

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTROMECAÁNICA

**Modelado de la red de distribución eléctrica del ICE haciendo
uso de software de simulación y Python en un entorno GIS
para líneas de baja tensión, líneas de media tensión, cargas y
transformadores**

Trabajo final de graduación para obtener el grado de:
**Licenciatura en Ingeniería en Mantenimiento
Industrial**

Marvin Andrés Mejías Núñez

Campus Central, Cartago, Costa Rica

Junio, 2024

Carrera acreditada por:



Modelado de la red de distribución eléctrica del ICE haciendo uso de software de simulación y Python en un entorno GIS para líneas de baja tensión, líneas de media tensión, cargas y transformadores © 2024 por Marvin Andrés Mejías Núñez se encuentra bajo licencia CC BY 4.0. Para ver una copia de esta licencia, visite <https://creativecommons.org/licenses/by/4.0/>

TEC | Tecnológico
de Costa Rica
SIBITEC

AUTORIZACIÓN DE PUBLICACIÓN

Fecha: 5 June, 2024

Señores
Instituto Tecnológico de Costa Rica
Sistema de Bibliotecas del Tecnológico

Yo, Marvin Andrés Mejías Núñez, envío al Sistema de Bibliotecas del Tecnológico (SIBITEC), el Trabajo Final de Graduación, del cual soy autor y fue presentado para optar por el grado de licenciatura, en la fecha 5 June, 2024, con el título del proyecto:

Modelado de la red de distribución eléctrica del ICE haciendo uso de software de simulación y Python en un entorno GIS para líneas de baja tensión, líneas de media tensión, cargas y transformadores

Para ser publicado en:

- Repositorio Institucional y Catálogo SIBITEC, con el objetivo de ser visualizado a través de la red Internet, o
- Catálogo SIBITEC, uso exclusivo de la Comunidad TEC.
- Autorizo con embargo¹ Fecha para publicar:

En el caso de que no se pueda publicar seleccione:

- Confidencial Justificación:

Datos personales:

Nombre autor: Marvin Andrés Mejías Núñez
Carrera en la que se presenta el TFG: Ingeniería en Mantenimiento Industrial
Campus Tecnológico / Centro: Académico: Tecnológico Central Cartago
Correo/s electrónico/s: mejiasandres08@gmail.com, mejiasandres08@estudiantec.cr
Carnet: 2017102082
Nº de identificación personal: 702710490


Firma del estudiante



LIMPIAR

¹ Embargo: período de tiempo durante el cual el acceso o consulta a una publicación no está disponible

Miembros del tribunal

Informe presentado a la Escuela de Ingeniería Electromecánica del Instituto Tecnológico de Costa Rica como requisito parcial para optar por el título de Ingeniero en Mantenimiento Industrial con el grado de licenciatura.

Magister Gonzalo Mora Jiménez
Supervisor

Magister Gonzalo Mora Jiménez
Lector 1

M.Sc. Gustavo Gómez Ramírez
Lector 2

Magíster Luis Carlos Muñoz Chacón
Asesor académico

Hoja de datos

Información del estudiante

Nombre: Marvin Andrés Mejías Núñez

Cédula: 702710490

Carné ITCR: 2017102082

Dirección de residencia en época lectiva: Casa AA4 diagonal al Súper Mento, Residencial Las Catalinas, El Tejar, El Guarco, Cartago

Teléfono: 8579-9935

Correo electrónico: mejiasandres08@estudiantec.cr

Información del proyecto

Título: Modelado de la red de distribución eléctrica del ICE haciendo uso de software de simulación y Python en un entorno GIS para líneas de baja tensión, líneas de media tensión, cargas y transformadores

Asesor industrial: Magíster Gonzalo Mora Jiménez

Profesor guía: Magíster Luis Carlos Muñoz Chacón

Jurado evaluador: M.Sc. Gustavo Gómez Ramírez y Magíster Gonzalo Mora Jiménez

Información de la empresa

Nombre: Instituto Costarricense de Electricidad

Dirección: Edificio ICE, Sabana Norte, San José

Actividad principal: Suministro de servicios de electricidad y telecomunicaciones

Contacto: Magíster Gonzalo Mora Jiménez

Teléfono: 2000-5728

Resumen

En este trabajo se indica el procedimiento con el cual se procesó, mediante Python, un circuito de la red de distribución del ICE compuesto por líneas de media tensión, líneas de baja tensión, transformadores y cargas para su posterior implementación en OpenDSS y OpenDSS-G. Esto con el fin de obtener información sobre la red de distribución respecto a su estado actual y ante posibles cambios.

Se realizó una minería de datos con el fin de eliminar información no requerida por los programas de simulación, así como limpiar datos que puedan generar errores en la simulación. Posteriormente, mediante un programa creado en Python, se les dio formato a cada tipo de elemento que compone la red de distribución acorde al formato solicitado por OpenDSS y OpenDSS-G. Seguidamente, utilizando archivos maestros, se ingresaron los datos del circuito a los programas de simulación de forma que se pudieron generar análisis con los cuales extraer información del circuito, en forma de reportes, gráficos y mapas de calor de variables relevantes, como lo son la tensión, la corriente y la potencia de la red, considerando siete perfiles de demanda.

Finalmente, se logra la implementación exitosa de los datos de la red de distribución en los programas de simulación y se exploran las capacidades del *software* para el análisis de estos circuitos, con la finalidad de extraer información de la situación actual del sistema, así como posibles cambios que pueda experimentar la red. Además, se brindan recomendaciones sobre la modificación de elementos en los datos de la simulación, la implementación de nuevos circuitos en los programas de simulación y posibles mejoras al código de Python para aumentar su eficiencia.

Palabras clave: OpenDSS, redes de distribución, Python, simulación, minería de datos.

Abstract

In this work, the procedure with which a circuit of the ICE distribution network composed of medium voltage lines, low voltage lines, transformers, and loads was processed using Python for its subsequent implementation in OpenDSS and OpenDSS-G is indicated. This is done in order to obtain information about the distribution network regarding its current state as well as possible changes.

To achieve the above, data mining was first carried out in order to eliminate information not required by the simulation programs as well as to clean data that may generate errors in the simulation. Subsequently, using a program created in Python, each type of element that makes up the distribution network was formatted according to the format requested by OpenDSS and OpenDSS-G. Then, using master files, the circuit data was entered into the simulation programs so that analyses could be generated from which to extract information from the circuit, in the form of reports, graphs, and heat maps of relevant variables such as voltage, current, and power of the network, considering seven demand profiles.

Finally, the successful implementation of the distribution network data in the simulation programs is achieved and the software's capabilities for the analysis of these circuits are explored, in order to extract information from the current situation of the system as well as possible changes that the network may experience. In addition, recommendations are provided on the modification of elements in the simulation data, the implementation of new circuits in the simulation programs, and possible improvements to the Python code to increase its efficiency.

Key words: OpenDSS, distribution circuits, Python, simulation, data mining.

Agradecimientos

Agradezco a mi familia y amigos por todo el apoyo brindado a lo largo de mi carrera y durante la elaboración de este proyecto.

También, agradezco a los profesores involucrados en la realización de este proyecto, por la ayuda y guía brindada.

Finalmente, agradezco al Instituto Costarricense de Costa Rica por la posibilidad de realizar mi trabajo final de graduación en su institución.

Dedicatoria

Dedico este trabajo a mis padres y mis hermanas, debido a su apoyo incondicional en cada decisión que he tomado a lo largo de mi vida.

También dedico este trabajo a todos los compañeros y amigos con los que he trabajado y con los cuales he podido llegar al final de mi carrera.

Finalmente, dedico este trabajo a los profesores que me han permitido convertirme en el profesional que soy hoy día y por la motivación e inspiración que han generado en mí.

Tabla de contenido

Autorización de publicación	i
Miembros del tribunal	ii
Hoja de datos	iii
Resumen	iv
Abstract	v
Agradecimientos	vi
Dedicatoria	vii
Tabla de contenido	viii
Lista de tablas	xi
Lista de figuras	xii
1. Introducción	1
1.1. Reseña de la empresa	1
1.2. Descripción del problema	2
1.3. Objetivos de la investigación	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	4
1.4. Justificación	4

1.5.	Viabilidad	5
1.6.	Metodología	5
1.6.1.	Alcance de la investigación	6
1.6.2.	Limitaciones	6
1.6.3.	Cronograma	7
2.	Marco teórico	10
2.1.	Sistemas eléctricos de potencia	10
2.1.1.	Generación	11
2.1.2.	Transmisión	12
2.1.3.	Distribución	12
2.2.	Elementos de una red de distribución	16
2.2.1.	Barras y líneas	17
2.2.2.	Interruptores y protecciones	19
2.2.3.	Transformadores	21
2.2.4.	Reguladores de tensión	25
2.2.5.	Cargas	29
2.3.	Flujos de potencia	30
2.4.	Sistema de Información Geográfica (GIS)	31
2.5.	Software de simulación	32
2.6.	Python	34
2.7.	Simulación	35
3.	Modelado de la red de distribución	36
3.1.	Archivo maestro	37
3.1.1.	Archivo maestro para cargas con demanda constante	37
3.1.2.	Archivo maestro para cargas con demanda variable	40
3.2.	Fuente de tensión	43
3.3.	Barras	45
3.3.1.	Barras: Minería de datos	46
3.3.2.	Barras: Formato OpenDSS	49

3.4. Líneas	50
3.4.1. Líneas: Minería de datos	54
3.4.2. Líneas: Formato de OpenDSS	59
3.5. Transformadores	64
3.5.1. Transformadores: Formato de OpenDSS	67
3.6. Cargas	77
3.6.1. Cargas: Formato de OpenDSS	79
3.7. Regulador de tensión	81
3.8. OpenDSS-G	84
4. Resultados de investigación	88
4.1. Reportes	88
4.2. Gráficos	89
4.3. Mapas de calor	90
5. Conclusiones y recomendaciones	112
5.1. Conclusiones	112
5.2. Recomendaciones	113
Bibliografía	116
Apéndice A. Código empleado	122
A.1. LimpiadorExcel.py	122
A.2. CreadorInputsOpenDSS.py	123
Apéndice B. WireData de conductores	145
B.1. Conductores AAAC	145
B.2. Conductores AAC	146
B.3. Conductores ACSR	147
B.4. Conductores de cobre	149

Lista de tablas

1.1. Cronograma proyectado de trabajo	8
3.1. Información suministrada de las barras	46
3.2. Información suministrada de las barras con columnas modificadas . . .	47
3.3. Información suministrada de las líneas	57

Lista de figuras

1.1. Diagrama de un sistema de potencia	2
1.2. Metodología del proyecto	6
1.3. Diagrama de Gantt del cronograma proyectado de trabajo	9
2.1. Evolución anual de las líneas de transmisión de Costa Rica, periodo 2006 – 2021	13
2.2. Diagrama de una subestación con dos transformadores	14
2.3. Diagrama de una sola línea de un alimentador de distribución trifásico	16
2.4. Conexión Delta - Estrella aterrizada	23
2.5. Conexión Estrella - Delta	23
2.6. Conexión Estrella abierta - Delta abierta	24
2.7. Conexión Estrella aterrizada - Estrella aterrizada	24
2.8. Conexión Delta - Delta	26
2.9. Transformador monofásico con derivación central	26
2.10. Transformador con tres devanados en conexión en serie	27
2.11. Estrella - Delta con transformador con derivación central	27
2.12. Estrella abierta - Delta abierta con transformador con derivación central	28
2.13. Regulador de tensión por pasos Tipo B	29
3.1. Ubicación del “bus1” de la fuente de tensión en OpenDSS-G en vista general	44
3.2. Ubicación del “bus1” de la fuente de tensión en OpenDSS-G en vista ampliada	45
3.3. Ejemplo de código de librería de una línea	55

3.4. Resultado del programa para generar LineCode	56
3.5. Anillos generados por las líneas eliminadas	58
3.6. Ejemplo de código de librería de un transformador	67
3.7. Botones para acceder a la herramienta “OpenDSS to OpenDSS-G Trans- lator”	86
3.8. Interfaz de la herramienta “OpenDSS to OpenDSS-G Translator” . . .	86
3.9. Visualización de la red de distribución en OpenDSS-G	87
3.10. “Tools Pallete” en modo simulación	87
4.1. Reporte de potencias en las terminales de los elementos de la red	91
4.2. Reporte de potencias en los nodos de las barras	92
4.3. Reportes de tensión línea - línea y línea - neutro en las barras	93
4.4. Reporte de tensión en los nodos de las barras	94
4.5. Reporte de tensión en los elementos de la red	95
4.6. Reporte de corriente en las terminales de los elementos de la red	96
4.7. Reporte de corriente en los nodos de las barras	97
4.8. Reporte de pérdidas en las líneas y transformadores	98
4.9. Resumen de OpenDSS para los diferente perfiles de carga	99
4.10. Perfil de tensión con cargas constantes	100
4.11. Perfil de tensión con demanda baja (<i>LoadShape_1</i>)	101
4.12. Perfil de tensión con demanda media (<i>LoadShape_3</i>)	102
4.13. Perfil de tensión con demanda alta (<i>LoadShape_6</i>)	103
4.14. Tensión en la fuente de alimentación a lo largo de un día considerando una demanda media (<i>LoadShape_3</i>)	104
4.15. Corriente en la fuente de alimentación a lo largo de un día considerando una demanda media (<i>LoadShape_3</i>)	105
4.16. Potencia real en la fuente de alimentación a lo largo de un día conside- rando una demanda media (<i>LoadShape_3</i>)	106
4.17. Potencia reactiva en la fuente de alimentación a lo largo de un día con- siderando una demanda media (<i>LoadShape_3</i>)	107

4.18. Visualización de potencias de un elemento en OpenDSS	108
4.19. Mapa de sobrecargas de la red de distribución	109
4.20. Mapa de nivel de tensión de la red de distribución	109
4.21. Mapa de flujo de potencia de la red de distribución en un área cercana a la alimentación	110
4.22. Mapa de elementos aislados	111
4.23. Reporte de elementos aislados	111

Capítulo 1

Introducción

1.1. Reseña de la empresa

El Instituto Costarricense de Electricidad (ICE) es la empresa estatal que brinda servicios de electricidad en Costa Rica, además de ofrecer servicios de telecomunicaciones. Actualmente, es parte del Grupo ICE, junto con la Compañía Nacional de Fuerza y Luz (CNFL) y Radiográfica Costarricense y Grupo Cobro (RACSA). El ICE y la CNFL son los encargados de generar casi el 78 % de la electricidad demandada en el país por medio 40 plantas de generación que operan por medio de fuentes renovables, tales como hidroeléctrica, solar, biomásica, geotérmica y eólica (Grupo ICE, 2022).

El ICE nace el 8 de abril de 1949, bajo el Decreto de Ley No. 449, con el fin de solucionar la escasez de electricidad que existía en el momento, así como desarrollar las fuentes de energía del país. Con ello, se pudo brindar acceso a la electricidad a la población en general y no solo a un grupo selecto de personas, con lo cual, el ICE pasar de un acceso a la energía eléctrica de un 14 % en 1949 a un 99,7 %. El éxito de la institución fue tal que el Estado le concedió el manejo y desarrollo de las telecomunicaciones del país y en 2008, por medio de la Ley No. 8660, pasa a conformarse el Grupo ICE, integrado por el ICE, la CNFL y RACSA (Grupo ICE, 2021).

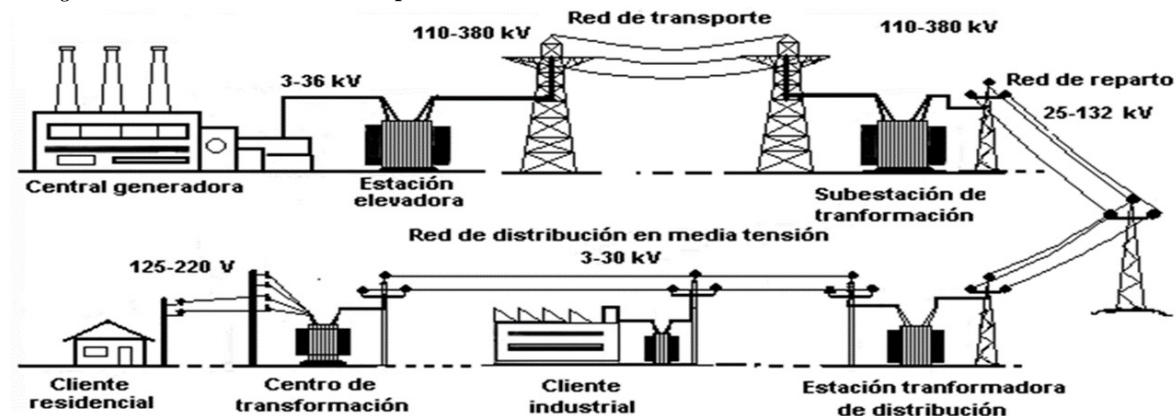
La misión del Grupo ICE es “brindar energía, conectividad y servicios digitales, seguros y sostenibles a los habitantes de Costa Rica”; por su parte, la visión es la siguiente: “El Grupo ICE liderará la electrificación renovable de la economía y proveerá

al país de un ecosistema seguro de telecomunicaciones digitales de última generación” respectivamente (Grupo ICE, 2023).

