Barra5, Barra6) : [[Interruptor10, Linea4, Seccionador2]]
 - (Barra6, Barra7) : [[Interruptor12, Transformador21, Interruptor14]]
 - Barra8 a Barra7:
 - (Barra8, Barra4) : [[Interruptor13, Transformador22, Interruptor15]]
 - (Barra4, Barra6) : [[Interruptor11]]
 - (Barra6, Barra7) : [[Interruptor12, Transformador21, Interruptor14]]
- Segundo paralelo:
 - Barra1 a Barra7:
 - (Barra1, Barra2) : [[Interruptor4, Transformador31, Interruptor2]]
 - (Barra2, Barra5) : [[Interruptor9, Linea3]]

- (Barra5, Barra6) : {[Interrupor10, Linea4, Seccionador2]}
- (Barra6, Barra7) : {[Interrupor12, Transformador21, Interrupor14]}
- Barra8 a Barra7:
 - (Barra8, Barra4) : {[Interrupor13, Transformador22, Interrupor15]}
 - (Barra4, Barra6) : {[Interrupor11]}
 - (Barra6, Barra7) : {[Interrupor12, Transformador21, Interrupor14]}

Estos paralelos obtenidos están representados en la Figura 4.2.1.3.7 y en la Figura 4.2.1.3.8.

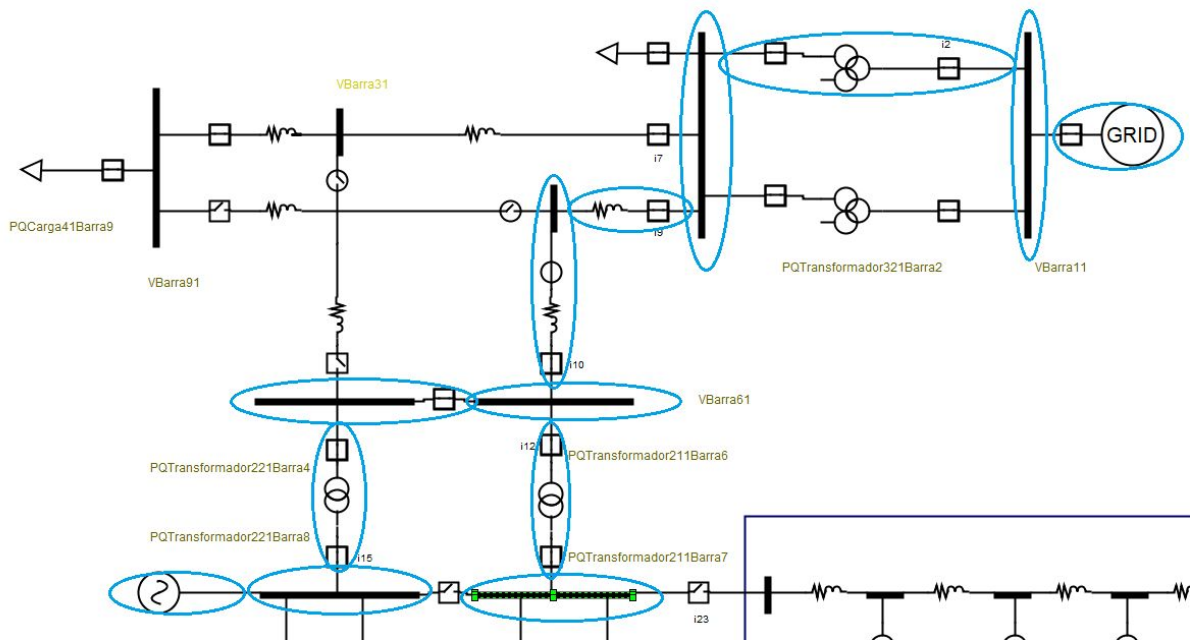


Figura 4.2.1.3.7: Primer paralelo, la Barra7 recibe energía de la Barra1 y la Barra8.

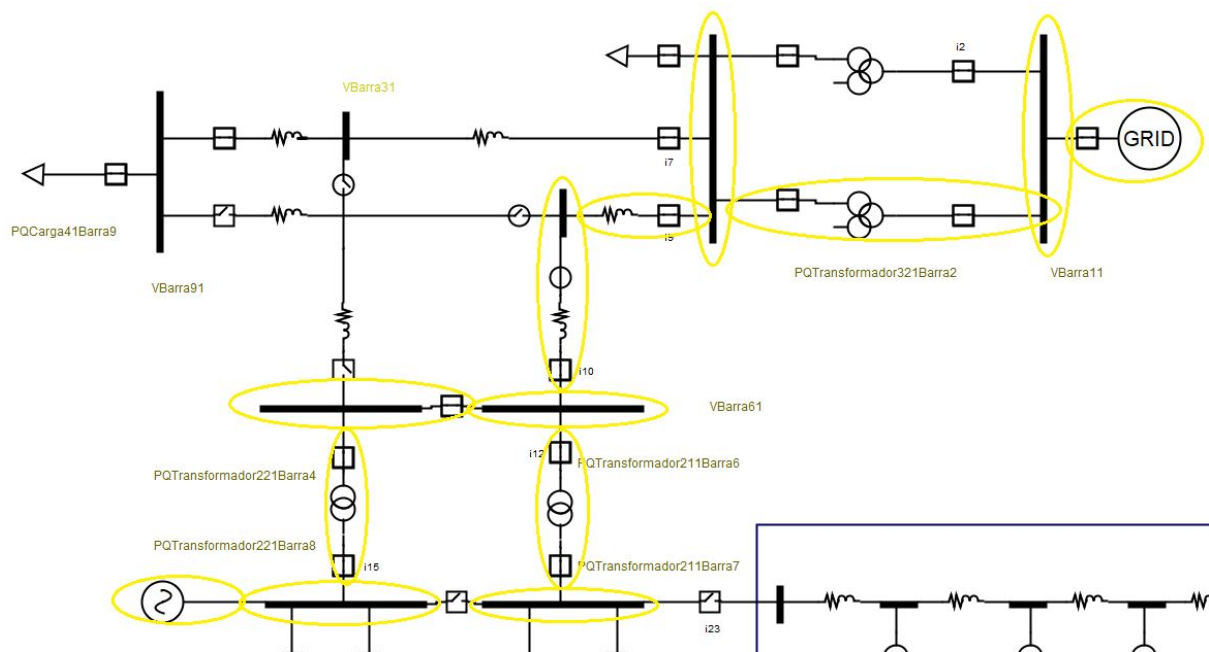
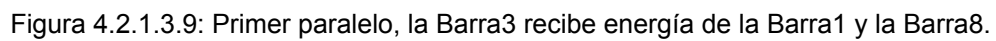