Respecto al proceso productivo, este se puede observar en la Figura 1.1. Este inicia con la generación eléctrica por medio de diferentes fuentes; en el caso de Costa Rica, la hídrica, geotérmica, eólica, biomasa, solar y combustibles fósiles (Instituto Costarricense de Electricidad, 2015), las plantas son manejadas por el ICE, las distribuidoras y las empresas privadas. Posteriormente, la energía generada es transportada por líneas de alta tensión, que permiten el transporte de la energía por largas distancias sin pérdidas considerables, hasta llegar a las subestaciones. El ICE es el encargado de la transmisión a nivel nacional. Luego, desde las subestaciones, la electricidad es distribuida por líneas de media tensión hasta llegar a un transformador que reduce la tensión para el uso de los usuarios, pasando primero por un medidor que registra la energía consumida por el usuario (Kersting & Kerestes, 2023). Las distribuidoras se encargan de esta labor.

Figura 1.1

Diagrama de un sistema de potencia



Nota. Los valores de tensión presentados en la figura pueden no corresponder a los empleados en Costa Rica. Tomado de “Análisis de sobretensiones transitorias en líneas de transmisión a través del método J Martí”, por Morales de la Cruz, 2018.

1.2. Descripción del problema

El 7 de diciembre de 2021, se firmó la Ley No. 10086, Promoción y Regulación de los Recursos Energéticos Distribuidos a partir de Fuentes Renovables (Presidencia de

la Republica de Costa Rica, 2021), la cual entró en vigencia el 8 de enero de 2022. Esta ley regula e incentiva la participación de los usuarios comerciales y residenciales en la generación eléctrica por medio de fuentes renovables, de forma que el excedente que produzca el usuario sea insertado en la red de distribución, recibiendo en beneficio créditos en sus futuras facturas eléctricas.

Lo anterior conlleva un reto para las empresas distribuidoras, ya que la implementación de fuentes renovables en la red de distribución puede impactar la calidad de la tensión que entrega el sistema y con las posibles consecuencias tanto para la red nacional como para los usuarios. Por otra parte, la Ley No. 10086 (Asamblea Legislativa, 2021), exige a las empresas distribuidoras mantener actualizada la capacidad de penetración segura de generación que utiliza fuentes renovables en el Sistema Eléctrico Nacional, además, esta información debe implementarse en un *software* con un Sistema de Información Geográfica (GIS), para la posterior generación y publicación de los mapas de capacidad de penetración (Imprenta Nacional, 2023).

En el caso del ICE, se cuenta con la información del GIS, pero esta no se encuentra implementada en un *software* para su respectiva manipulación y procesamiento según lo requerido, por lo que surge la necesidad de importar estos datos GIS a un *software* de simulación.

1.3. Objetivos de la investigación

A continuación, se detallan los objetivos general y específicos de la investigación del trabajo final de graduación para licenciatura:

1.3.1. Objetivo general

Modelar un circuito de la red distribución eléctrica del ICE utilizando *software* de simulación, incluyendo líneas de baja tensión, líneas de media tensión, cargas y transformadores dentro de un sistema de información geográfica (GIS) implementado en el *software* de simulación mediante Python, generando información sobre el estado actual de la red, así como ante posibles cambios para una mejor toma de decisiones.

1.3.2. Objetivos específicos

- a. Realizar una minería de datos en los archivos con la información del circuito brindados por el ICE, con el fin de que se garantice la calidad de los datos para su posterior uso y manipulación.
- b. Programar en Python un *software* que convierta los archivos con formato en Excel a archivos con formato compatible con el *software* de simulación, para su implementación en análisis de la red de distribución.
- c. Ingresar los archivos generados con Python en el *software* de simulación, de forma que el circuito de la red distribución quede cargado para su posterior uso en análisis del sistema.
- d. Generar análisis eléctricos de la red de distribución del ICE en el *software* de simulación, obteniendo así datos e información relevante para la toma de decisiones respecto al Sistema Eléctrico Nacional.

1.4. Justificación

La entrada en vigencia de la Ley No. 10086 obliga a contar con la información de la red de distribución eléctrica actualizada con información geográfica, de forma que la implementación de la generación distribuida sea lo más óptima posible, sin impactar negativamente el estado actual de la red y permitiendo un incremento de la potencia que proviene de fuentes renovables y en beneficio del usuario.

Además, al contar con un GIS, es posible obtener modelos de simulación más precisos, con los cuales se puede llevar una planificación más eficiente del crecimiento de la infraestructura de la red, una mejor detección de puntos débiles del sistema y fallos en este. Todo esto impacta positivamente en la calidad y confiabilidad del suministro eléctrico y repercute en beneficios económicos.

Por otra parte, una buena implementación de sistemas de simulación con GIS puede determinar efectivamente la cantidad de generación distribuida que se puede implemen-

tar, lo que permite el ingreso de potencia generada con fuentes renovables que puede sustituir energía generada con fuentes contaminantes.

Respecto a la satisfacción de los clientes, la implementación de modelos con sistemas de información geográfica permite determinar con rapidez si el usuario es capaz de integrarse al sistema de generación distribuida, lo cual agiliza el proceso y ahorra tiempos tanto a la distribuidora como al cliente. Asimismo, en caso de ingresar al modelo de generación distribuida, esto repercute en un beneficio económico para el usuario, impacta en su satisfacción y resalta el hecho de contar con un sistema preciso, con el cual determinar la capacidad de penetración de la generación distribuida.

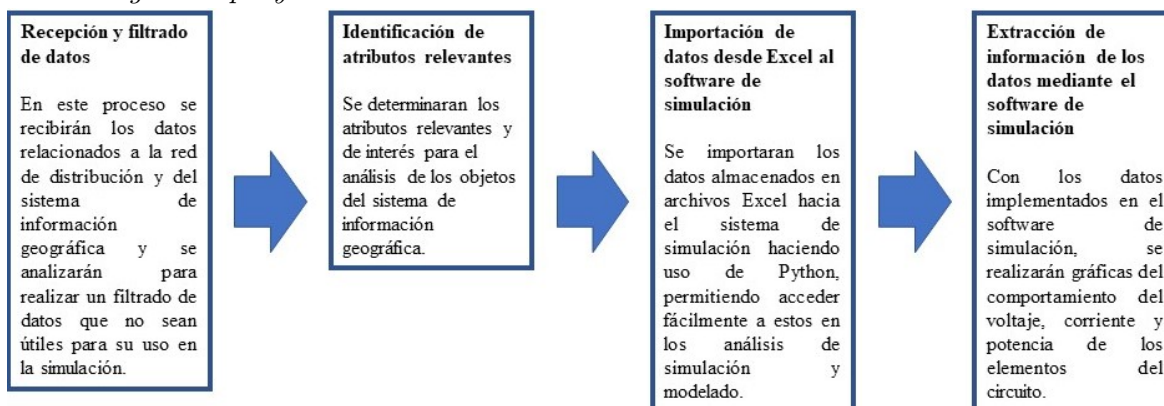
Finalmente, es importante señalar que el ICE no cuenta con un GIS implementado en su sistema de simulación de la red de distribución, por lo que no se cumple con lo que solicita la regulación además de afectar la eficiencia de la organización.

1.5. Viabilidad

Para la realización de este proyecto, se cuenta con la información del GIS, así como de la red de distribución suministrada por el ICE, con el *software* OpenDSS y OpenDSS-G para lo referente a la simulación de la red distribución y al que se le integrará la información del GIS y, por último, con el lenguaje de programación Python para realizar el código con el cual trabajar los archivos GIS y lograr su implementación en el *software* de simulación.

1.6. Metodología

La realización del proyecto está compuesto por diferentes fases, las cuales incluyen recepción y limpieza de datos, exportación de los datos a distintos formatos, entrada de los datos en el programa de simulación y extracción de información de los datos por medio de las herramientas brindadas por el *software* de simulación. En la Figura 1.2 se muestra el procedimiento a seguir.

Figura 1.2*Metodología del proyecto***1.6.1. Alcance de la investigación**

Con la implementación de este proyecto en el ICE, se contará con un sistema de modelado de la red de distribución con datos GIS, con ello, se puede verificar de forma rápida y precisa la capacidad de penetración de la generación distribuida. Además de que contar con información relacionada con GIS permite una mejor planificación de la ampliación y mejora de la red de distribución, lo que, a su vez, aumenta la productividad y eficiencia de esta.

Por otra parte, el análisis de calidad de los datos y su filtrado para eliminar datos innecesarios en los análisis permitirá mejores resultados a la hora de realizar simulaciones con estos datos. Además, al analizar los datos, se determinarán las áreas en las que es posible mejorar la calidad de estos, con lo que se deja abierta la posibilidad de implementar estrategias que permitan una recolección de datos con mayor calidad y optimizar el sistema de simulación.

1.6.2. Limitaciones

La principal limitación con la que cuenta este proyecto es referente al *software* de simulación y su capacidad de manejar datos GIS, así como los análisis y simulaciones que pueda realizar. Además, de que los *softwares* de simulación cuentan con un tamaño límite del sistema que se puede modelar, lo que puede limitar la dimensión del sistema de distribución a analizar. Por último, está la situación de que el tamaño de la red

a analizar sea considerable y demande demasiada capacidad de procesamiento, lo que puede causar que los equipos a utilizar no sean capaces de realizar los análisis.

1.6.3. Cronograma

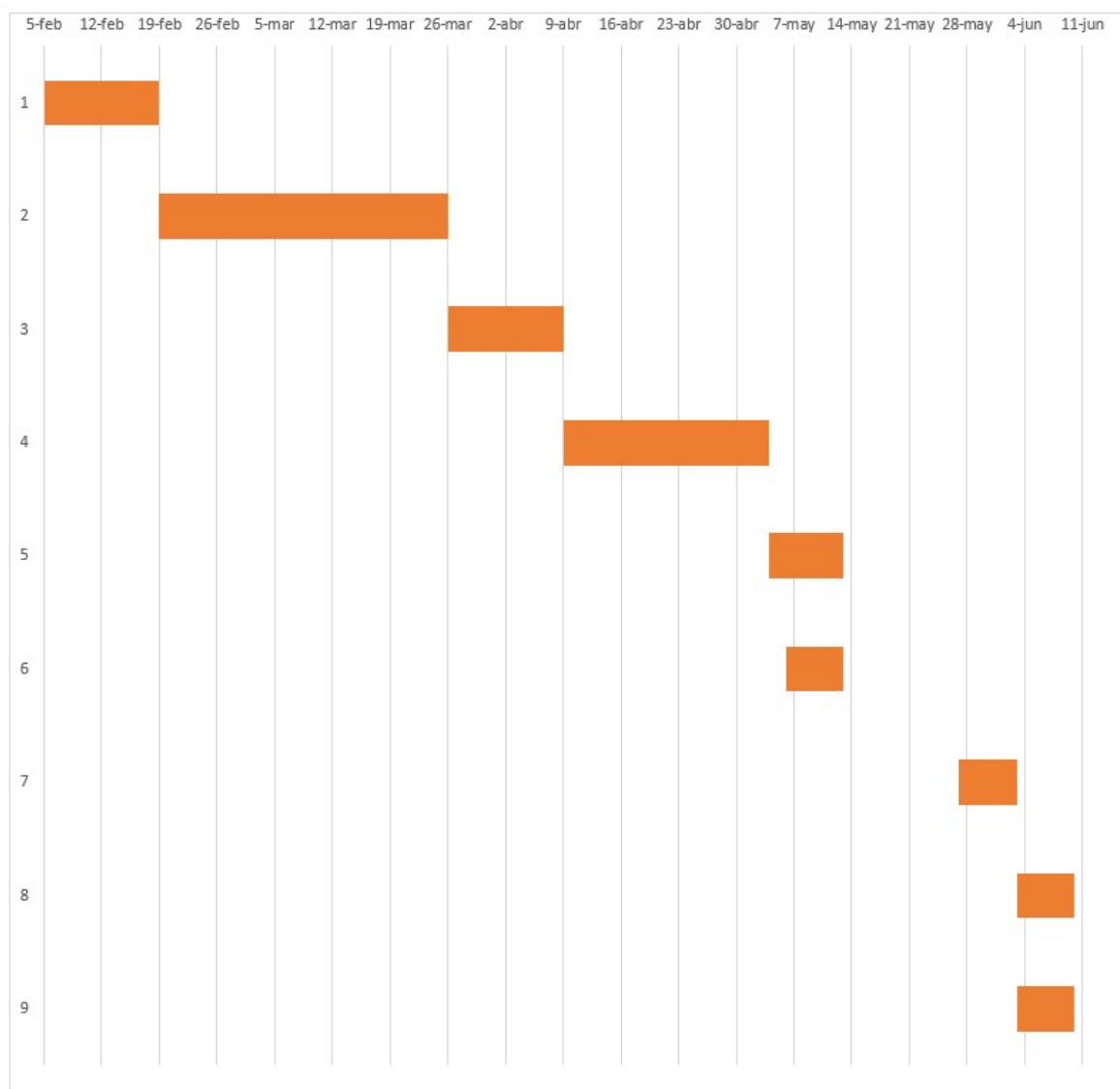
En la Tabla 1.1 se observan las actividades para la realización del proyecto en conjunto con las fechas proyectadas para su realización. Además, en la Figura 1.3 se muestra el diagrama de Gantt que representa de forma gráfica la información de la tabla anterior.

Tabla 1.1*Cronograma proyectado de trabajo*

	Actividad	Inicio	Duración (Días)	Fin
1	Realizar minería de datos en la información suministrada para asegurar una correcta implementación en la simulación	5-feb	14	19-feb
2	Programar en Python un software que convierta la información brindada en Excel en archivos .dss compatibles con el software de simulación	19-feb	35	26-mar
3	Generar archivos maestros con los cuales unificar los documentos con la información de la red e introducir comandos relacionados a la simulación	26-mar	14	9-abr
4	Implementar la red de distribución eléctrica del ICE dentro de un entorno GIS en el software de simulación	9-abr	25	4-may
5	Generar análisis con los cuales extraer información de la red de distribución en forma de reportes, gráficos y mapas de calor	4-may	9	13-may
6	Entrega de borrador del documento escrito a profesor guía y a profesores lectores	6-may	7	13-may
7	Estudiante somete a revisión del filólogo el informe final revisado por el profesor guía y los profesores lectores	27-may	7	3-jun
8	Estudiante entrega al representante de la empresa el documento con el proyecto final y solicita comprobante de recibido	3-jun	7	10-jun
9	Estudiante entrega a coordinador de trabajo final de graduación y a la biblioteca el proyecto final, carta de entendimiento de la biblioteca, carta de recibido de la empresa y la carta del filólogo.	3-jun	7	10-jun

Figura 1.3

Diagrama de Gantt del cronograma proyectado de trabajo



Capítulo 2

Marco teórico

2.1. Sistemas eléctricos de potencia

Un sistema eléctrico de potencia es el encargado de suministrar la energía eléctrica en el momento que se requiera y al nivel de tensión y frecuencia necesarios, esto mediante la generación de la energía en centrales generadoras, estas son transmitidas por líneas de transmisión hasta llegar a subestaciones, donde, posteriormente, la energía es transportada por líneas de distribución hasta llegar al usuario. Con ello, se garantiza el suministro de energía con ciertos niveles de calidad, confiabilidad y seguridad (Quintero et al., 2019). Los sistemas eléctricos de potencia están compuestos de tres grandes secciones, las cuales son la generación, la transmisión y la distribución.

Para lograr un funcionamiento adecuado de estos sistemas, con continuidad de en el servicio, a una buena regulación de tensión y con un buen control de la frecuencia, se utilizan los indicadores de confiabilidad entre los que se incluyen las tasas de salidas forzadas por mantenimiento correctivo en periodos de verano e invierno, la tasa de salidas forzadas por mantenimiento anual y las tasas de tiempos de reparación por salidas forzadas en verano, invierno y anual. Estos indicadores se pueden medir y controlar para cada elemento, aunque principalmente se deben medir en elementos críticos del sistema. Finalmente, la importancia de los indicadores de confiabilidad es que permiten tomar decisiones referentes a la operación, mantenimiento y crecimiento de los sistemas de potencia, con base en datos cuantificables, mejorando así la eficiencia

de estos sistemas (Gómez-Ramírez, 2016).

Referente al Sistema Eléctrico Nacional, en el 2023 se tuvo una demanda de 12 291,12 GWh con una producción de 11 939,72 GWh de los cuales se utilizaron 8701,38 GWh para satisfacer la demanda nacional (correspondiente a un 71 %). El restante de energía demandada fue importada del Mercado Eléctrico Regional; del mismo modo, el excedente producido fue exportado dentro del mismo mercado (Madrigal, 2024). Además, de acuerdo con Gómez-Ramírez, Meza et al. (2023), el Sistema Eléctrico de Costa Rica brindaba acceso a la electricidad a un 99,4% de la población, con una capacidad instalada de 3566 MW, con una demanda máxima de 1715 MW y unas pérdidas de transmisión del 11,6 %.

2.1.1. Generación

En la generación se transforma la energía proveniente de diferentes fuentes en energía eléctrica para su posterior aprovechamiento. Entre las fuentes existen no renovables, como combustibles fósiles o energía nuclear, así como fuentes renovables, tales como solar, hidroeléctrica, eólica, geotérmica, mareomotriz, entre otras (Carta González et al., 2009). Por último, en la generación es requerido el uso de sistemas mecánicos, como turbinas y rotores, que permitan la conversión de la energía proveniente de las fuentes anteriormente mencionadas en energía eléctrica.

En el 2023, en Costa Rica, se tuvo una generación de energía eléctrica de 11 939,72 GWh, de los cuales 69,74% corresponden a energía hidroeléctrica, 12,39% a energía geotérmica, 12,24% a energía eólica, 5,1% proveniente de fuentes no renovables y un 0,53% a energía solar y biomasa (Madrigal, 2024). Como se observa, Costa Rica cuenta con una generación eléctrica proveniente en su gran mayoría de fuentes renovables y amigables con el ambiente. Por otra parte, en el 2020, se contaban con 444 unidades generadoras instaladas para la producción de electricidad, además de que el ICE tenía una participación del 74,13% de la potencia instalada del Sistema Eléctrico Nacional, siendo el principal generador de energía eléctrica del país (Instituto Costarricense de Electricidad, 2020).

2.1.2. Transmisión

La transmisión en sistemas eléctricos de potencia consiste en el transporte de la energía eléctrica desde las plantas de generación hasta las subestaciones a altas tensiones haciendo uso de líneas de transmisión y transformadores (Llano & Moreno, 2014). A nivel nacional, la Autoridad Reguladora de los Servicios Públicos (2015) menciona que la alta tensión se considera como el “nivel de tensión igual o superior a 100 kV e igual o menor de 230 kV”. En el caso del Instituto Costarricense de Electricidad (2021), 138 y 230 kV son las tensiones empleadas en transmisión. Cabe aclarar que en otros países y regiones los niveles de alta tensión utilizados pueden variar, como en el caso de Llano y Moreno (2014) que considera como alta tensión 115, 230, 500 y 750 kV. Por otra parte, según Gómez-Ramírez, Meza et al. (2023), Costa Rica cuenta con una capacidad de transformación de potencia de 4068 MVA en transformadores de dos devanados y de 7305 MVA en transformadores de tres devanados, además de contar con 80 MVAR de compensación inductiva y 267 MVAR de compensación reactiva.

También se debe mencionar que en Costa Rica el único agente encargado de la transmisión del Sistema Eléctrico Nacional es el ICE, y este en el 2021 contaba con un total de 2986 km de líneas de transmisión con un total de 2378 km de líneas correspondientes a una tensión de 230 kV y 608 km de líneas a una tensión de 138 kV (Instituto Costarricense de Electricidad, 2021), en la Figura 2.1 se aprecia el crecimiento de las líneas de transmisión desde el 2006 al 2021 en Costa Rica.

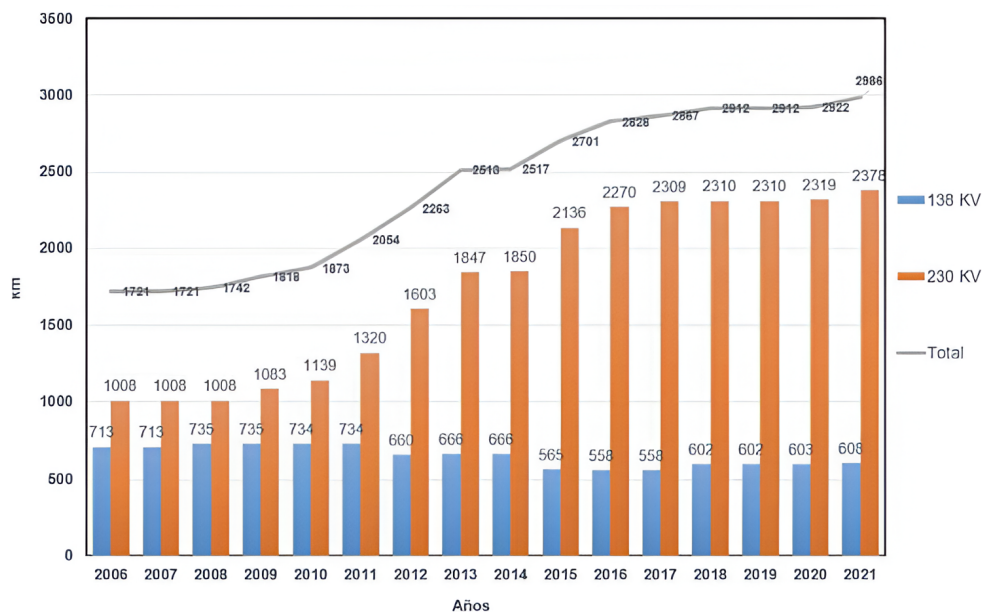
2.1.3. Distribución

La distribución es la parte del sistema eléctrico de potencia encargada de transportar la electricidad desde las subestaciones -donde se convierte de alta tensión a media tensión mediante transformadores-, hasta los usuarios finales a través de líneas de distribución. Posteriormente, se realiza una nueva conversión con transformadores para adaptar la tensión a los niveles requeridos por los usuarios (Kersting & Kerestes, 2023).

Respecto a las subestaciones, están compuestas por interruptores que permiten desconectarse tanto de la línea de transmisión como de los circuitos de distribución

Figura 2.1

Evolución anual de las líneas de transmisión de Costa Rica, periodo 2006 – 2021

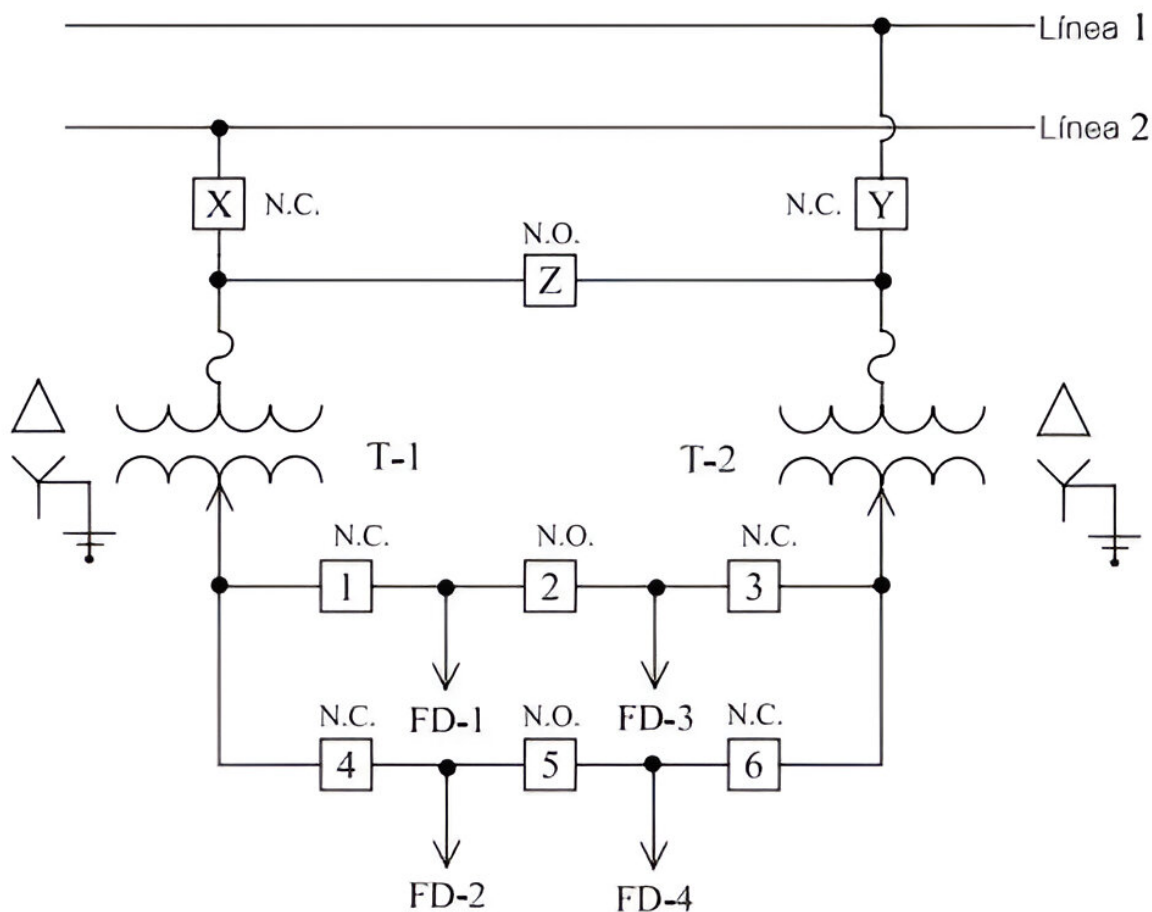


Nota. Tomado de “Plan de Expansión de la Transmisión 2021-2031”, por Instituto Costarricense de Electricidad, 2021.

en caso de un evento, un sistema de transformadores encargados de reducir la alta tensión de transferencia a un nivel de tensión utilizado en distribución, elementos de regulación de la tensión que permita ajustar este valor acorde a la carga evitando así caídas significativas de la tensión, protecciones en contra de fallas o eventos como lo son cortocircuitos y, finalmente, equipos de medición que registran los valores mínimos, máximos y promedios de corriente, tensión, potencia, factor de potencia, entre otros en un periodo de tiempo determinado, con el fin de monitorear el comportamiento de la subestación (Kersting & Kerestes, 2023). Respecto a las subestaciones a nivel nacional, la Autoridad Reguladora de los Servicios Públicos (2019) menciona que en 2019 había 82 de estas instalaciones; de las cuales, 62 eran operadas por el ICE. En la Figura 2.2 se observa el diagrama de una subestación con dos transformadores, en este caso cada uno de los dos transformadores está dimensionado de forma que, en caso de que uno de los dos transformadores sea desconectado, el otro pueda seguir suministrando energía a todos los circuitos de distribución.

Figura 2.2

Diagrama de una subestación con dos transformadores



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Referente a la topología de las redes de distribución, en la mayoría de casos suelen ser radiales, lo que significa que solo hay un camino para el flujo de potencia desde la subestación hasta la carga, aunque existen sistemas de distribución, los cuales cuentan con más de un camino en forma de anillos, principalmente en zonas metropolitanas (Kersting & Kerestes, 2023). El sistema radial cuenta con la ventaja de que se reducen costos, al no requerir de interruptores, protecciones especiales y conductores extras para controlar el flujo de potencia, pero en caso de un fallo en la línea, todo lo que esté aguas abajo del lugar de la falla queda desconectado hasta la resolución del problema. En el caso del sistema de anillos, aunque aumenta los costos, permite una

mejor continuidad del servicio, ya que, en caso de falla, se desconecta la zona afectada y, por medio de interruptores, se activa otro camino hacia las zonas que estaban aguas abajo del lugar de la falla, lo que permite el suministro de potencia, aun cuando se está reparando la falla (Hashemi-Dezaki et al., 2015).

Continuando con las redes de distribución radiales, estos son sistemas no balanceados, debido al gran número de cargas monofásicas desiguales que deben ser alimentadas, además de que los conductores no tienen espaciado equilateral en configuraciones trifásicas, lo que aumenta aun más el desbalance de las fases de la red. Debido a lo anterior, las técnicas de análisis empleadas en los sistemas de transferencia no son adecuadas para las redes de distribución y se deben emplear otros métodos para este caso (Kersting & Kerestes, 2023).

Por otra parte, la Autoridad Reguladora de los Servicios Públicos (2015) en “Supervisión de la comercialización del suministro eléctrico en baja y media tensión (norma técnica regulatoria AR-NT-SUCOM)” menciona que media tensión corresponde a un valor “mayor a 1 kV pero menor a 100 kV” y que baja tensión a un “nivel de tensión igual o menor de 1 kV”, indicando que los magnitudes de baja tensión a utilizar son 120, 208, 240, 277 y 480 V; por otra parte, el Instituto Costarricense de Electricidad (2021) indica que las magnitudes empleados en media tensión corresponden a 69; 34,5; 29,9 y 13,8 kV.

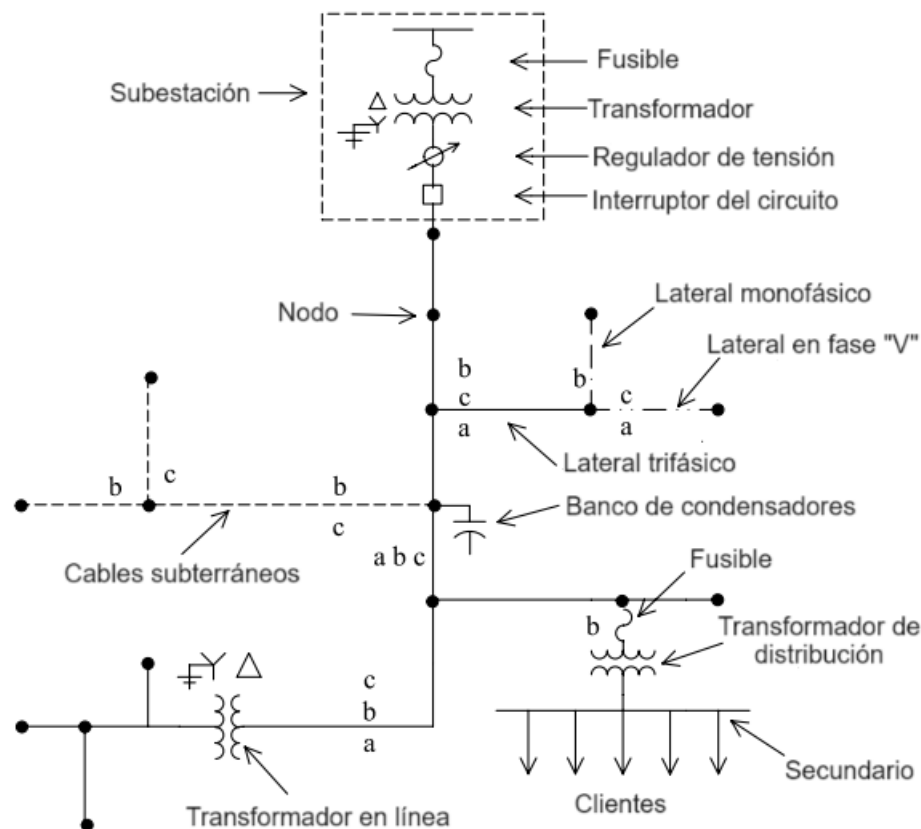
Respecto a la generación distribuida, se refiere al conjunto de tecnologías y equipos utilizados para la generación de energía eléctrica a partir de fuentes de energía renovables, empleados por una persona física o jurídica para autoconsumo y con entrega de excedentes a la red (Asamblea Legislativa, 2021). La generación distribuida se caracteriza por el motivo de que su punto de conexión se encuentra cerca de las cargas, lo que genera que los flujos de potencia puedan circular en ambos sentidos, contrario al diseño radial típico de las redes de distribución con una sola dirección de flujo, impactando las condiciones operativas del sistema. Por otra parte, la presencia de varios generadores distribuidos pequeños conlleva una alteración de las corrientes de cortocircuito, lo cual puede provocar una pérdida de coordinación de las protecciones e impactar en la seguridad y la confiabilidad de la red (Tuta & Hincapié, 2011).

2.2. Elementos de una red de distribución

En la Figura 2.3 se aprecia un diagrama de un alimentador radial perteneciente a una red de distribución, en este se pueden observar diferentes elementos que componen las redes de distribución, como son las barras, líneas de media y baja tensión, protecciones de la red, interruptores, transformadores, reguladores de tensión, bancos de condensadores y, finalmente, las cargas, presentadas como los clientes en la figura. También se indican las fases de cada línea donde se observan líneas monofásicas, bifásicas y trifásicas.

Figura 2.3

Diagrama de una sola línea de un alimentador de distribución trifásico



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

2.2.1. Barras y líneas

Las barras son los componentes que utilizan el resto de elementos de la red para interconectarse entre sí. Estos suelen tener desde 1 hasta N cantidad de nodos correspondientes a las fases de los elementos conectados (Leal Parra, 2016).

En el caso de las líneas, son los elementos que transmiten la energía desde un punto inicial hacia otro (Leal Parra, 2016). Estas pueden ser tanto aéreas como subterráneas y estar compuestas de 1 o más conductores. Para su manejo en análisis de redes de distribución en ciertos casos, se asume que las líneas son balanceadas y transpuestas, de forma que se simplifique el proceso de análisis y se pueda obtener un resultado aproximado (Kersting & Kerestes, 2023). Considerando una línea trifásica balanceada transpuesta, su impedancia está dada por la siguiente ecuación:

$$z_{linea} = r + j0,12134ln \left(\frac{D_{eq}}{GMR} \right) (\Omega/milla) \quad (2.1)$$

Donde r corresponde a la resistencia del conductor, D_{eq} a la distancia equivalente de los conductores como se muestra en la ecuación 2.2 y GMR al radio medio geométrico. En el caso de la resistencia del conductor y el radio medio geométrico, se suelen obtener de tablas de conductores.

$$D_{eq} = \sqrt[3]{D_{ab} \cdot D_{bc} \cdot D_{ca}} (ft) \quad (2.2)$$

Donde D_{ab} corresponde a la distancia entre los conductores a y b, D_{bc} a la distancia entre los conductores b y c, y D_{ca} a la distancia entre los conductores a y c.