Figura 4.2.1.3.8: Segundo paralelo, la Barra7 recibe energía de la Barra1 y la Barra8.

La Barra1 y de la Barra8 también le transmiten en energía la Barra3. Los caminos por los cuales se forman los paralelos son:

- Primer paralelo:
 - Barra1 a Barra3:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
 - Barra8 a Barra3:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
- Segundo paralelo:
 - Barra1 a Barra3:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra3) : {[Interruptor7, Linea1]}
 - Barra8 a Barra3:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}

- En efecto, estos paralelos quedan representados en las Figuras 4.2.1.3.9 y 4.2.1.3.10.



Al mismo tiempo, la Barra2 recibe energía de la Barra1 y de la Barra8. La Barra1 tiene asociado el Generador Slack1 y la Barra8 tiene asociado el Generador2. Los caminos por los cuales se forman los paralelos son:

- Primer paralelo:
 - Barra1 a Barra2:
 - (Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}
 - Barra8 a Barra2:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}
- Segundo paralelo:
 - Barra1 a Barra2:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - Barra8 a Barra2:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
 - (Barra5, Barra2) : {[Interruptor9, Linea3]}

Por lo tanto en la Figura 4.2.1.3.11 y en la Figura 4.2.1.3.12 se pueden observar los paralelos anteriores.

- (Barra2, Barra5) : {[Interruptor9, Linea3]}
- Barra8 a Barra5:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}
- Segundo paralelo:
 - Barra1 a Barra5:
 - (Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}
 - (Barra2, Barra5) : {[Interruptor9, Linea3]}
 - Barra8 a Barra5:
 - (Barra8, Barra4) : {[Interruptor13, Transformador22, Interruptor15]}
 - (Barra4, Barra6) : {[Interruptor11]}
 - (Barra6, Barra5) : {[Interruptor10, Linea4, Seccionador2]}

Así pues los paralelos obtenidos se pueden ver en la Figura 4.2.1.3.13 y en la Figura 4.2.1.3.14.

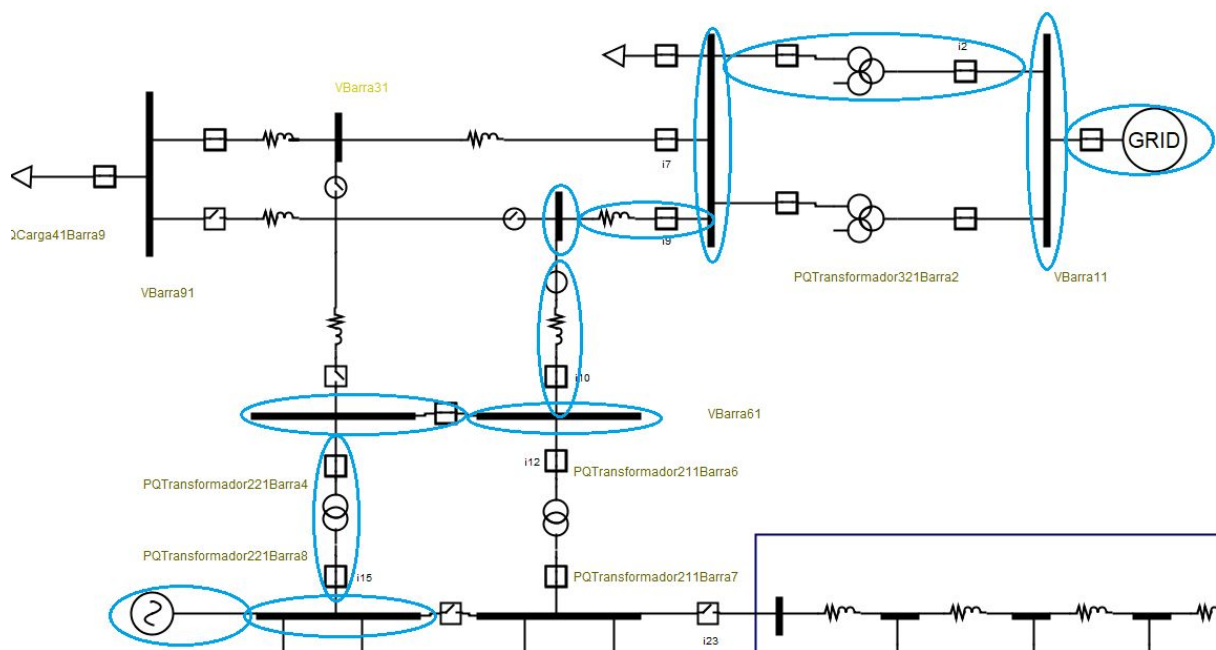


Figura 4.2.1.3.13: Primer paralelo, la Barra5 recibe energía de la Barra1 y la Barra8.