Sin embargo, debido a que las redes de distribución son sistemas no balanceados que están compuestos de líneas trifásicas, bifásicas y monofásicas no transpuestas, se deben considerar tanto las impedancias propias como mutuas de los conductores para obtener un resultado más próximo al comportamiento de la red. Debido a lo anterior, en 1926, John Carson desarrolló una técnica con la cual obtener las impedancias propias y mutuas para un conjunto de N conductores, utilizando dos ecuaciones (Kersting & Kerestes, 2023). Posteriormente, estas ecuaciones fueron simplificadas mediante suposiciones, de forma que se obtienen las ecuaciones de Carson modificadas:

$$\hat{Z}_{ii} = r + 0,09530 + j0,12134 \left(\ln \frac{1}{GMR_i} + 7,93402 \right) (\Omega/milla) \quad (2.3)$$

$$\hat{Z}_{ij} = 0,09530 + j0,12134 \left(\ln \frac{1}{D_{ij}} \right) (\Omega/milla) \quad (2.4)$$

Haciendo uso de estas ecuaciones, es posible generar una matriz conocida como matriz primitiva de impedancias, de un tamaño de N por N, siendo N el número de conductores (Kersting & Kerestes, 2023). Por ejemplo, para una línea compuesta de tres conductores con neutro concéntrico, su matriz primitiva corresponde a:

$$\left[\hat{Z}_{primitiva} \right] = \left[\begin{array}{ccc|ccc} \hat{Z}_{aa} & \hat{Z}_{ab} & \hat{Z}_{ac} & | & \hat{Z}_{an1} & \hat{Z}_{an2} & \hat{Z}_{an3} \\ \hat{Z}_{ba} & \hat{Z}_{bb} & \hat{Z}_{bc} & | & \hat{Z}_{bn1} & \hat{Z}_{bn2} & \hat{Z}_{bn3} \\ \hat{Z}_{ca} & \hat{Z}_{cb} & \hat{Z}_{cc} & | & \hat{Z}_{cn1} & \hat{Z}_{cn2} & \hat{Z}_{cn3} \\ \hline \hat{Z}_{n1a} & \hat{Z}_{n1b} & \hat{Z}_{n1c} & | & \hat{Z}_{n1n1} & \hat{Z}_{n1n2} & \hat{Z}_{n1n3} \\ \hat{Z}_{n2a} & \hat{Z}_{n2b} & \hat{Z}_{n2c} & | & \hat{Z}_{n2n1} & \hat{Z}_{n2n2} & \hat{Z}_{n2n3} \\ \hat{Z}_{n3a} & \hat{Z}_{n3b} & \hat{Z}_{n3c} & | & \hat{Z}_{n3n1} & \hat{Z}_{n3n2} & \hat{Z}_{n3n3} \end{array} \right] \quad (2.5)$$

$$\left[\hat{Z}_{primitiva} \right] = \left[\begin{array}{cc} \hat{Z}_{ij} & \hat{Z}_{in} \\ \hat{Z}_{nj} & \hat{Z}_{nn} \end{array} \right] \quad (2.6)$$

Luego de obtener la matriz primitiva, esta debe ser reducida a una matriz de 3 por 3, a excepción de líneas trifásicas de tres conductores (Kersting & Kerestes, 2023). Esto se puede realizar por medio del método de Kron, el cual consiste en aplicar la siguiente ecuación:

$$\left[\hat{Z}_{abc} \right] = \left[\hat{Z}_{ij} \right] - \left[\hat{Z}_{in} \right] \cdot \left[\hat{Z}_{nn} \right]^{-1} \cdot \left[\hat{Z}_{nj} \right] \quad (2.7)$$

Obteniendo así que en la matriz solo queden presentes las impedancias propias y equivalentes de las tres fases, mientras que las impedancias de los neutros se han integrado dentro de las impedancias de las fases (Kersting & Kerestes, 2023). Pos-

teriormente, esta matriz de impedancias es utilizada para el análisis de las redes de distribución.

2.2.2. Interruptores y protecciones

Los interruptores son los dispositivos encargados de controlar el flujo de corriente, a través de la interrupción o restablecimiento de esta. Este cambio se puede realizar con el fin de despejar fallas, por razones de mantenimiento o de conexión entre los distintos elementos de la red. Estos se suelen clasificar por el medio que utilizan para apagar el arco eléctrico que se forma entre los contactores, al realizar un cambio de estado del interruptor, entre estos medios se encuentran: aire, aire comprimido, aceite y al vacío (Giménez Gutiérrez, 2000).

Las protecciones son todo aquel elemento utilizado para evitar o minimizar el daño a los equipos y personas, producto de un fallo o condición anómala de la red de distribución, además, con las protecciones se busca conseguir la más alta confiabilidad, seguridad y resiliencia del sistema, con lo cual se minimizan las interrupciones del suministro eléctrico y la cantidad de usuarios afectadas por estas. Para lograr lo anterior, se deben proveer las protecciones adecuadas a los diferentes elementos que componen la red de distribución (Zhaoyu, s.f.). Según Zhaoyu (s.f.) en “Distribution System Protection”, las protecciones de una red de distribución se pueden clasificar en:

- **Fusibles de acción única:** Estos son utilizados para abrir partes de la red en las cuales este pasando una corriente de gran magnitud. También pueden ser utilizados para crear zonas dentro de la red de distribución. Son económicos, pero conllevan un reto a nivel operacional al tener que ser reemplazados después de su uso.
- **Reconectores de circuito automático:** Son equipos automáticos con la capacidad de detectar sobrecorrientes e interrumpir el flujo de la corriente y con la capacidad de reconectar y reenergizar el circuito otra vez, cuando la corriente vuelve a valores nominales.

- **Seccionador:** Son equipos que suelen trabajar en conjunto con un reconector o un disyuntor, con la capacidad de contar la cantidad de veces que el reconector o disyuntor interrumpe el flujo de corriente, después de alcanzado un valor de interrupciones continuas el seccionador abre el circuito y aísla la sección con la falla.
- **Disyuntor:** Son equipos utilizados para la protección contra sobrecorrientes. Debido a su costo y a su tamaño voluminoso, suelen ser empleados para la protección de los alimentadores conectados a la subestación.
- **Relés de sobrecorriente de tiempo:** Son equipos electromecánicos con la capacidad de detectar condiciones anormales de la red de distribución, pudiendo así enviar una señal a los disyuntores de forma que estos disparen y cierren el circuito. Para el funcionamiento de estos equipos, se requieren de transformadores con los cuales reducir la tensión y la corriente a un valor de 120 V y 5 A, respectivamente. Son equipos confiables, pero no cuentan con la capacidad de realizar funciones complejas de protección.
- **Relés estáticos o de estado sólido:** A diferencia de los relés electromecánicos, no tiene componentes móviles, de forma que cuentan con tiempos de respuesta más rápidos y precisos, pero requieren de una fuente de corriente directa.
- **Relés digitales:** Estos equipos son capaces de realizar funciones de protección complejas así como de control, además de poder efectuar autoanálisis y notificar a los operarios sobre fallos. Debido a sus funciones, son capaces de simplificar las labores de operación y mantenimiento de las subestaciones. Requieren de una alimentación de corriente directa para su operación.
- **Interruptor de ruptura de carga:** Son equipos utilizados para desconectar diferentes elementos o secciones de la red de distribución, siendo una forma confiable y segura de aislamiento y control de la red.
- **Interruptor del circuito:** Equipos diseñados para abrir y cerrar un circuito de

forma manual y para abrir el circuito cuando se alcanza un valor determinado de sobrecorriente.

Por otra parte, se debe realizar una correcta coordinación de protecciones, de forma que se pueda minimizar el impacto de las fallas en los usuarios, así como optimizar el proceso de restauración del circuito (Zhaoyu, s.f.). Además, se debe considerar el impacto de redes de distribución con generación distribuida en la coordinación de protecciones, ya que en este caso se presenta un incremento de las corrientes de cortocircuito, provocando que las protecciones de la red pierdan selectividad y velocidad de disparo, requiriendo así emplear una metodología diferente a la tradicional en la coordinación de las protecciones (Tuta & Hincapié, 2011).

2.2.3. Transformadores

Acorde a Chapman (2012), un transformador se define como “un dispositivo que cambia la potencia eléctrica alterna con un nivel de tensión a potencia eléctrica alterna con otro nivel de tensión mediante la acción de un campo magnético”. Está compuesto de dos o más bobinas, las cuales son enrollados alrededor de un núcleo ferromagnético.

Para el caso de redes de distribución, principalmente, se utilizan dos tipos de transformadores: transformadores trifásicos (ubicados en las subestaciones, los cuales se emplean para reducir la alta tensión de transmisión a media tensión de distribución) y los transformadores (que reducen la media tensión de distribución a baja tensión para uso de los usuarios) (Kersting & Kerestes, 2023). Al primer tipo de transformador se le conoce como transformador de subestación, mientras que al segundo se le llama transformador de distribución (Chapman, 2012).

Para el caso de transformadores de subestación, cuentan con diferentes conexiones, las cuales se eligen acorde a las características de las cargas. Estas conexiones se detallan a continuación:

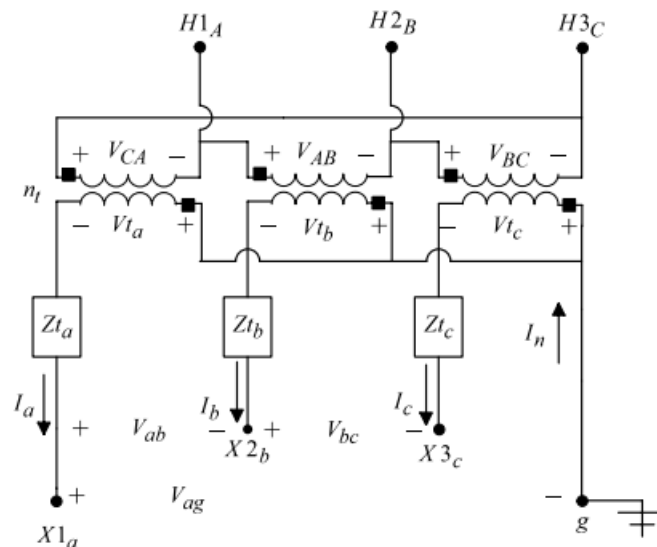
- **Delta - Estrella aterrizada:** Conexión utilizada en subestaciones en alimentadores de cuatro conductores. También son usados para alimentar cargas de una sola fase, esto debido a la conexión en estrella del secundario, ya que es posible

conectar tres circuitos monofásicos, lo cual resulta en que el banco de transformadores esté balanceado (Kersting & Kerestes, 2023). En la Figura 2.4 se observa este tipo de configuración.

- **Estrella - Delta:** Es usada típicamente para brindar servicio a una combinación de una carga monofásica de iluminación en conjunto con una carga trifásica de potencia. En caso de que una de las fases quede abierta, esta configuración puede seguir suministrando una fuente trifásica en forma de una Estrella abierta - Delta abierta, solo que los dos transformadores restantes pueden experimentar una sobrecarga, provocando que estos equipos se quemem (Kersting & Kerestes, 2023). En la Figura 2.5 se aprecia la configuración anterior.
- **Estrella abierta - Delta abierta:** Al igual que en la conexión anterior, se utiliza para alimentar a una combinación de una carga monofásica de iluminación en conjunto con una carga trifásica de potencia, pero en este caso la carga trifásica es pequeña, en comparación con la monofásica, lo que permite tener un sistema de dos transformadores que brinda tensiones trifásicas de línea a línea a una combinación de cargas (Kersting & Kerestes, 2023). Esta configuración se muestra en la Figura 2.6.
- **Estrella aterrizada - Estrella aterrizada:** Se utiliza principalmente para suministrar cargas monofásicas y trifásicas en sistemas de cuatro cables con conexión a tierra múltiple (Kersting & Kerestes, 2023). Esta conexión requiere de calibres de conductores mayores al presentar mayores corrientes, además, en caso de fallo de una fase, se generarán armónicos, lo cual evitaría que pueda alimentar cargas trifásicas (Leal Parra, 2016). En la Figura 2.7 se puede visualizar la configuración anteriormente mencionada.
- **Delta - Delta:** Empleadas principalmente para alimentar circuitos de tres conductores, proveyendo energía a combinaciones de cargas trifásicas y monofásicas (Kersting & Kerestes, 2023). En la Figura 2.8 se muestra esta configuración.

Figura 2.4

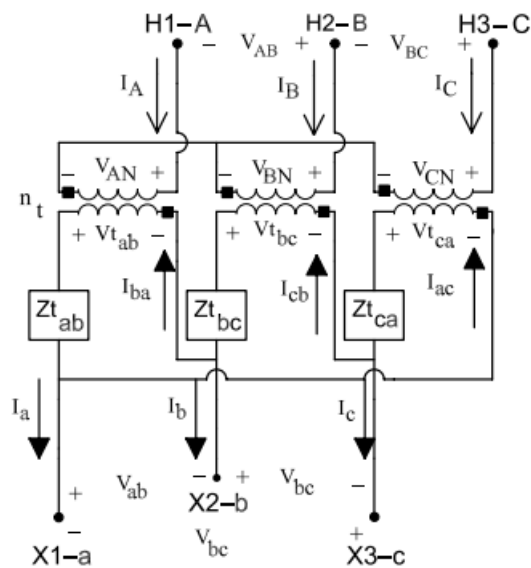
Conexión Delta - Estrella aterrizada



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.5

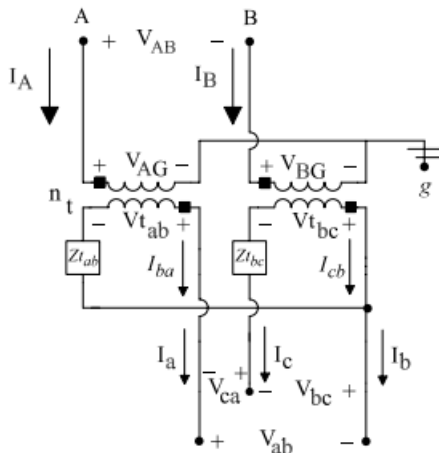
Conexión Estrella - Delta



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.6

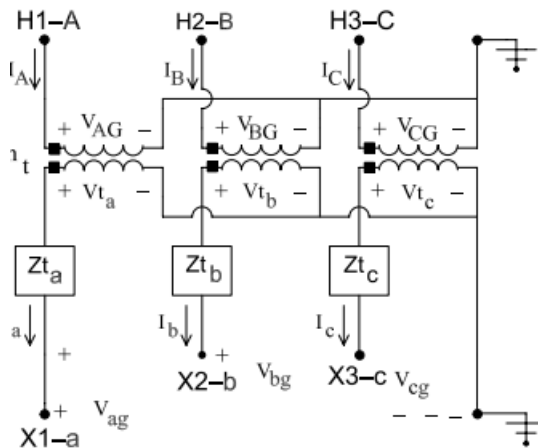
Conexión Estrella abierta - Delta abierta



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.7

Conexión Estrella aterrizada - Estrella aterrizada



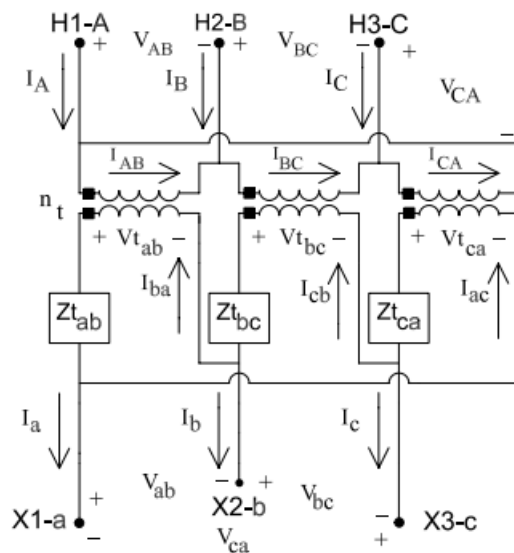
Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Respecto a los transformadores de distribución, también se hace uso de diferentes combinaciones acorde al tipo de carga que se vayan a conectar, las cuales se indican a continuación:

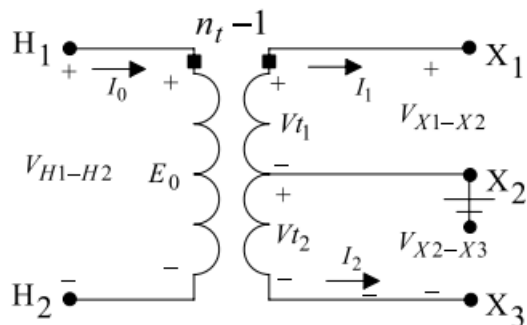
- **Transformador monofásico con derivación central:** Brindan una tensión de 120 entre la fase y la derivación, además de 240 V entre las dos fases del secundario (Kersting & Kerestes, 2023). En la Figura 2.9 se muestra un diagrama de este transformador.
- **Transformador de tres devanados:** Cuando los secundarios se conectan en serie, son capaces de proveer tensiones de 240 V y 120 V. Cuando es conectado en paralelo, se utiliza para alimentar 3 circuitos de 120 V y una tensión de trifásica de línea-línea de 208 V (Kersting & Kerestes, 2023). La conexión en serie se puede observar en la Figura 2.10.
- **Estrella - Delta con transformador con derivación central:** Popularmente utilizado para brindar servicio a una combinación de cargas trifásicas y monofásicas. Para alimentar las cargas monofásicas, uno de los transformadores tendrá derivación central (Kersting & Kerestes, 2023). En la Figura 2.11 se muestra esta configuración de transformadores.
- **Estrella abierta - Delta abierta con transformador con derivación central:** Utilizada para alimentar combinaciones de cargas monofásicas y trifásicas por medio de un transformador de potencia y un transformador con tres devanados, pero requiere que el primario esté aterrizado (Kersting & Kerestes, 2023). Una de los posibles formas de lograr esta configuración se muestra en la Figura 2.12

2.2.4. Reguladores de tensión

Debido a la variabilidad de las cargas en los sistemas de distribución, se requiere implementar medios para regular la tensión, de forma que el valor de tensión que recibe cada usuario se mantenga dentro de un rango aceptable (Kersting & Kerestes, 2023).