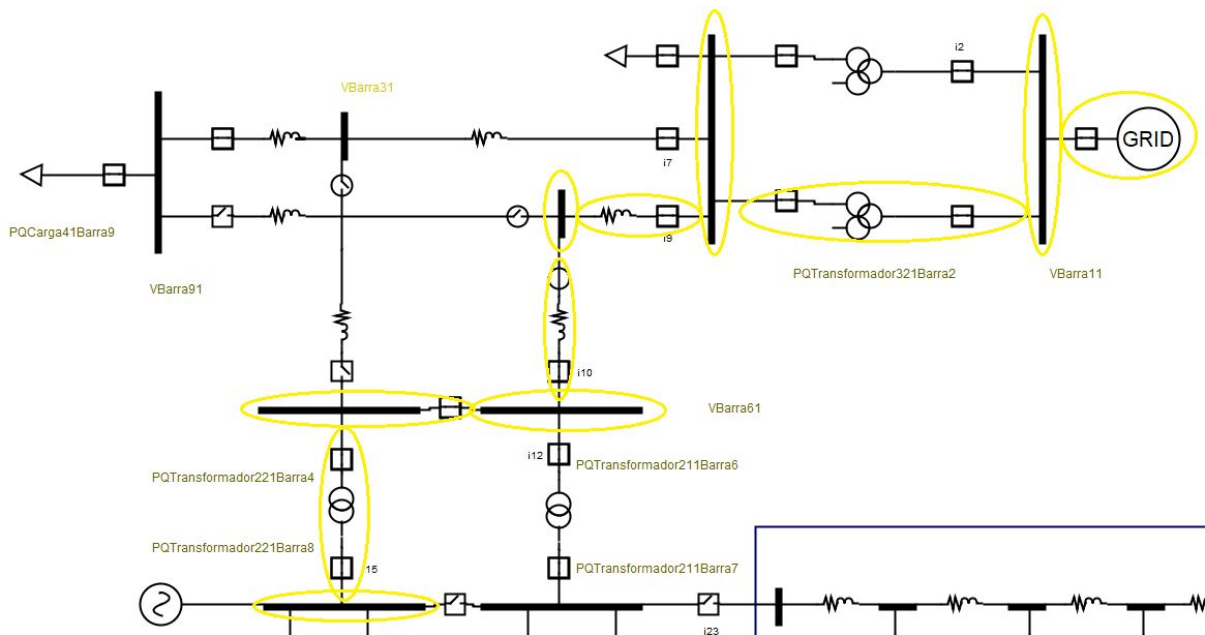


Figura 4.2.1.3.14: Segundo paralelo, la Barra5 recibe energía de la Barra1 y la Barra8.

Luego de analizar los resultados se puede afirmar que el resultado obtenido es correcto ya que la apertura de los interruptores 22 y 23 no impiden el paso de la corriente de las fuentes al resto del circuito. Es decir que con el GeneradorSlack1 y Generador1 se forman paralelos en la red eléctrica.

Se debe recordar que los tiempos de construcción de los grafos son los mismos que se obtuvieron anteriormente, ya que el proceso no cambia para calcular zonas desenergizadas, lazos y paralelos.

Por último, el tiempo promedio de ejecución para el algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener paralelos (3.3.3) fue de 1.91 ms, y el tiempo de respuesta promedio para la transformación de los resultados del grafo reducido al grafo real (3.4) que se obtuvo fue de 4.14 ms. Nuevamente, la transformación de los resultados es más costoso ya que no sólo se debe recorrer el grafo sino también un hash que contiene los resultados pero con datos del grafo reducido.

Para finalizar, se puede concluir que tanto la construcción del grafo con la del grafo reducido son los procesos más costosos, y para las 3 detecciones no hay gran diferencia entre los tiempos de los algoritmos de búsqueda y la transformación de los resultados al grafo real.

4.2.2. Caso 2: Apertura de los interruptores 23, 22 y 9

Para este caso se realizó la apertura de los interruptores 22, 23 y 9 sobre el entorno del editor como se puede observar en la Figura 4.2.2.1.

Nuevamente primero se analizaron las zonas desenergizadas (2.7.1.1), y luego se realizó la búsqueda de lazos (2.7.1.2) y paralelos (2.7.1.3) como consecuencia de la apertura de dichos interruptores.

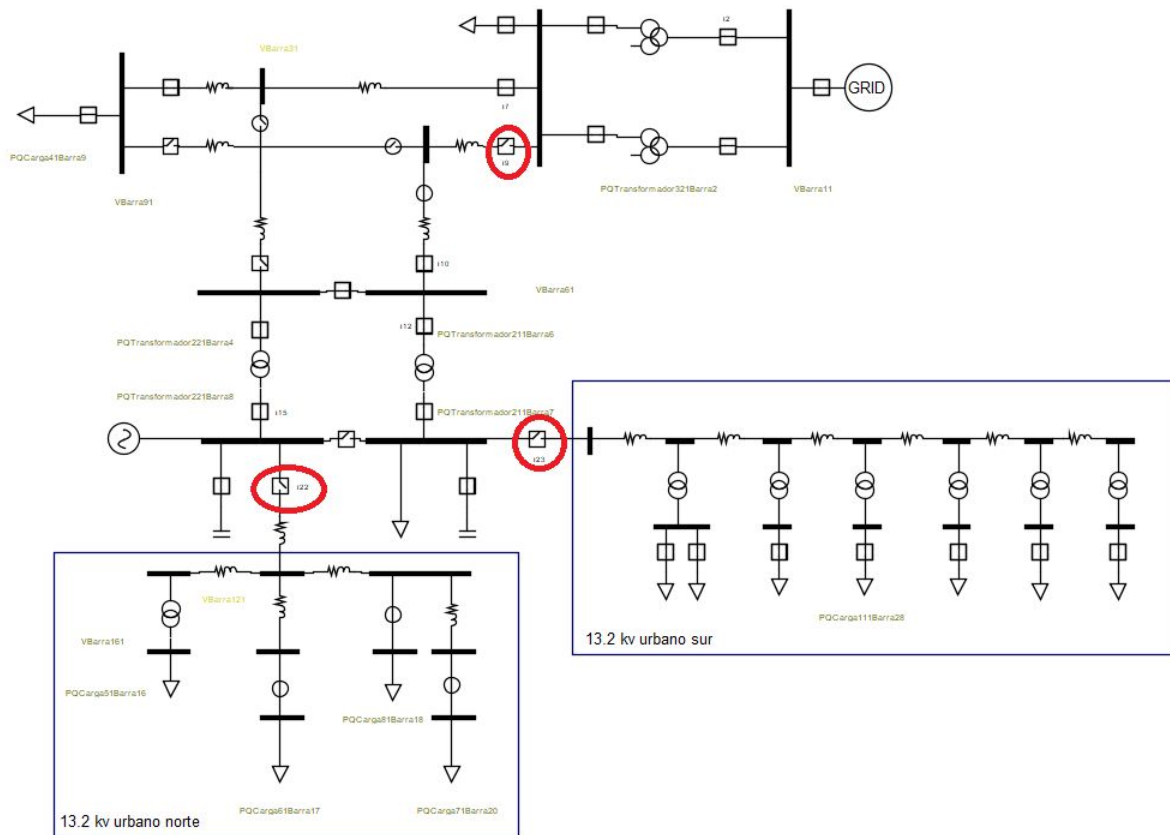


Figura 4.2.2.1: Apertura de los interruptores 22, 23 y 9 en un sector de la red de Armstrong.

Luego teniendo en cuenta la red con los cambios de estado realizados se construye un grafo asociado (3.2.2). Después de haber realizado la apertura de los interruptores se obtiene el mismo grafo que el caso anterior (4.2.1).

A continuación se crea un grafo reducido (3.2.3) sobre el cual se van aplicar los distintos algoritmos de recorrido (3.3). Este grafo es diferente al caso anterior ya que se pierde un arco al abrir el interruptor 9. El arco que no existe respecto del caso 1 es:

- Barra2_Fic, Barra5_Fic: Linea3

4.2.2.1. Zonas desenergizadas

Al ejecutar el algoritmo de búsqueda (3.3.1) sobre el grafo reducido se obtuvo como resultado las mismas 22 barras desenergizadas que en el caso 1 (Figura 4.2.1.1.1, 4.2.1.1.1).

Con respecto a la Figura 4.2.1.1.1 se puede decir que el resultado es correcto debido a que los interruptores 22 y 23 son los únicos componentes que vinculan las barras desenergizadas con las fuentes de la red eléctrica, y la apertura del interruptor 9 no provoca que las barras adyacentes al mismo queden sin energía. Se puede destacar que las fuentes son GeneradorSlack1 y Generador1.

Como resultado de las pruebas se obtuvo que el tiempo total de ejecución asociado al proceso de detección desde el grafo inicial es 1480.52 ms en promedio.

Los pasos que se tuvieron en cuenta para calcular el tiempo total son los siguientes:

- Construcción del grafo a partir del modelo CIM del circuito (3.2.2): 995.28 ms.
- Construcción del grafo reducido a partir del grafo anterior (3.2.3): 482.94 ms.
- Ejecución del algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener zonas desenergizadas (3.3.1): 1.15 ms.
- Transformación de los resultados del grafo reducido al grafo real (3.4): 1.15 ms.

De forma similar al caso anterior, la construcción del grafo a partir del modelo es más costoso que la construcción del grafo reducido debido a que para el primero se debe recorrer toda la red por completo para crear todos los nodos y las conexiones entre éstos. Además, los tiempos del algoritmo de búsqueda y de la transformación los de resultados son prácticamente iguales con el redondeo en el segundo decimal.

4.2.2.2. Lazos

Después de analizar las zonas desenergizadas se efectuó la búsqueda de lazos (2.7.1.2). Ejecutando el algoritmo de búsqueda (3.3.2) obtuvimos la siguiente salida:

Lazos:

Barra 1:Barra1, Barra 2:Barra9

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

(Barra3, Barra9) : {[Interruptor19, Linea5]}

Barra 1:Barra1, Barra 2:Barra3

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

(Barra2, Barra3) : {[Interruptor7, Linea1]}

Barra 1:Barra1, Barra 2:Barra2

Recorrido 0:

(Barra1, Barra2) : {[Interruptor4, Transformador31, Interruptor2]}

Recorrido 1:

(Barra1, Barra2) : {[Interruptor5, Transformador32, Interruptor3]}

En este caso se puede ver que en los resultados obtenidos hay tres lazos, y estos son iguales a los obtenidos en el caso anterior (4.2.1.2). Por un lado, en las figuras 4.2.1.2.1 y 4.2.1.2.2 se ve reflejado el lazo existente entre la Barra1 y la Barra9. En cambio el lazo que existe entre la Barra1 y la Barra3 está representado en las figuras 4.2.1.2.5 y 4.2.1.2.6. Por último las figuras 4.2.1.2.7 y 4.2.1.2.8 muestran el lazo que hay entre la Barra1 y la Barra2.