Figura 2.8*Conexión Delta - Delta*

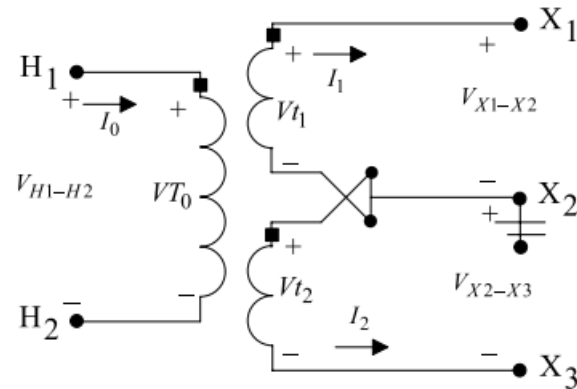
Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.9*Transformador monofásico con derivación central*

Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.10

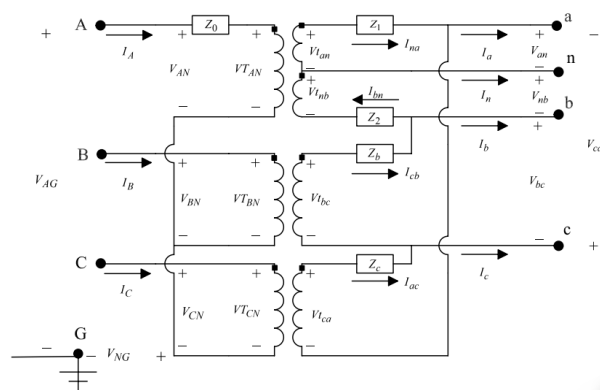
Transformador con tres devanados en conexión en serie



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.11

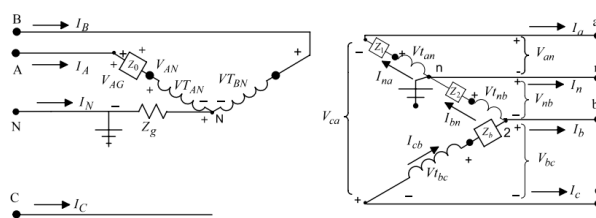
Estrella - Delta con transformador con derivación central



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Figura 2.12

Estrella abierta - Delta abierta con transformador con derivación central



Nota. Tomado de *Distribution System Modeling and Analysis with MATLAB® and WindMil®*, por Kersting y Kerestes, 2023.

Por otra parte, debido a las impedancias de las líneas, se genera una caída de tensión conforme las cargas se alejan de la fuente de alimentación, por lo que regular esta caída es importante para asegurar la confiabilidad de la red de distribución. Para regular la tensión, existen varios métodos entre los que se encuentran bancos de condensadores en derivación, transformadores de cambio de tomas (TAP) de cargas y reguladores de tensión de paso.

En el caso de reguladores de tensión de paso estos son equipos compuestos de un autotransformador y un mecanismo de cambio de tomas de cargas. De forma que en uno de los devanados donde se ubica el cambio de tomas, irá variando acorde con el cambio de la carga, logrando así una regulación de la tensión (Leal Parra, 2016). Reguladores de tensión estándar permiten un rango de regulación de $\pm 10\%$, usualmente en 32 pasos, correspondiente a un $5/8\%$ por paso. Estos equipos pueden ser conectados en “Tipo A” o “Tipo B”, de acuerdo con la norma ANSI/IEEE C57.15-1986 (Kersting & Kerestes, 2023), en la Figura 2.13 se muestra un regulador de tensión Tipo B. Para su correcto funcionamiento, se debe indicar al circuito de control la siguiente información:

- **Nivel de tensión:** Magnitud deseada en la carga.
- **Ancho de banda:** La variación de tensión permitida en la carga. De forma que la tensión en la carga será \pm la mitad del ancho de banda.
- **Tiempo de retraso:** Cantidad de tiempo en la cual un aumento o caída de la tensión no generará un cambio de los pasos. Esto para prevenir un cambio de

- **Demanda:** Carga promedio en un periodo de tiempo.
- **Demanda máxima:** La demanda más grande en un periodo de tiempo específico.
- **Demanda promedio:** El promedio de las demandas en un periodo tal como días, semanas, meses y años.
- **Demanda diversificada:** Suma de las demandas producidas por un grupo de varias cargas, para un periodo de tiempo determinado.
- **Demanda máxima diversificada:** Para un grupo de cargas, la suma de mayor tamaño de las demandas individuales de cada carga en el mismo intervalo de tiempo, en un periodo de tiempo determinado.
- **Demanda máxima no coincidente:** Para un grupo de cargas, la suma de las demandas máximas individuales de cada carga.
- **Factor de demanda:** Razón de la demanda máxima respecto a la carga conectada.
- **Factor de utilización:** Razón de la demanda máxima y la capacidad nominal.

Además, las cargas pueden ser monofásicas, bifásicas o trifásicas y para su análisis se pueden modelar como conexiones en estrella o conexiones en delta, donde se incluyen los siguientes modelos: potencia activa y reactiva constantes, corriente constante, impedancia constante o una combinación de las anteriores (Kersting & Kerestes, 2023).

2.3. Flujos de potencia

El flujo de potencia es un método utilizado para el análisis de sistemas eléctricos de potencia, tanto a nivel de transmisión como de distribución, lo que permite conocer el estado del sistema, usualmente en forma de tensiones en las diferentes barras que componen el sistema (Garcés-Ruiz, 2022).

Con esta información, es posible evaluar aspectos del sistema, tales como la estabilidad de tensión, pérdidas de energía, presencia de sobrecargas, entre otros. Esto permite obtener información sobre el estado actual del equipo, así como analizar diferentes casos hipotéticos para predecir el funcionamiento del sistema ante futuras ampliaciones o cambios.

En el caso de distribución, existen diferentes métodos de análisis de flujo de potencia donde se resalta el método de escalera. En el caso del método de escalera, es un método iterativo compuesto de dos partes, un barrido hacia adelante y un barrido hacia atrás. En el barrido hacia adelante se realiza un cálculo aguas abajo de las tensiones de carga, utilizando los valores de corriente de carga más actualizados y asumiendo para la primera iteración que las corrientes de carga son cero. Posteriormente, el barrido hacia atrás consiste en calcular las corrientes de carga aguas arriba, utilizando las tensiones de carga obtenidas en el barrido hacia adelante y finalizando así la iteración. Este proceso se repite hasta que el error entre el valor nuevo y anterior de tensión de carga se encuentra dentro de una tolerancia definida (Kersting & Kerestes, 2023).

Por otra parte, los programas de simulación de redes de distribución hacen uso de métodos y técnicas más complejos para el análisis de flujos de potencia como lo son el método de simulación de Monte Carlo (Gallego et al., 2008) o variaciones del método de escalera (Issicaba & Coelho, 2016), haciendo uso de las capacidades computacionales de los equipos modernos para obtener resultados más aproximados a la realidad y con menor error.

2.4. Sistema de Información Geográfica (GIS)

Son el conjunto de *hardware* y *software* de datos, métodos y procedimientos, con los cuales se introducen, almacenan, administran, actualizan y analizan datos con referencia geográfica de los cuales se puede obtener información relacionadas a un espacio geográfico (Tabitas-Aguilar et al., 2018).

Para implementar GIS, se debe hacer uso de un sistema de referencia de coordenadas, con el fin de conocer la posición de un punto en la superficie terrestre. Cada sistema

utiliza una parametrización propia de las coordenadas de los puntos, de forma que dos sistemas de referencia de coordenadas pueden ubicar el mismo punto con coordenadas distintas (Furones, 2011), también se debe considerar que hay sistemas que hacen uso de 2 ejes para la ubicación de las coordenadas, mientras que otros utilizan 3 ejes. De los sistemas de referencia de coordenadas, se resalta el sistema de EPSG (*European Petroleum Survey Group*), el cual sigue la norma ISO 19111:2019 y que se ha convertido en el sistema más usado en la industria geoespacial (European Petroleum Survey Group, s.f.). En el caso de Costa Rica, se hace uso del sistema de coordenadas horizontales denominado CR05, donde a su proyección cartográfica se le denomina CRTM05, utilizando EPSG: 5367 como código identificador en el sistema EPSG. A nivel nacional, también se utilizan otros códigos según la proyección empleada, los cuales son: EPSG: 5456 para la proyección Lambert Norte, EPSG: 5457 para la proyección Lambert Sur y el código 4326 para la proyección WGS84 (Instituto Geográfico Nacional, 2016).

La información suministrada por el ICE para la elaboración del proyecto se encuentra en código EPSG: 5367, el cual es utilizado únicamente para coordenadas dentro del territorio nacional y utiliza dos ejes, uno norte y otro este, en el que las unidades empleadas de los ejes corresponden a metros (Spatial Reference, s.f.-b). En el caso de OpenDSS y OpenDSS-G, se hace uso del código EPSG: 4326, el cual se utiliza a nivel mundial para ubicar cualquier posición en el globo terrestre, utiliza dos ejes, latitud geodésica y longitud geodésica, medidos en grados (Spatial Reference, s.f.-a). Por lo que es necesario realizar una conversión de los datos suministrados por el ICE a EPSG: 4326 para poder implementarlos en el *software* de simulación.

2.5. Software de simulación

Respecto al *software* de simulación, en el mercado existen varias opciones, tales como OpenDSS, ETAP, SIMARIS, entre otras opciones, cada una con diferentes especificaciones y módulos con los cuales realizar análisis de los sistemas de potencia eléctrica. Para el desarrollo de este proyecto y por indicación del ICE, se utilizará tanto OpenDSS como OpenDSS-G.

En el caso del Simulador de Sistemas de Distribución Abierto (OpenDSS, por sus siglas en inglés), como su nombre indica es un *software* de simulación de redes de distribución de código abierto con el cual realizar análisis de flujos de potencia, análisis de armónicos, análisis de fallas, entre otros. Además, al ser de código abierto permite la modificación del *software*, permitiendo agregar módulos y/o librerías acordes a la necesidad del usuario. Este se caracteriza por el hecho de que su interfaz es a base de texto, por lo que se dificulta la visualización del sistema de distribución. Entre los elementos que es capaz de simular el *software* se incluyen barras, líneas, transformadores, cargas, bancos de condensadores, reguladores de tensión, autotransformadores, generadores, equipos de almacenamiento de energía, protecciones, entre otros (Dugan & Montenegro, 2020b).

Respecto a OpenDSS-G corresponde a una interfaz gráfica para OpenDSS, esto con el fin de facilitar la manipulación de los elementos del programa de simulación, así como la visualización de los resultados. Con ello, se simplifica el uso del programa y se permite al usuario acceder de forma sencilla a las funciones avanzadas de OpenDSS. Por otra parte, incluye mapas de calor, así como nuevos gráficos con los cuales extraer mayor información de la simulación del circuito (Dugan & Montenegro, 2020a).

Por otra parte, se debe mencionar el *software* ETAP, el cual es utilizado para el análisis de los sistemas eléctricos de potencia y para simulación y estudio de sectores industriales, de transporte y edificaciones. Este *software* cuenta con una herramienta de GIS adicional que se puede utilizar en las simulaciones de los sistemas de potencia. Este programa fue utilizado para simular el Sistema de Interconexión Eléctrica de los Países de América Central (SIEPAC), inicialmente simulando cada uno de los sistemas eléctricos de los países miembros y, posteriormente, agregando las interconexiones, para un total de más 2150 barras y una desviación del 0,5 % de los resultados obtenidos, respecto a los que se utilizaron de referencia (Gómez-Ramírez, Mora-Jiménez & Meza, 2023).

2.6. Python

Es un lenguaje de programación interpretado, de propósito general y cuenta con tipado dinámico, desarrollado en 1980 por Guido van Rossum (Nolasco Valenzuela, 2018). Posee una amplia variedad de librerías que se pueden importar, entre las que se encuentran:

- **Pandas:** Es una librería utilizada para el manejo, limpieza y análisis de gran cantidad de datos, de forma rápida, flexible y poderosa. Permite la importación de datos desde archivos CSV y archivos Excel, así como la exportación de los datos a diferentes formatos (Pandas, s.f.).
- **PyPROJ:** Interfaz de Python para el uso de PROJ, el cual es un *software* para la transformación de coordenadas geospaciales de un sistema de referencia de coordenadas a otro. Incluyendo tanto proyecciones cartográficas como transformaciones geodésicas (PROJ Contributors, 2024).
- **Math:** Módulo que brinda acceso a funciones matemáticas definidas por el estándar C, permitiendo el acceso a funciones las cuales no están presentes en Python base (Python Software Foundation, 2024).

En el caso de este proyecto, la librería Pandas se utilizará para manipulación y limpieza de los datos suministrados por el ICE, de forma que se les pueda dar a los datos el formato solicitado por los programas de simulación. Además de que permite el manejo de la información brindada por el ICE, pues la gran cantidad de datos dificultaría la realización del proyecto al utilizar medios manuales.

Respecto a PyPROJ, se utilizará para convertir las coordenadas del código EPSG: 5367 en el cual se encuentran los datos del ICE, al código EPSG: 4326 el cual utilizan los programas de simulación empleados. Respecto al módulo Math, este se utilizará para realizar cálculos que incluyan raíces, debido a que Python por si solo no incluye esta función.

2.7. Simulación

En la “Ley N°10086. Promoción y regulación de recursos energéticos distribuidos partir de fuentes renovables” (2021) se indica que las empresas distribuidoras deben realizar estudios técnicos para la correcta implementación de recursos energéticos distribuidos en la red de distribución, en pro de la eficiencia y costo del servicio. Además, señala que se debe contar con la capacidad de penetración de los distintos recursos energéticos distribuidos actualizados, publicar las características de sus circuitos y la cantidad de recursos de generación distribuida presentes en estos, así como la capacidad límite de penetración de la generación distribuida. De forma que la simulación de la red de distribución sirva como herramienta para cumplir con estas regulaciones.

Por otra parte, el uso de simulaciones en redes de distribución permite conocer el estado actual del sistema, la capacidad de crecimiento de la red, el límite de crecimiento de la demanda antes de presentar problemas en la calidad de la red, así como poder analizar opciones ante problemáticas que presentan las redes de distribución, como lo son las pérdidas, la implementación de fuente de energía renovables con generación variable, problemas en la regulación de la tensión y la frecuencia, así como interrupciones del suministro eléctrico (Gómez-Ramírez et al., 2021) (Gómez-Ramírez et al., 2022). Con ello, se pueden tomar así decisiones óptimas, en favor de una red de distribución más confiable, eficiente y amigable con el medio ambiente.

Capítulo 3

Modelado de la red de distribución

El modelado está compuesto por la generación de los archivos con la información de la red de distribución, de acuerdo con el formato de OpenDSS, así como su implementación en el *software* de simulación. Entre los documentos generados, se encuentran los archivos maestros, la fuente de tensión, el documento de coordenadas de las barras, las líneas en baja y media tensión en conjunto con el archivo de sus impedancias, los transformadores, las cargas y el regulador de tensión.

Debido al tamaño del circuito y la gran cantidad de datos a utilizar, se hizo uso del *software* Python para lograr dar a los datos el formato solicitado por OpenDSS de forma rápida y eficiente, debido a que realizarlo de forma manual sería inviable para la realización de este proyecto. Además, es importante resaltar que se utilizaron diferentes bibliotecas de Python para la realización del modelado, entre las cuales se encuentran Pandas para el manejo de los datos de Excel en Python, la biblioteca Math para la realización de cálculos con raíces, la biblioteca CVS para evitar errores a la hora de exportar los archivos a formato de OpenDSS y la función Transformer de PyProj para la conversión de coordenadas.

Respecto al circuito, se compone por diferentes líneas de media tensión a 34,5 kV y líneas de baja tensión a 0,24 kV, las líneas -dependiendo de su tipo- utilizan diferentes conductores y diferentes configuraciones de la geometría de los conductores, además de haber ciertas líneas, las cuales son subterráneas. Para el caso de los transformadores, se cuentan con transformadores trifásicos en conexión estrella-delta con una tensión

de 34,5/0,24 kV, transformadores bifásicos a 19,92/0,24 kV modelados como estrella abierta - delta abierta y transformadores monofásicos de tres devanados a 19,92 kV el primario y a 0,12/0,24 kV en el secundario. Asimismo, se cuentan con diferentes cargas a diferentes potencias modeladas tanto en demanda constante como en demanda variable. Finalmente, se cuenta con un regulador de tensión a 34,5 kV con una tensión de referencia de 0,12 kV y un ancho de banda de 2 V.

3.1. Archivo maestro

El archivo maestro o master es el archivo donde se llaman el resto de documentos que contienen la información de la red de distribución, de forma que se pueda manejar de manera ordenada los datos del circuito. También en este archivo se incluyen ciertos comandos referentes al manejo de la simulación. Para la realización del proyecto, se utilizaron dos archivos principales: uno para el modelado de la red de distribución considerando las cargas con demanda constante y otro considerando las cargas con demanda variable.

3.1.1. Archivo maestro para cargas con demanda constante

A continuación, se muestra el código empleado en el archivo maestro para cargas con demanda constante.

```
//Se inicia limpiando cualquier configuración anterior
clear

//Se define la frecuencia base del circuito
Set DefaultBaseFrequency=60

//Crea un nuevo circuito
new circuit.circuitoTFG
```

```
//Se definen el máximo de iteraciones
Set maxcontroliter=30

//Redirige a un archivo con la fuente de tensión
Redirect V_source.dss

//Redirige las matrices de R, X y C segun la configuracion de la linea
Redirect LineCode.dss

//Redirige a un archivo de lista de líneas de media tensión trifásicas
Redirect Lista_Lineas_MT_3F.dss

//Redirige un archivo de lista de líneas de media tensión bifásicas
Redirect Lista_Lineas_MT_2F.dss

//Redirige a un archivo de lista de líneas de media tensión monofásicas
Redirect Lista_Lineas_MT_1F.dss

//Redirige a un archivo de lista de líneas de baja
Redirect Lista_Lineas_BT.dss

//Redirige a un archivo con los transformadores monofásicos
Redirect Trafo1F.dss

//Redirige a un archivo con los transformadores bifásicos
Redirect Trafo2F.dss

//Redirige a un archivo con los transformadores trifásicos
Redirect Trafo3F.dss
```

```
//Redirige a un archivo con las cargas
Redirect Load.dss

//Redirige a un archivo con los reguladores de tensión
Redirect Regulador3F_tipoA.dss

//Crea una lista de buses en el sistema
MakeBusList

//Se establecen las tensiones bases de los buses
Set Voltagebases=(0.24, 34.5)

//Importa las coordenadas de los buses
Buscoord BusCoords.dss

//Se crean monitores en la fuente para medir V, I, P y Q
new monitor.alimentacionVI element=line.1247375__0 terminal=1 mode=0
new monitor.alimentacionPQ element=line.1247375__0 terminal=1
~ mode=1 ppolar=no

calcv
solve

Plot Profile Phases

//show topology
```

El archivo maestro comienza limpiando la memoria de OpenDSS, mediante el comando *Clear*, con el objetivo de eliminar cualquier archivo que haya sido cargado anteriormente y que pueda afectar la simulación. Posteriormente, se define la frecuencia base del circuito estableciéndola a un valor de 60 Hz; seguidamente, se crea un circuito

al cual se le cargará la información y se establece el límite de iteraciones del método de análisis (Dugan & Montenegro, 2020b).