De los resultados se puede determinar que la apertura de los interruptores 22, 23 y 9 no impiden el paso de la corriente de las fuentes al resto del circuito, por lo tanto se puede afirmar que el resultado obtenido es correcto. Es decir que con el GeneradorSlack1 y Generador1 se forman lazos.

Por otro lado, los tiempos de construcción promedio del primer grafo (3.2.2) y del reducido (3.2.3) son los mismos que se obtuvieron durante el estudio de zonas desenergizadas (3.3.1). En cambio los tiempos promedio del algoritmo de búsqueda (3.3.2) y de la conversión de los resultados (3.4) son diferentes. Luego de ejecutar diez veces la aplicación los tiempos promedios fueron de 1.74 ms para el algoritmo de búsqueda y 1.34 ms para la conversión de los resultados.

Para concluir no hay gran diferencia con los tiempos obtenidos en el caso 1 (4.2.1), aunque son levemente menores para este caso ya que el volumen de la salida que hay que convertir al grafo original es mucho más pequeño.

4.2.2.3. Paralelos

Por último se buscan los paralelos (2.7.1.3) dentro del circuito (Figura 4.2.2.1). Estas circunstancias se dan cuando una barra recibe corriente de dos fuentes de energía distintas.

Pero para este caso no se obtienen paralelos ya que al estar abierto el interruptor 9 corta la comunicación entre las dos fuentes de energía. Al no tener caminos en común el Generador2 y el GeneradorSlack1 no es posible que se encuentren paralelos en la red eléctrica.

Los tiempos de respuesta de los algoritmos fueron:

- Ejecución del algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener paralelos (3.3.3): 1.85 ms.
- Transformación de los resultados del grafo reducido al grafo real (3.4): 0.85 ms.

Recordemos que para el caso 1 (4.2.1) estos tiempos habían sido de 1.91 ms y de 4.14 ms respectivamente. Así, se puede concluir que entre ambos casos no hay diferencia en el tiempo de ejecución del algoritmo de búsqueda, con la diferencia de que en este caso de estudio no hay paralelos debido a la apertura del interruptor 9.

4.2.3. Caso 3: Apertura de los interruptores 23, 22, 9 y 2

Por último se realizó la apertura de los interruptores 22, 23, 9 y 2 sobre el mismo escenario de la Figura 4.2.3.1.

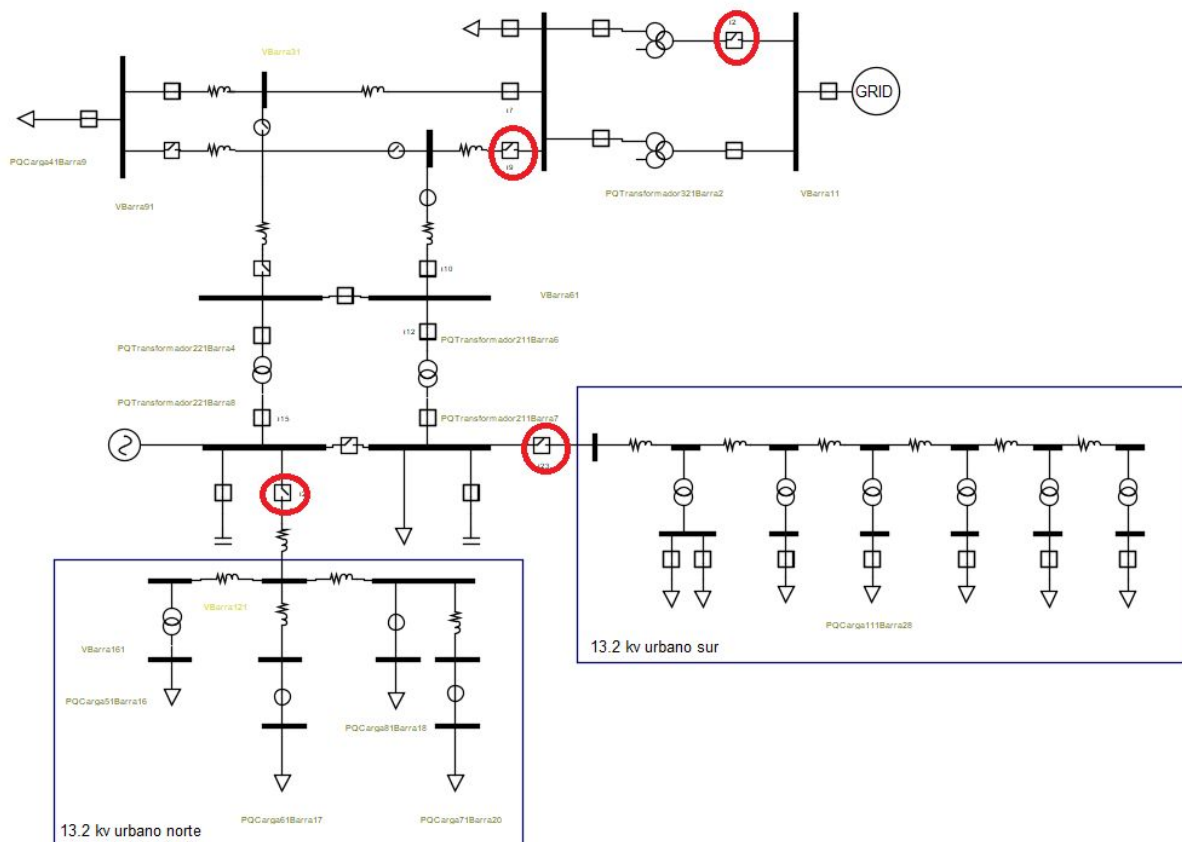


Figura 4.2.3.1: Apertura de los interruptores 22, 23, 9 y 2 en un sector de la red de Armstrong.