Posteriormente, se llaman los archivos con la información de la red de distribución mediante el comando *Redirect*. Luego, haciendo uso del comando *MakeBusList*, se crea una lista de barras y se importan las coordenadas con el comando *Buscoord* (Dugan & Montenegro, 2020b), además de indicar las tensiones bases línea-línea de las barras, estableciendo los dos niveles de tensión de 0,24 y 34,5 kV. Seguidamente, se crean dos monitores para medir tensión, corriente, potencia real y potencia reactiva.

Con *Calcv* se obtienen las tensiones de las barras y *Solve* inicia el modo de solución de OpenDSS. Con *Plot Profile Phases*, se grafica la tensión de las fases en p.u. respecto a la distancia desde la fuente de tensión en kilómetros. Finalmente, con *Show Topology*, se muestra información referente a la topología de la red de distribución (Dugan & Montenegro, 2020b).

3.1.2. Archivo maestro para cargas con demanda variable

A continuación, se muestra el código del archivo maestro para cargas con demanda variable.

```
//Se inicia limpiando cualquier configuración anterior
clear

//Se define la frecuencia base del circuito
Set DefaultBaseFrequency=60

//crea un nuevo circuito
new circuit.circuitoTFG

//Se definen el máximo de iteraciones
Set maxcontroliter=30
```

```
//Se establece el perfil de demanda de las cargas
New Loadshape.Perfil npts=144 minterval=10 mult=(file=loadshape_1.txt)

//Redirige a un archivo con la fuente de tensión
Redirect V_source.dss

//Redirige las matrices de R, X y C segun la configuracion de la linea
Redirect LineCode.dss

//Redirige a un archivo de lista de líneas de media tensión trifásicas
Redirect Lista_Lineas_MT_3F.dss

//Redirige un archivo de lista de líneas de media tensión bifásicas
Redirect Lista_Lineas_MT_2F.dss

//Redirige a un archivo de lista de líneas de media tensión monofásicas
Redirect Lista_Lineas_MT_1F.dss

//Redirige a un archivo de lista de líneas de baja tensión
Redirect Lista_Lineas_BT.dss

//Redirige a un archivo con los transformadores monofásicos
Redirect Trafo1F.dss

//Redirige a un archivo con los transformadores bifásicos
Redirect Trafo2F.dss

//Redirige a un archivo con los transformadores trifásicos
Redirect Trafo3F.dss
```

```
//Redirige a un archivo con las cargas
Redirect LoadVar.dss

//Redirige a un archivo con los reguladores de tensión
Redirect Regulador3F_tipoA.dss

//Crea una lista de buses en el sistema
MakeBusList

//Se establecen las tensiones bases de los buses
Set Voltagebases=(0.24, 34.5)

//Importa las coordenadas de los buses
Buscoord BusCoords.dss

//Se crean monitores en la fuente para medir V, I, P y Q
new monitor.alimentacionVI element=line.1247375__0 terminal=1 mode=0
new monitor.alimentacionPQ element=line.1247375__0 terminal=1
~ mode=1 ppolar=no

//Se establece el modo de simulación de la demanda
set mode=daily stepsize=10m number=144

calcv
solve

Plot Profile Phases

show topology
```

Este archivo maestro agrega el perfil de carga mediante el comando *New Loadshape*, con un total de 144 mediciones a intervalos de 10 minutos para cubrir la demanda las 24 horas del día, además, accede a los archivos con los perfiles de demanda mediante el parámetro *Mult*, de los cuales se cuenta con 6 archivos. Luego, se sustituye el archivo de cargas con demanda fija por el archivo de cargas con demanda variable. Finalmente, mediante el comando *Set*, se establece el modo de la simulación de la demanda, en el que se indica el parámetro *Mode* que sea un análisis diario, con intervalos de medición de la demanda de 10 minutos, mediante el parámetro *Stepsize* y con un total de 144 mediciones con el parámetro *Number*.

3.2. Fuente de tensión

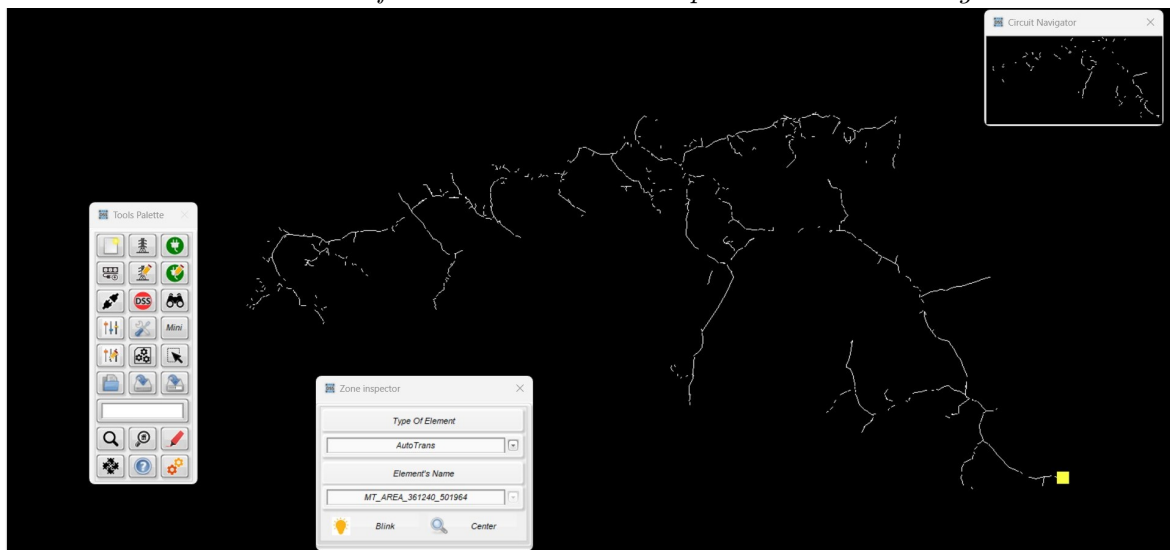
Para el modelado de la fuente de tensión o Vsource de OpenDSS, se hizo uso de un archivo de referencia brindado por el ICE. Se tomaron valores por defecto de este archivo y se adaptaron otros valores acorde al circuito asignado. A continuación, se muestra el código empleado en OpenDSS para la creación del Vsource:

```
Edit "Vsource.source" bus1=MT_AREA_361240_501964 basekv=34.5  
R1=0 X1=0.001 R0=0 X0=0.001 MVAsc3=457.992
```

En este se puede apreciar que la primera indicación consiste en modificar el Vsource interno de OpenDSS con los parámetros que se le indican posteriormente. En el caso del parámetro *Bus1*, se indica la barra donde se ubicará la terminal por la cual la fuente de tensión se conectará al resto del circuito (Dugan & Montenegro, 2020b), debido a que no se indicó por parte del ICE la barra conectada a la subestación se tuvo que elegir una para la conexión del sistema. Se escogió una barra que solo fuera utilizado por una línea y que funcionara como punto de origen radial para el resto de barras, considerando lo anterior la barra elegida fue MT_AREA_361240_501964, la cual se puede visualizar en la Figura 3.1 en vista general y en la Figura 3.2 en vista ampliada, donde corresponde al cuadrado amarillo.

Figura 3.1

Ubicación del “bus1” de la fuente de tensión en OpenDSS-G en vista general



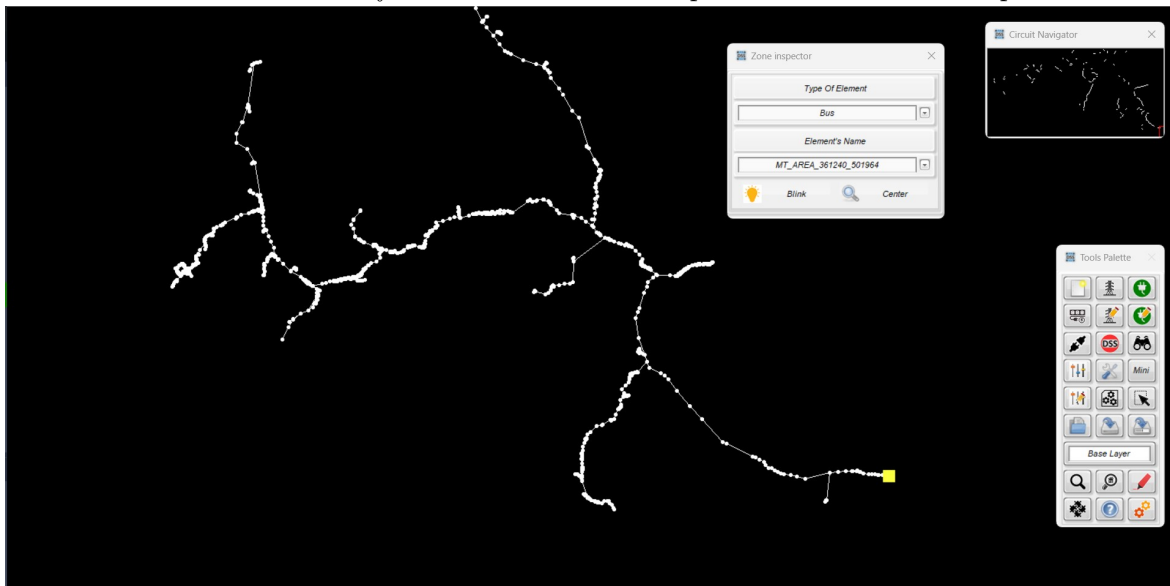
Nota. La ubicación corresponde al cuadrado amarillo. Las separaciones del circuito se deben al zoom empleado para la visualización de la red.

Respecto al parámetro $Basekv$, con él se indica la tensión línea-línea base de la fuente de tensión, donde se escogió al valor de media tensión de la red de distribución, que corresponde a 34,5 kV. En el caso de $R1$ y $R0$, corresponden a la resistencia de secuencia positiva y la resistencia de secuencia cero respectivamente; del mismo modo, $X1$ y $X0$, corresponden a la reactancia de secuencia positiva y la reactancia de secuencia cero respectivamente. Referente al parámetro $MVAsc3$, este corresponde a la potencia trifásica de cortocircuito en MVA (Dugan & Montenegro, 2020b). Los parámetros $R1$, $R0$, $X1$, $X0$ y $MVAsc3$ se dejan igual a la referencia, debido a que no se indica otro valor con el cual modificarlos.

También cabe mencionar que la fuente de tensión puede solicitar los siguientes parámetros: pu que corresponde a la tensión base por unidad que por defecto es 1; $Phases$ que serían las fases del circuito que por defecto son 3; $frequency$ que corresponde a la frecuencia del sistema y la cual se indicó de 60 Hz en el archivo maestro, y $MVAsc1$ que corresponde a la potencia monofásica de cortocircuito en MVA que, por defecto, es de 2100 (Dugan & Montenegro, 2020b) (Leal Parra, 2016).

Figura 3.2

Ubicación del “bus1” de la fuente de tensión en OpenDSS-G en vista ampliada



Nota. La ubicación corresponde al cuadrado amarillo. Cada punto blanco corresponde a una barra.

3.3. Barras

En OpenDSS solo existe un tipo de barra, la cual cuenta con 1 o más nodos acorde a los elementos conectados a él, además, los buses no cuentan con ninguna restricción en la cantidad de elementos que pueden conectarse a él. En OpenDSS, los buses se crean al llamarlos como parámetros en otros elementos, por lo que propiamente no existe un comando con el cual crear los buses (Dugan & Montenegro, 2020b). Pero sí existen comandos con los cuales definir las coordenadas de los buses, así como las tensiones base:

```
//Coordenadas
//Busname, X, Y
MT_AREA_361240_501964, -85.26575029901038, 10.031486773552952

//Tensiones base
Set Voltagebases=(0.24, 34.5)
```

Para el caso del comando de las coordenadas, este consta de tres parámetros *Busna-*

me (que corresponde a la cadena o *string* que identifica a la barra), *X* (que corresponde a la coordenada de latitud) y *Y* (que corresponde a la coordenada de longitud). Cabe mencionar que estos valores son principalmente requeridos en OpenDSS-G, aunque OpenDSS los acepta, no son necesarios para la simulación. Respecto al comando de tensiones base, solo recibe un parámetro, que es un vector o *array* de tensiones de líneas de las barras en kV (Dugan & Montenegro, 2020b).

3.3.1. Barras: Minería de datos

Respecto a la información brindada por el ICE, el archivo con la información del circuito cuenta con 20 993 barras, en la Tabla 3.1 se muestra en forma reducida los datos de los buses suministrados, donde se aprecia que hay información repetida, como lo son el identificador de la barra y las coordenadas, además de que existen columnas vacías.

Tabla 3.1

Información suministrada de las barras

	Name	AliasName	Un	Uset	Umax	Umin	Ir	Ipmax	Zone	Area	CoordX1	CoordY1	CoordX2	CoordY2
0	MT_AREA_318761_765812	MT_AREA_318761_765812	34,5								318761,765812	1099130,40225	318761,765812	1099130,40225
1	MT_AREA_318793_532414	MT_AREA_318793_532414	34,5								318793,532414	1099084,63751	318793,532414	1099084,63751
...
20991	2853445_T	2853445_T	0,24								322519,971005	1095293,80030	322519,971005	1095293,80030
20992	2866252_T	2866252_T	0,24								339257,997944	1090088,46719	339257,997944	1090088,46719

Nota. Información original del Instituto Costarricense de Electricidad.

Aunque se puede procesar la información con este formato en Python mediante Pandas, requiere mayor capacidad de procesamiento por parte del equipo, así como dificultar el manejo de las columnas y filas en Python, por lo que se realiza un trabajo de formato al archivo, con el fin de facilitar el manejo para la creación del documento de coordenadas de las barras. Por otra parte, existen ciertos buses con coordenadas iguales o muy cercanas, las cuales generan que no se conecten debidamente las líneas del circuito, debido al funcionamiento de OpenDSS, por lo que detectar y eliminar estos duplicados es de importancia para la correcta implementación de los datos en el *software* de simulación.

Para lograr lo anterior, se implementó un programa nombrado LimpiadorExcel.py, el cual se puede visualizar completo en el anexo A.1. En este programa primero se le

carga la hoja de barras a Python en forma de una tabla o *dataframe* de Pandas llamado *Busbar* (línea 3). Posteriormente, se eliminan las columnas con información repetida de identificador y coordenadas (línea 4). Enseguida, se crea un *array* con el nombre de las columnas que se utilizará para eliminar mediante un ciclo *for* las columnas vacías (líneas 6-8). Con ello, se logra que el *dataframe* quede con el formato de la Tabla 3.2.

```

1  import pandas as pd
2
3  Busbar=pd.read_excel('CircuitoTFG.xlsx', index_col=0, decimal='.')
4
5  Busbar=Busbar.drop(columns=['AliasName', 'CoordX2', 'CoordY2'])
6  headers=Busbar.columns
7  for k in range(2, (len(headers)-2)):
8      Busbar=Busbar.drop(columns=[headers[k]])

```

Tabla 3.2

Información suministrada de las barras con columnas modificadas

Name	Un	CoordX1	CoordY1
MT_AREA_318761_765812	34,5	318761,765812	1099130,40225
MT_AREA_318793_532414	34,5	318793,532414	1099084,63751
...
2853445_T	0,24	322519,971005	1095293,80030
2866252_T	0,24	339257,997944	1090088,46719

Nota. Información original del Instituto Costarricense de Electricidad.