Nuevamente, teniendo en cuenta la red luego de aplicar los cambios de estado de los interruptores se construye un grafo asociado, obteniendo el mismo grafo que los dos casos anteriores.

No obstante, el grafo reducido sobre el cual se realizan los recorridos es diferente a los dos casos anteriores ya que se pierde un nuevo arco al abrir el interruptor 2. El resto del grafo reducido es el mismo que se obtuvo para el caso 1 (4.2.1). Los arcos que no existen en la nueva salida son:

- Barra2_Fic, Barra5_Fic: Linea3
- Barra2_Fic, Barra1_Fic: Transformador31

4.2.3.1. Zonas desenergizadas

Luego de ejecutar el algoritmo de búsqueda (3.3.1) sobre el grafo reducido se obtuvo como resultado las mismas 22 barras desenergizadas que en el caso 1 (Figura 4.2.1.1.1, 4.2.1.1) y el caso 2 (4.2.2.1).

En cuanto al tiempo total ejecución, dió como resultado 1432.69 ms. Cada parte tuvo asociado los siguientes tiempos:

- Construcción del grafo a partir del modelo CIM del circuito (3.2.2): 961.72 ms.
- Construcción del grafo reducido a partir del grafo anterior (3.2.3): 468.83 ms.
- Ejecución del algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener zonas desenergizadas (3.3.1): 1.08 ms.
- Transformación de los resultados del grafo reducido al grafo real (3.4): 1.06 ms.

Nuevamente, la construcción del grafo a partir del modelo es más costoso que la construcción del grafo reducido debido a que para el primero se debe recorrer toda la red por completo para crear todos los nodos y las conexiones entre éstos. También los tiempos del algoritmo de búsqueda y de la transformación los de resultados son prácticamente iguales.

4.2.3.2. Lazos

El siguiente paso después de analizar las zonas desenergizadas fue la búsqueda de lazos (2.7.1.2). Luego de ejecutar el algoritmo de búsqueda (3.3.2) no se obtuvieron lazos, debido a que en la apertura del interruptor 2 deja de existir dos caminos distintos entre la Barra1 y la Barra2.

Luego, los tiempos de respuesta de los algoritmos fueron:

- Ejecución del algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener lazos (3.3.2): 1.66 ms.
- Transformación de los resultados del grafo reducido al grafo real (3.4): 0.71 ms.

Estos mismos tiempos para el caso 1 fueron de 1.91 ms y de 4.14 ms respectivamente, y para el segundo caso 1.74 ms y 1.34 ms. Se puede observar que prácticamente no hay diferencia entre los tiempos de ejecución del algoritmo de búsqueda para los tres casos.

4.2.3.3. Paralelos

Finalmente se buscan los paralelos (2.7.1.3) dentro del circuito (Figura 4.2.3.1). Al igual que el caso 2 (4.2.2.3), no se obtienen paralelos a causa de estar abierto el interruptor 9 que corta la comunicación entre las dos fuentes de energía. Como no

hay caminos en común el Generador2 y el GeneradorSlack1 no es posible que se encuentren paralelos en la red eléctrica.

Los tiempos de respuesta de los algoritmos fueron:

- Ejecución del algoritmo de búsqueda para recorrer el grafo reducido con el fin de obtener paralelos (3.3.3): 1.66 ms.
- Transformación de los resultados del grafo reducido al grafo real (3.4): 0.73 ms.

Para el caso 2 (4.2.2.3) estos tiempos habían sido de 1.85 ms y de 0.85 ms respectivamente, y no hay casi diferencia con este caso.

4.3. Conclusión

Durante este capítulo se evaluaron tres casos sobre un sector de la red eléctrica de Armstrong. Para cada uno de ellos se decidió abrir ciertos interruptores y ver cómo respondía la aplicación a los siguientes puntos:

- La construcción del grafo completo (3.2.2).
- La construcción del grafo reducido (3.2.3).
- Búsqueda de zonas desenergizadas (3.3.1).
- Búsqueda de lazos (3.3.2).
- Búsqueda de paralelos (3.3.3).
- Conversión de los resultados al grafo completo (3.4).

Luego de realizar diez pruebas para cada uno de los casos se pudo observar que en la construcción del grafo completo (3.2.2) y el grafo reducido (3.2.3) los tiempos promedio tienen una diferencia despreciable. En cambio, entre las búsquedas se puede observar una diferencia. Los tiempos promedio de búsqueda de lazos (3.3.2) y paralelos (3.3.3) es un poco mayor a la búsqueda de zonas desenergizadas (3.3.1). Esto es debido a que las primeras requieren buscar todos los caminos posibles en el grafo y por lo tanto hay nodos que seguramente se vuelvan a visitar.

Para finalizar, en relación a la conversión de los resultados obtenidos de las búsquedas (3.4), los tiempos promedio para zonas desenergizadas son menores que los tiempos promedio de la transformación de los lazos y los paralelos. La causa de esto es que las estructuras donde se almacenaron los lazos (2.7.1.2) y paralelos (2.7.1.3) son más complejas de recorrer y luego reconstruirlas con los datos del grafo completo.