Posteriormente, se crea una copia del *dataframe* con la información de las barras (línea 9). En esta copia, a cada coordenada de cada barra se le reducirán la cantidad de caracteres a 11 mediante un ciclo *for* (líneas 10-12), de forma que las barras que tienen coordenadas muy cercanas pasen a tener coordenadas iguales. Seguidamente, se crea un *dataframe* llamado *Filtro*, el cual mediante la función *duplicated* detecta las barras que tengan repetido la tensión, la coordenada X y la coordenada Y, y las almacena (línea 14) solo que esta función no indica cuáles barras son iguales.

```

9   Busbarcor=Busbar.copy()
10  for k in range (len(Busbarcor)):
11      Busbarcor.iat[k,2]= float(str(Busbarcor.iloc[k,2])[:11])
12      Busbarcor.iat[k,3]= float(str(Busbarcor.iloc[k,3])[:11])
13
14  Filtro=Busbarcor[Busbarcor.duplicated(
    subset=['Un', 'CoordX1', 'CoordY1'], keep=False) == True]

```

Para identificar las parejas de barras iguales, se crea un *dataframe* vacío (línea 15) llamado *BusRep* con dos columnas “Bus1” y “Bus2” y con una cantidad de filas igual al *dataframe* *Filtro*, posteriormente, mediante dos ciclos *for*, se recorren las barras presentes en *Filtro* y se comparan las coordenadas (líneas 16-20), de forma que, si comparten las mismas coordenadas, se almacenan los nombres de las dos barras iguales en *BusRep*. Luego, se eliminan las filas vacías de *BusRep* (línea 22).

```

15  BusRep=pd.DataFrame(columns=['Bus1', 'Bus2'], index=range(len(Filtro)))
16  for k in range(len(Filtro)):
17      for c in range(k+1, len(Filtro)):
18          if Filtro.iloc[k, 3] == Filtro.iloc[c, 3] and
            Filtro.iloc[k, 2] == Filtro.iloc[c, 2]:
19              BusRep.iat[k,0]=Filtro.iat[k,0]
20              BusRep.iat[k,1]=Filtro.iat[c,0]
21
22  BusRep=BusRep.dropna()

```

Después, mediante un ciclo *for*, se eliminan del *dataframe* *Busbar* las barras que aparecen en la columna “Bus2” de *BusRep* (líneas 23-27). Para esto, se crea una variable que almacena la fila de *Busbar* que tenga el identificador de barra igual al identificador presente en la columna “Bus2” de *BusRep* (línea 24), posteriormente, de esta variable se extrae el número de fila correspondiente a la fila almacenada (línea 25-26). Este valor se utilizará en la función *drop* para eliminar la fila con la barra con coordenadas iguales a otra (línea 27).

```

23 for k in range(len(BusRep)):
24     Bus = Busbar[Busbar['Name'] == BusRep.iloc[k,1]]
25     BusInd=Bus.index
26     BusInd=BusInd.to_list()
27     Busbar=Busbar.drop(BusInd[0], axis=0)

```

Finalmente, el *dataframe* *Busbar* se exporta a un archivo Excel para su posterior uso y/o visualización, sin columnas innecesarias y sin barras con coordenadas iguales o muy cercanas, reduciendo el número de barras a un total de 20 957. Cabe mencionar que el reemplazo de las barras eliminadas por las barras con las mismas coordenadas que no fueron eliminadas debe realizarse en la información de las líneas, la cual se explica en la sección 3.4.1.

3.3.2. Barras: Formato OpenDSS

Para el ingreso de las coordenadas de las barras en los programas de simulación, especialmente en OpenDSS-G, es requerido crear un archivo con extensión .dss, donde se muestre cada barra y sus coordenadas acorde al formato indicado al principio de la sección 3.3. Para esto, se hace uso del programa *CreadorInputsOpenDSS.py*, el cual se puede observar completo en el anexo A.2.

Primero, se importa la hoja de Excel con los datos de las barras a un *dataframe* llamado *BusOrig* (línea 6), posteriormente, se eliminan las columnas innecesaria y se almacena en un nuevo *dataframe* llamado *BusCoords* (línea 7). Luego, se crea una variable llamada *transformer*, en la que se almacena la función *from_crs* (línea 9), la cual se utiliza para indicar que se requiere convertir de código EPSG: 5367 a código EPSG: 4326, siendo el primer y el segundo parámetro de la función respectivamente.

Seguidamente, se actualiza el *dataframe* *BusCoords* al formato de OpenDSS (líneas 10-14), primero, realizando la conversión de las coordenadas por medio de la función *transform* (línea 11) que devuelve una tupla con las coordenadas en código EPSG: 4326. Posteriormente, se le agrega una coma al identificador de la barra (línea 12), después, se actualiza la coordenada X o de latitud (línea 13) del *dataframe* *BusCoords*

y se le agrega una coma, luego, para terminar el ciclo se actualiza la coordenada Y o de longitud (línea 14). Finalmente, se exporta el *dataframe* con el formato de OpenDSS al archivo BusCoords.dss (línea 15).

```

1  import pandas as pd
2  import csv
3  import math
4  from pyproj import Transformer
5
6  BusOrig=pd.read_excel('CircuitoLimpio.xlsx', decimal='.')
7  BusCoords=BusOrig.drop(columns=['Un'])
8
9  transformer = Transformer.from_crs("EPSG:5367", "EPSG:4326")
10 for k in range(0,len(BusOrig)):
11     Conv=transformer.transform(BusCoords.iloc[k, 2],
12                               BusCoords.iloc[k, 1])
12     BusCoords.iat[k,0]=(BusCoords.iat[k,0]+",")
13     BusCoords.iat[k,1]=(str(Conv[1])+",")
14     BusCoords.iat[k,2]=Conv[0]
15 BusCoords.to_csv('BusCoords.dss', sep=' ', index=False,
16                 header=False)

```

3.4. Líneas

Respecto a las líneas en OpenDSS, son elementos de suministro de potencia conectados a dos nodos descritos por su matriz de impedancias. En OpenDSS las líneas se pueden definir de dos maneras (Dugan & Montenegro, 2020b):

- Estableciendo la información propia de los conductores de las líneas con *WireData* y definiendo la geometría de los conductores mediante un *LineGeometry*, de forma que el programa calcule la matriz de impedancias.

- Importando la matriz de impedancias mediante un *LineCode*.

Para la elaboración de este proyecto, se hizo uso de ambos métodos, el primero con el fin de generar los *LineCode* para cada tipo de línea que utiliza la red distribución; y el segundo para cargar la matriz de impedancias de las líneas en la simulación sin requerir hacer el mismo cálculo de impedancias varias veces. Respecto a *WireData* y *LineGeometry*, estos cuentan con el siguiente formato:

```
//WireData:
New WireData.ACSR_2_7STR Rac=0.338 Runits=kft GMR=0.0107
~ GMRunits=ft DIAM=0.325 Radunits=in Normamps=184

//LineGeometry:
New Linegeometry.L1 nconds=2 nphases=1 reduce=y
~ cond=1 Wire=ACSR_2_7STR x=0 h=10.74
~ cond=2 Wire=ACSR_2_7STR x=0 h=6.7
```

En lo que respecta al *WireData*, comienza con el comando *New WireData*, seguido de un identificador propio que corresponde a los caracteres posteriores al punto, luego, recibe los siguientes parámetros (Dugan & Montenegro, 2020b):

1. *Rac*: Corresponde a la resistencia en corriente alterna, en ohms por unidad de longitud a una frecuencia de 60 Hz.
2. *Runits*: La unidad de longitud utilizada en el parámetro anterior.
3. *GMR*: Radio medio geométrico (GMR) del conductor.
4. *GMRunits*: Corresponde a las unidades del GMR.
5. *Diam*: Diámetro externo del conductor, puede reemplazarse por el parámetro *Radius* que corresponde al radio externo del conductor.
6. *Radunits*: Unidades del radio o diámetro externo del conductor.
7. *Normamps*: Corresponde a la corriente normal del conductor en amperes.

Para el caso del *LineGeometry*, este se inicia mediante el comando *New LineGeometry*, seguido de un identificador, posteriormente, se dan los parámetros del comando (Dugan & Montenegro, 2020b):

1. *Nconds*: Cantidad de conductores presentes en la línea.
2. *Nphases*: Cantidad de conductores que son fases.
3. *Reduce*: Se indica si se desea realizar una reducción de Kron. Puede ser *Yes* o *No*.
4. *Cond*: Establece el número del conductor.
5. *Wire*: Indica el *WireData* al cual debe redirigirse para obtener la información del conductor.
6. *X*: Coordenada perpendicular a la altura. Es una coordenada relativa y puede tener una referencia arbitraria que usualmente corresponde a un conector, el cual se define con la posición 0.
7. *H*: Altura del conductor respecto a la tierra.

Respecto a los *LineCode*, estos comienzan con el comando *New Linecode* seguido de un identificador. Posteriormente, se le ingresan los parámetros (Dugan & Montenegro, 2020b):

1. *Nphases*: Cantidad de fases.
2. *Units*: Unidad de longitud de la matriz de impedancias.
3. *Rmatrix*: Matriz de resistencias en ohms por unidad de longitud.
4. *Xmatrix*: Matriz de reactancias en ohms por unidad de longitud.
5. *Cmatrix*: Matriz de capacitancias nodales en derivación, en nanofaradios por unidad de longitud.

```
//LineCode:
New Linecode.MT_ABC:_2_ACSR_AAAC_1/0_ACSR_1/0_2.1_Y nphases=3 Units=km
~ Rmatrix=[1.24011 |0.133652 1.24512 |0.131186 0.133652 1.24011]
~ Xmatrix=[0.683609 |0.271606 0.672079 |0.191101 0.214339 0.683609]
~ Cmatrix=[8.31661 |-2.3618 8.59806 |-0.814425 -1.32743 7.87278]
```

Por otra parte, para crear el elemento de línea en OpenDSS, se utiliza el comando *New Line* seguido de un identificador propio para cada línea. Los parámetros que recibe son los siguientes:

1. *Bus1*: Nombre de la barra que sirve como terminal 1. Si después del identificador de la barra se le coloca un punto, se puede indicar los nodos a los cuales se conectan los conductores por medio de números. En caso de no indicarse OpenDSS, se colocaran los conductores empezando desde el nodo 1.
2. *Bus2*: Nombre de la barra que sirve como terminal 2. Se pueden indicar los nodos al cual se conectarán los conductores del mismo modo que se hace con *Bus1*.
3. *LineCode*: Identificador del LineCode del cual se importará la matriz de impedancias.
4. *Length*: Longitud de la línea, la cual se utilizará para multiplicar la matriz de impedancias.

```
New line.52422__0 bus1=MT_AREA_338083_748526.3 bus2=MT_SUB_338084_688.3
~ LineCode=MT_SUB_C:__1/0_COBRE_EPR_33 length=0.0009479950997093654
```

Finalmente, se debe mencionar que los *WireData* de todos los conductores empleados, así como los *LineCode* de las líneas de servicio y subterráneas de baja tensión fueron brindados por el Arguello et al. (2018). Los *WireData* se pueden observar en el anexo B.

3.4.1. Líneas: Minería de datos

Antes de mostrar la forma como se generaron los *LineCode*, primero se debe mencionar que en la información brindada por el ICE, cada línea cuenta con un código de librería que brinda información sobre los conductores y su geometría, como el que se aprecia en la Figura 3.3. La información que se puede extraer es la siguiente:

1. **Término en amarillo:** Se indica la tensión de la línea. “MT” para media tensión y “BT” para baja tensión.
2. **Término en verde:** Indica la fase o las fases de la línea. “A”, “B” o “C” para líneas monofásicas, “AB”, “BC” o “AC” para líneas bifásicas y “ABC” para líneas trifásicas. También pueden incluir un “SUB” al principio para indicar que son líneas subterráneas.
3. **Término en celeste:** Indica el calibre y el tipo de cable del conductor de fase.
4. **Término en rosa:** Indica el calibre y el tipo de cable del conductor neutro. En caso de no contar con conductor neutro, se coloca “NT_NT”.
5. **Término en naranja:** Indica el calibre y el tipo de cable del hilo guarda. En caso de no contar con hilo guarda se coloca “NT_NT”.
6. **Término en violeta:** En el caso de las líneas de media tensión, se indica la configuración geométrica de la línea acorde a *Proyecto: Control y Aseguramiento de la calidad de los datos del SIRDE, Configuraciones MT Aérea y Subterránea* del Instituto Costarricense de Electricidad (2023). Para el caso de líneas de baja tensión, si es un “1” indica que es aérea, si indica otro número la línea es de servicio. Las líneas de baja tensión subterráneas no incluyen este término.

Para la obtención de los *LineCode* de las líneas, se utilizó el siguiente programa en OpenDSS, el cual se obtuvo de “Manual de uso del programa OpenDSS para el curso de distribución y transmisión de potencia” por Leal Parra (2016):

Figura 3.3

Ejemplo de código de librería de una línea

MT_ABC: 4/0 ACSR_ACSR_1/0_ACSR_1/0_2.1_Y

Nota. Información original del Instituto Costarricense de Electricidad.

```
!MT_ABC: 4/0 ACSR_ACSR_1/0_ACSR_1/0_2.1_Y
```

```
Clear
```

```
New Circuit.3FAer basekv=34.5 pu=1.0 angle=0 frequency=60
```

```
~ phases=3 Mvasc3=2000 Mvasc1=1500
```

```
set EarthModel=Carson
```

```
Redirect WireDataACSR.dss
```

```
Redirect WireDataAAAC.dss
```

```
Redirect WireDataAAC.dss
```

```
Redirect CNData_CU_EPR133.dss
```

```
!!! Definimos la configuracion de la linea
```

```
New Linegeometry.L1 nconds=5 nphases=3 reduce=y
```

```
~ cond=1 Wire=ACSR_4/0_6STR x=-1.12 h=8.64 units=m
```

```
~ cond=2 Wire=ACSR_4/0_6STR x=-0.406 h=8.64 !units=m
```

```
~ cond=3 Wire=ACSR_4/0_6STR x=1.12 h=8.64 !units=m
```

```
~ cond=4 Wire=ACSR_1/0_6STR x=0 h=6.7 ! units=m ! neutro
```

```
~ cond=5 Wire=ACSR_1/0_6STR x=0 h=10.29 ! units=m ! guarda
```

```
!!Definimos la conexion
```

```
New Line.lineprimaria Bus1=B1.1.2.3 Bus2=B2.1.2.3 length=1 units=km
```

```
!!!Muestra resultados
```

```
Show lineconstants freq=60 units=km
```

Donde, primero se limpia la memoria de OpenDSS con *Clear*. Se crea un nuevo circuito en el cual se cargará la línea para la obtención de su matriz de impedancias, seguidamente, se establece que el modelo empleado para la obtención de la matriz es el de Carson mediante el comando *Set EarthModel*. Luego, se cargan los *WireData* desde otros archivos. Se crea la configuración geométrica de la línea con el comando *LineGeometry*. Posteriormente, se crea una línea que se utilizará como muestra para la creación del *LineCode* y, por último, se obtiene este último con el comando *Show LineConstants* mostrando un resultado como el que se aprecia en la Figura 3.4. A este *LineCode* se le modifica el identificador acorde al código de librería y se agrega a un documento que recopila todos los *LineCode*, finalmente, se repite el proceso para cada línea. En el caso de las líneas de baja tensión subterráneas y de servicio, al contar directamente con los *LineCode*, simplemente se les modificó el identificador acorde al código de librería y se agregó al archivo que contiene los *LineCode*.

Figura 3.4

Resultado del programa para generar LineCode

```
!--- OpenDSS Linecodes file generated from Show LINECONSTANTS command
!--- Frequency = 60 Hz, Earth resistivity = 100 ohm-m
!--- Earth Model = Carson

New Linecode.l1 nphases=3 Units=km
~ Rmatrix=[0.52626 |0.138391 0.531441 |0.13584 0.138391 0.52626 ]
~ Xmatrix=[0.648513 |0.275805 0.637142 |0.195222 0.218539 0.648513 ]
~ Cmatrix=[9.14014 |-2.80075 9.48957 |-0.922997 -1.54983 8.56664 ]
```

Por otra parte, respecto a la información brindada por el ICE, esta cuenta con 19 995 líneas, en la Tabla 3.3 se observa de forma simplificada la información de líneas suministrada, donde se puede apreciar que se cuenta con columnas con información innecesaria para OpenDSS, las cuales se eliminarán para facilitar el manejo de la información. Por otra parte, debido a que se eliminaron barras con coordenadas iguales

o muy cercanas para asegurar la conexión de las líneas, se debe actualizar las barras de las líneas. Para esto, se continuará haciendo uso del programa `LimpiadorExcel.py`, debido a que ya cuenta con la información de las barras a reemplazar.

Tabla 3.3

Información suministrada de las líneas

	Switch1	Switch2	Node1	Node2	Name	AliasName1	LibraryType	LineType	Phase	Length	Un
0	True	True	BT_SUB_317811_886402	BT_SUB_317776_732343	10006__0	10006__0	BT_SUB_A: _250 COBRE RHH_1/0		4	0,045179	0,24
1	True	True	BT_SUB_317776_732343	BT_SUB_317786_228639	10007__0	10007__0	BT_SUB_A: _250 COBRE RHH_1/0		4	0,009527	0,24
...
19993	True	True	MT_AREA_348227_487805	MT_AREA_348228_520168	9135094__0	9135094__0	MT_A: 6 CO_TR_NT_NT_NT_NE_N		1	0,001250	34,5
19994	True	True	MT_AREA_317752_941225	MT_AREA_317753_357782	10256995__0	10256995__0	MT_ABC: 6 CO_TR_NT_NT_NT_NE_N		7	0,001250	34,5

Nota. Información original del Instituto Costarricense de Electricidad.