Capítulo 5

Conclusiones

En este trabajo se desarrolló una herramienta de NCA para asistir al administrador u operador de la red de energía eléctrica. Sobre la base de los estados de los interruptores (2.6) permite obtener las zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3) dentro de la red. Esta información es de extrema importancia a la hora de tener que tomar de decisiones ante fallos o cortes del servicio.

Para una mayor interoperabilidad en la representación de la red se trabajó sobre un modelo CIM (Common Information Model) (2.1). Su función principal es describir los datos de comunicación entre las distintas herramientas empleadas para la gestión de la red, como por ejemplo un DMS (Distribution Management System).

Ante la necesidad de instanciar el modelo CIM se utilizó grafos RDF (2.2) y Jena (2.3). Este último provee un framework para manejar y administrar datos representados con grafos RDF. Y finalmente, para persistir la información en RDF se utilizó RDF/XML (2.4). En todos los casos se analizaron los pros y contras que ayudaron a decidir sobre las distintas tecnologías.

Durante la implementación del NCA surgió la necesidad de aplicar algoritmos de búsqueda, por lo tanto se modeló una clase Grafo (2.5). Este tipo de estructuras permiten estudiar las interrelaciones entre unidades que interactúan unas con otras. El estado de los interruptores (2.6) juega un papel muy importante para calcular la topología de la red. En consecuencia se creó a partir del Grafo (3.2.2) un “Grafo Reducido” (3.2.3), que eliminó interruptores cerrados y abiertos, y fusionó barras conectadas por un interruptor cerrado. El Grafo Reducido también se debía encarga de la conversión de los resultados (3.4) al grafo original, ya que cada nodo hacía referencia a una “barra ficticia”; es decir si se fusionaban dos barras se creaba una “barra ficticia” que hacía referencia a las mismas.

Para la representación con grafos se aplicaron conceptos de programación orientada a objetos y patrones de diseño. Primero se utilizó Abstract Factory (3.2.1) para la creación de los distintos tipos de grafos. Luego se escogió el patrón de diseño Strategy (3.3) para la implementación de los algoritmos de búsqueda de zonas desenergizadas (2.7.1.1), lazos (2.7.1.2) y paralelos (2.7.1.3).

Los algoritmos de búsqueda desarrollados son una variante del DFS (Depth-First Search) (2.5.1). Estos algoritmos se aplicaron en el Grafo Reducido (3.2.3) construido sobre el análisis del estado de los interruptores. Luego de obtener los resultados del proceso de detección se realizó la conversión al grafo original (3.4).

Finalmente se plantearon tres casos de estudios (4.2) con el fin de analizar los resultados obtenidos. En cada caso se tuvo en cuenta el tiempo de ejecución referidos a:

- Construcción del grafo asociado al circuito de energía eléctrica (3.2.2).
- Construcción del grafo reducido (3.2.3).
- Ejecución de los algoritmos de búsqueda para recorrer el grafo reducido con el fin de obtener zonas desenergizadas, lazos y paralelos (3.3).
- Conversión de los resultados al grafo original (3.4).

Para ello se ejecutó diez veces la aplicación en cada caso y de los tiempos obtenidos se calculó el promedio. Se pudo observar que en la construcción del grafo completo (3.2.2) y el grafo reducido (3.2.3) los tiempos promedio tuvieron una diferencia despreciable. Sin embargo, los tiempos promedio de búsqueda de lazos (3.3.2) y paralelos (3.3.3) fueron un poco mayor a la búsqueda de zonas desenergizadas (3.3.1). Esto se debe a que la búsqueda de lazos y paralelos se realiza sobre todos los caminos posibles.

Por último, con respecto a la conversión de los resultados obtenidos de las búsquedas (3.4), el tiempo promedio para zonas desenergizadas fue menor que los tiempos promedio de la conversión de los lazos y los paralelos. La razón de esto fue que las estructuras donde se almacenaron los lazos y paralelos son más complejas de recorrer, y requieren de un proceso de reconstrucción sobre los datos del grafo completo.

5.1. Trabajos futuros

La herramienta desarrollada presenta distintos aspectos a mejorar, los cuales podrían implementarse en trabajos futuros. A continuación se describen algunos de ellos.

Actualmente la aplicación se ejecuta en modo simulación (a pedido del operador), no obstante se puede modificar para que pueda ser utilizada en tiempo real. De ello se desprende la necesidad de analizar tanto los métodos de construcción de los grafos como los algoritmos de búsqueda de zonas desenergizadas, lazos y paralelos, con el fin de aumentar su eficiencia y lograr la ejecución en tiempo real.

Otro punto es la detección de cambios de estados de los interruptores dentro de la red para que se ejecute automáticamente la construcción del grafo reducido y los diferentes algoritmos de búsqueda. Por ahora la herramienta no cuenta con esta funcionalidad, lo cual sería de extrema necesidad durante la ejecución en tiempo real.

Detectar interruptores que energizan una zona podría ayudar a los operadores en la toma de decisiones. La aplicación actual detecta zonas desenergizadas, pero no marca que interruptores son los que abren el paso de la energía para dichas zonas. De todos modos, cuando se detectan lazos y paralelos se puede ver el camino completo incluyendo los interruptores que son parte de los resultados. Esto podría ser de ayuda para las futuras mejoras.

Por último, el NCA debería tener la capacidad de modificar la visualización de la red eléctrica en tiempo real. Sería de mucha utilidad que el usuario pueda seleccionar el perfil de visualización que desea, y la aplicación ajuste el coloreado de la red de acuerdo al estado actual y a los valores que pueden ser teledados. Por ejemplo, coloreado según zonas energizadas y desenergizadas, o coloreados de acuerdo a los niveles de tensión (alta, baja y media tensión).

Bibliografía

- [1] Aho A., Ullman J., "Foundations of Computer Science, Computer Science Press C Edition".
- [2] Análisis de Interruptores de Media y Alta Tensión". Recuperado el 28/05/2018 de [<http://www.sectorelectricidad.com/14486/analisis-de-interruptores-de-media-y-alta-tension>].
- [3] "An Introduction to RDF and the Jena RDF API". Recuperado el 15/03/2018 de [https://jena.apache.org/tutorials/rdf_api.html].
- [4] Antunez M., Boroni G., Fiaschetti L., Risso M., Rubiales A., "Incorporación del estándar CIM en sistemas de gestión de energía".
- [5] Antunez M., Boroni G., Fiaschetti L., Risso M., Rubiales A., Trapani E., Valenzuela L., "Monitoring and controlling energy distribution: Implementation of a distribution management system based on Common Information Model", Accepted 23 June 2017.
- [6] Apache jena (2011-2015). Recuperado de [<https://jena.apache.org/index.html>].
- [7] Chen Y.Z., Sun Y.Z., Wu L.L., Feng D., Zhou Q., Xie H.F., "Cim/svg based graphics-module integration power platform with secondary development of visio". In: Applied Mechanics and Materials. vol. 446, pp. 1636-1641. Trans Tech Publ (2014).
- [8] Cim users group (2009-2016). Recuperado de [<http://cimug.ucaiug.org/>].
- [9] Clements K. A., Davis P.W., "Detection and Identification of Topology Errors in Electric Power Systems," IEEE Trans. Power Syst., vol. 3, no. 4, pp. 1748-1753, Nov. 1988.
- [10] Combariza Germán, Profesor del Colegio Santa María Bogotá-Colombia, "Una introducción a la teoría de grafos", 2003.
- [11] Cormen T., Lieserson C., Rivest R., "Introduction to Algorithms", Third Edition.
- [12] "Distribution Management System". Recuperado el 12/04/2018 de [<http://distributionmanagementsystem.blogspot.com>].
- [13] "Distribution Systems: DMS, Volt/Var control". Recuperado el 23/04/2018 de [<https://www.electrical-guru.com/Subject.aspx?id=3&code=6EE5A&unitid=3&topicid=23>].
- [14] Farhangi H., "The Path of the Smart Grid", IEEE Power & Energy Magazine, vol. 8, pp. 18-28, Jan/Feb. 2010.
- [15] Frances B.M.T., Koc Y., Kooij R.E., Warnier M., "Structural vulnerability assessment of electric power grids", Electr. Power Syst. Res., vol. 81, pp. 1334–1340, 2011.
- [16] "IntelliGrid Common Information Model Primer", Second Edition (2013).
- [17] Kolluri S., Werho T., Vittal V., Wong S.M., "Power System Connectivity Monitoring Using a Graph Theory Network Flow Algorithm".
- [18] Moreno R., Torres A., "Network Topological Analysis to Assess the Power Smart Grid", 2012.

- [19] Moreno R., Torres A., "Topological Characterization of the Power Systems: Security Applications", to be submitted to IEEE Trans. Power Syst, 2012.
- [20] "Network Connectivity Analysis". Recuperado el 29/06/2018 de [https://etap.com/es/product/network-connectivity-analysis].
- [21] Oscar Javier Blancarte Iturralde, "Introducción a los patrones de diseño, un enfoque práctico", Primera Edición".
- [22] "Patrón Abstract Factory". Recuperado el 08/09/2018 de [http://lineadecodigo.com/patrones/patron-abstract-factory/].
- [23] "Pruebas y Análisis de Interruptores de Alta Tensión". Recuperado el 02/06/2018 de [http://www.emb.cl/electroindustria/articulo.mvc?xid=1872].
- [24] Réka A., István A., Gary L.N., "Structural Vulnerability of the North American Power Grid", 2004.
- [25] Simmins J.J., "The impact of pap 8 on the common information model (cim)". In: Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES. pp. 1-2. IEEE (2011).
- [26] "Sistemas de distribución de C/A". Recuperado el 14/05/2018 de [http://apuntescientificos.org/redes.html].
- [27] "Sobre Etap". Recuperado el 29/06/2018 de [https://etap.com/es/company/about-us].
- [28] "Strategy Pattern". Recuperado el 25/09/2018 de [http://codigolinea.com/2015/03/19/strategy-pattern-patron-estrategia].
- [29] "TENERGIA – Soluciones tecnológicas aplicadas a la distribución de redes eléctricas". Recuperado de [http://www.pladema.net/pladema/2014/06/03/tenergia-soluciones-tecnologicas-aplicadas-a-la-distribucion-de-redes-electricas/].
- [30] "Topologías de las Redes de Distribución". Recuperado el 14/05/2018 de [https://distribucion.webnode.com.co/topologias-de-las-redes-de-distribucion/].
- [31] Torres A., Anders G., "Innovations in Power Systems Reliability, Chapter: Strategic Lines and Substations in an Electric Power Network". Springer Series in Reliability Engineering, pp. 169-190, 2011.
- [32] "Un apagón afectó a gran parte de la Ciudad de Buenos Aires". Recuperado el 19/10/2018 de [https://www.infobae.com/sociedad/2018/06/06/un-apagon-afecta-a-gran-parte-de-la-ciudad-de-buenos-aires/].
- [33] "Varios barrios sin luz. Apagón en Capital: problemas en siete subestaciones afectaron a 290 mil vecinos". Recuperado el 19/10/2018 de [https://www.clarin.com/ciudades/apagon-ciudad-buenos-aires-falla-red-alta-tension_0_rJ6y2srx7.html].