Para lograr lo anterior, primero se carga a un *dataframe* llamado *LineAsym* la información de las líneas (línea 29), después, se eliminan las columnas con información que no es requerida (línea 30). Luego, mediante la función *merge* se unen los *dataframe* de *LineAsym* y *BusRep* en un *dataframe* llamado *LimpBus1* (línea 32), donde solo quedan las líneas que requieren que se les reemplace una barra en la terminal 1, además del identificador de la barra a reemplazar y el identificador de la barra con el cual se va a reemplazar. Posteriormente se realiza el reemplazo de las barras de la terminal 1 de las líneas, primero, se crea una variable con la cual se localiza la fila que contiene la barra a reemplazar (líneas 34-35), seguidamente, de esta variable se extrae el número de fila correspondiente a la fila almacenada (línea 36-37). Utilizando este valor, se reemplaza la barra eliminada con la barra que tiene las mismas coordenadas (línea 38). Se repite el proceso anterior para la terminal 2 (líneas 40-46) reemplazando así todas las barras eliminados. Finalmente, se reordenan las columnas (línea 47), de forma que tengan el orden acorde al formato solicitado por OpenDSS.

```

29 LineAsym=pd.read_excel('CircuitoTFG.xlsx', sheet_name='LineAsym',
    decimal='.')
30 LineAsym=LineAsym.drop(columns=['Index', 'Switch1', 'Switch2',
    'AliasName1', 'LineType', 'Phase', 'Un'])
31
32 LimpBus1=LineAsym.merge(BusRep, left_on='Node1', right_on='Bus2')
33 for k in range(len(LimpBus1)):

```

```

34     Bus = LineAsym[LineAsym['Node1']] == LimpBus1.iloc[k,6]]
35     Bus = Bus[Bus['Node2']] == LimpBus1.iloc[k,1]]
36     BusInd=Bus.index
37     BusInd=BusInd.to_list()
38     LineAsym.iat[BusInd[0],0]=LimpBus1.iat[k,5]
39
40 LimpBus2=LineAsym.merge(BusRep, left_on='Node2', right_on='Bus2')
41 for k in range(len(LimpBus2)):
42     Bus = LineAsym[LineAsym['Node2']] == LimpBus2.iloc[k,6]]
43     Bus = Bus[Bus['Node1']] == LimpBus2.iloc[k,0]]
44     BusInd=Bus.index
45     BusInd=BusInd.to_list()
46     LineAsym.iat[BusInd[0],1]=LimpBus2.iat[k,5]
47 LineAsym=LineAsym[['Name', 'Node1', 'Node2', 'LibraryType', 'Length']]

```

Además, dentro de la información suministrada, hay tres líneas que generan anillos en el circuito las cuales se pueden visualizar en la Figura 3.5 y que corresponde a las líneas 11376012__0, 10549086__0 y 11584324__0, las cuales se identificaron mediante el comando *Show Loops* y su visualización en OpenDSS-G. Estas líneas fueron eliminadas manualmente del archivo Excel con la información del circuito. Por otra parte, al agregar el regulador de tensión, este queda en paralelo con la línea 7780791__1, por lo que se debió eliminar de forma manual esta línea para la correcta implementación del regulador de tensión. Con ello, se reduce la cantidad de líneas a un total de 19 991.

Figura 3.5

Anillos generados por las líneas eliminadas

```

Loops and Paralleled Lines in all EnergyMeter Zones
(sub) Line.11376013__0: LOOPED TO      Line.11376012__0
(sub) Line.11376012__0: LOOPED TO      Line.11376013__0
(sub) Line.2281346__0: PARALLEL WITH Line.10549086__0
(sub) Line.2281346__0: LOOPED TO      Line.10549086__0
(sub) Line.10549086__0: PARALLEL WITH Line.2281346__0
(sub) Line.10549086__0: LOOPED TO      Line.2281346__0
(sub) Line.11584324__0: LOOPED TO      Line.11584325__0
(sub) Line.11584325__0: LOOPED TO      Line.11584324__0

```

3.4.2. Líneas: Formato de OpenDSS

Para dar el formato solicitado por OpenDSS a las líneas, se hizo uso del programa CreadorInputsOpenDSS.py. Primero, se importa a un *dataframe* llamado *LineAsym*, la hoja con la información de las líneas (línea 17), posteriormente, se crean *dataframe* vacíos donde se clasificarán las líneas acorde a su tipo, de forma que se les pueda dar el formato adecuado a cada tipo de línea (líneas 18-21). También se crean variables con cadenas o *strings*, las cuales se utilizarán para identificar el tipo de línea (líneas 23-37).

```

17 LineAsym=pd.read_excel('CircuitoLimpio.xlsx',
    sheet_name='LineAsym', decimal='.')
18 Lista_Lineas_MT_3F=pd.DataFrame()
19 Lista_Lineas_MT_2F=pd.DataFrame()
20 Lista_Lineas_MT_1F=pd.DataFrame()
21 Lista_Lineas_BT=pd.DataFrame()
22
23 comp1='MT_ABC'
24 comp2='MT_SUB_ABC'
25 comp3='MT_A:'
26 comp4='MT_B:'
27 comp5='MT_C:'
28 comp6='MT_SUB_A:'
29 comp7='MT_SUB_B:'
30 comp8='MT_SUB_C:'
31 comp9='MT'
32 comp10='MT_AB:'
33 comp11='MT_SUB_AB:'
34 comp12='MT_AC:'
35 comp13='MT_SUB_AC:'
36 comp14='MT_BC:'
37 comp15='MT_SUB_BC:'

```

Luego, se revisa el código de librería de cada línea, en caso de que los primeros caracteres sean iguales a “MT_ABC” o a “MT_SUB_ABC”, entonces la línea es de media tensión trifásica y se agrega al respectivo *dataframe* (líneas 40-42). De no ser el caso, se entra al siguiente *elif* donde se comparan los primeros caracteres del código de librería y si coinciden con “MT_A:”, “MT_B:”, “MT_C:”, “MT_SUB_A:”, “MT_SUB_B:” o “MT_SUB_C:” es una línea de tensión monofásica y se agrega al respectivo *dataframe* (líneas 43-45). De no cumplir con las condiciones anteriores, continua con el siguiente *elif*, el cual compara los dos primeros caracteres del código de librería y si es igual a “MT” corresponde a una línea de media tensión bifásica al ser el único tipo de líneas de medio tensión restante, posteriormente se agrega al respectivo *dataframe* (líneas 46-48). Finalmente, en caso de no cumplir con ninguna de las condiciones anteriores, entonces la línea corresponde a una de baja tensión y se agrega a su respectivo *dataframe* (líneas 49-51).

```

39 for k in range(len(LineAsym)):
40     if LineAsym.iloc[k,3][0:6] == comp1 or
        LineAsym.iloc[k,3][0:10] == comp2:
41         if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
42             Lista_Lineas_MT_3F =
                Lista_Lineas_MT_3F.append(LineAsym.iloc[k])
43     elif LineAsym.iloc[k,3][0:5] == comp3
        or LineAsym.iloc[k,3][0:5] == comp4
        or LineAsym.iloc[k,3][0:5] == comp5
        or LineAsym.iloc[k,3][0:9] == comp6
        or LineAsym.iloc[k,3][0:9] == comp7
        or LineAsym.iloc[k,3][0:9] == comp8:
44         if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
45             Lista_Lineas_MT_1F =
                Lista_Lineas_MT_1F.append(LineAsym.iloc[k])
46     elif LineAsym.iloc[k,3][0:2] == comp9:
47         if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:

```

```

48         Lista_Lineas_MT_2F =
           Lista_Lineas_MT_2F.append(LineAsym.iloc[k])
49     else:
50         if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
51             Lista_Lineas_BT =
               Lista_Lineas_BT.append(LineAsym.iloc[k])

```

Respecto al formato de las líneas, se hará mención al caso de las líneas de media tensión trifásico, pero el procedimiento es el mismo para todas las líneas, con pequeñas variaciones en las líneas monofásicas y bifásicas de media tensión. Primero, se crea el comando *New Line* (línea 54) en conjunto con el identificador propio de la línea, posteriormente, se definen los parámetros *Bus1* y *Bus2* de acuerdo con las barras correspondientes (líneas 55 y 56). Luego, se establece el parámetro *LineCode* con el código de librería de la línea (línea 57), después, se le da el parámetro *Length* la longitud de la línea (línea 58). Finalmente, se exporta el *dataframe* a un archivo en *.dss* (línea 59) con la información acorde al formato de OpenDSS.

Para el caso de las líneas bifásicas de media tensión, se debe verificar con qué fases cuenta la línea para indicarles a los parámetros *Bus1* y *Bus2* los nodos a los cuales debe conectarse los conductores. De forma similar, para las líneas monofásicas de media tensión, se debe verificar la fase para asignar a las barras el nodo correspondiente al cual conectarlas.

```

53 for k in range(len(Lista_Lineas_MT_3F)):
54     Lista_Lineas_MT_3F.iat[k,0]=
        ('New line.'+Lista_Lineas_MT_3F.iat[k,0])
55     Lista_Lineas_MT_3F.iat[k,1]=
        ('bus1='+Lista_Lineas_MT_3F.iat[k,1])
56     Lista_Lineas_MT_3F.iat[k,2]=
        ('bus2='+Lista_Lineas_MT_3F.iat[k,2])
57     Lista_Lineas_MT_3F.iat[k,3]=
        ('LineCode='+Lista_Lineas_MT_3F.iat[k,3].replace(' ','_'))

```

```

58     Lista_Lineas_MT_3F.iat[k,4]=
        ('length='+str(Lista_Lineas_MT_3F.iat[k,4]))
59 Lista_Lineas_MT_3F.to_csv('Lista_Lineas_MT_3F.dss', sep=' ',
        index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
60
61 for k in range(len(Lista_Lineas_MT_2F)):
62     if Lista_Lineas_MT_2F.iloc[k,3][0:6] == comp10 or
        Lista_Lineas_MT_2F.iloc[k,3][0:10] == comp11:
63         Lista_Lineas_MT_2F.iat[k,1]=
            ('bus1='+Lista_Lineas_MT_2F.iat[k,1]+' .1.2')
64         Lista_Lineas_MT_2F.iat[k,2]=
            ('bus2='+Lista_Lineas_MT_2F.iat[k,2]+' .1.2')
65     elif Lista_Lineas_MT_2F.iloc[k,3][0:6] == comp12 or
        Lista_Lineas_MT_2F.iloc[k,3][0:10] == comp12:
66         Lista_Lineas_MT_2F.iat[k,1]=
            ('bus1='+Lista_Lineas_MT_2F.iat[k,1]+' .1.3')
67         Lista_Lineas_MT_2F.iat[k,2]=
            ('bus2='+Lista_Lineas_MT_2F.iat[k,2]+' .1.3')
68     elif Lista_Lineas_MT_2F.iloc[k,3][0:6] == comp14 or
        Lista_Lineas_MT_2F.iloc[k,3][0:10] == comp15:
69         Lista_Lineas_MT_2F.iat[k,1]=
            ('bus1='+Lista_Lineas_MT_2F.iat[k,1]+' .2.3')
70         Lista_Lineas_MT_2F.iat[k,2]=
            ('bus2='+Lista_Lineas_MT_2F.iat[k,2]+' .2.3')
71     Lista_Lineas_MT_2F.iat[k,0]=
        ('New line.'+Lista_Lineas_MT_2F.iat[k,0])
72     Lista_Lineas_MT_2F.iat[k,3]=
        ('LineCode='+Lista_Lineas_MT_2F.iat[k,3].replace(' ','_'))
73     Lista_Lineas_MT_2F.iat[k,4]=
        ('length='+str(Lista_Lineas_MT_2F.iat[k,4]))

```

```

74 Lista_Lineas_MT_2F.to_csv('Lista_Lineas_MT_2F.dss', sep=' ',
    index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
75
76 for k in range(len(Lista_Lineas_MT_1F)):
77     if Lista_Lineas_MT_1F.iloc[k,3][0:5] == comp3 or
        Lista_Lineas_MT_1F.iloc[k,3][0:9] == comp6:
78         Lista_Lineas_MT_1F.iat[k,1]=
            ('bus1='+Lista_Lineas_MT_1F.iat[k,1]+' .1')
79         Lista_Lineas_MT_1F.iat[k,2]=
            ('bus2='+Lista_Lineas_MT_1F.iat[k,2]+' .1')
80     elif Lista_Lineas_MT_1F.iloc[k,3][0:5] == comp4 or
        Lista_Lineas_MT_1F.iloc[k,3][0:9] == comp7:
81         Lista_Lineas_MT_1F.iat[k,1]=
            ('bus1='+Lista_Lineas_MT_1F.iat[k,1]+' .2')
82         Lista_Lineas_MT_1F.iat[k,2]=
            ('bus2='+Lista_Lineas_MT_1F.iat[k,2]+' .2')
83     elif Lista_Lineas_MT_1F.iloc[k,3][0:5] == comp5 or
        Lista_Lineas_MT_1F.iloc[k,3][0:9] == comp8:
84         Lista_Lineas_MT_1F.iat[k,1]=
            ('bus1='+Lista_Lineas_MT_1F.iat[k,1]+' .3')
85         Lista_Lineas_MT_1F.iat[k,2]=
            ('bus2='+Lista_Lineas_MT_1F.iat[k,2]+' .3')
86     Lista_Lineas_MT_1F.iat[k,0]=
        ('New line.'+Lista_Lineas_MT_1F.iat[k,0])
87     Lista_Lineas_MT_1F.iat[k,3]=
        ('LineCode='+Lista_Lineas_MT_1F.iat[k,3].replace(' ','_'))
88     Lista_Lineas_MT_1F.iat[k,4]=
        ('length='+str(Lista_Lineas_MT_1F.iat[k,4]))
89 Lista_Lineas_MT_1F.to_csv('Lista_Lineas_MT_1F.dss', sep=' ',
    index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')

```

```

90
91 for k in range(len(Lista_Lineas_BT)):
92     Lista_Lineas_BT.iat[k,0]=
          ('New line.'+Lista_Lineas_BT.iat[k,0])
93     Lista_Lineas_BT_Servicio.iat[k,1]=
          ('bus1='+Lista_Lineas_BT.iat[k,1])
94     Lista_Lineas_BT_Servicio.iat[k,2]=
          ('bus2='+Lista_Lineas_BT.iat[k,2])
95     Lista_Lineas_BT.iat[k,3]=
          ('LineCode='+Lista_Lineas_BT.iat[k,3].replace(' ','_'))
96     Lista_Lineas_BT.iat[k,4]=
          ('length='+str(Lista_Lineas_BT.iat[k,4]))
97 Lista_Lineas_BT.to_csv('Lista_Lineas_BT.dss', sep=' ',
          index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')

```

3.5. Transformadores

Los transformadores en OpenDSS son elementos de suministro de energía que constan de dos o más devanados, los cuales cuentan siempre con un conductor más al número de fases, que en conexiones estrella corresponde al neutro y en conexiones delta se encuentra abierto internamente (Dugan & Montenegro, 2020b). A continuación, se muestra el código empleado para crear un transformador en OpenDSS:

```

New transformer.200047_T phases=1 windings=3 %noloadloss=0.36 %imag=1.00
~ buses=[MT_AREA_318032_552256.2.0 BT_AREA_318032_552256.1.0
          BT_AREA_318032_552256.0.2]
~ conns=[wye wye wye] kvs=[19.91858428704209 0.12 0.12]
~ kvas=[25.0 25.0 25.0] Xh1=5.65 Xht=5.65
~ Xlt=3.77 %Rs=[0.81 1.62 1.62]

```

Estos elementos comienzan indicando el comando *New Transformer*, seguido de un

indicador para el elemento creado. Posteriormente, se le ingresan los parámetros que se explican a continuación (Dugan & Montenegro, 2020b):

1. *Phases*: Las fases con las que cuenta el transformador.
2. *Windings*: Cantidad de devanados que conforman el transformador.
3. *%Noloadloss*: Porcentaje de pérdida sin carga a tensión nominal. En la simulación este valor se representa como una rama resistiva paralela a la inductancia de magnetización.
4. *%Imag*: Porcentaje de corriente de magnetización. Se representa por medio de una inductancia.
5. *Buses*: Vector que contiene la información de las barras a las que se conecta cada devanado. El orden de los elementos del vector corresponde al orden de los devanados a los cuales se asignan.
6. *Conns*: Vector con la conexión de los devanados. El orden de los elementos del vector corresponden al orden de los devanados.
7. *KVs*: Vector con las tensiones nominales de los devanados. Para transformadores bifásicos y trifásicos, se debe ingresar la tensión de línea-línea. Para otras denominaciones se coloca directamente la tensión nominal. El orden de los elementos del vector corresponden al orden de los devanados.
8. *KVAs*: Vector con las potencias aparentes nominales. El orden de los elementos del vector corresponden al orden de los devanados.
9. *XHL*: Porcentaje de reactancia del devanado de alta tensión al devanado de baja tensión.
10. *XHT*: Porcentaje de reactancia del devanado de alta tensión al devanado terciario. Utilizado únicamente con transformadores de tres devanados.
11. *XLT*: Porcentaje de reactancia del devanado de baja tensión al devanado terciario. Utilizado únicamente con transformadores de tres devanados.

12. *%Rs*: Vector de porcentajes de resistencias de los devanados. El orden de los elementos del vector corresponden al orden de los devanados.

En el caso de los parámetros *%Noloadloss*:, *%Imag*:, *XHL*:, *XHT*:, *XLT*: y *%Rs*:, fueron brindados por Arguello et al. (2018) en forma de diccionarios de Python, los cuales se pueden visualizar en la sección A.2 desde la línea 99 hasta la línea 906. Respecto a la información brindada por el ICE de los transformadores, se cuentan con 1026 transformadores monofásicos de tres devanados, 3 transformadores bifásicos estrella abierta - delta abierta y 6 transformadores en estrella - delta, para un total 1035 transformadores a modelar.

Referente a la minería de datos en los transformadores, solamente se eliminan columnas con información que no es utilizada por OpenDSS. Para lo anterior, se utilizó el programa `LimpiadorExcel.py`, primero, se muestra la información de los transformadores a un *dataframe* (líneas 49 y 52), luego, mediante la función *drop* se eliminan las columnas que no se requieren en OpenDSS (líneas 50 y 53) y, finalmente, se exporta el *dataframe* a un archivo Excel para su posterior uso.

```

49  TrafoAsym=pd.read_excel('CircuitoTFG.xlsx',
    sheet_name='Trafo2WindingAsym', index_col=0, decimal='.')
50  TrafoAsym=TrafoAsym.drop(columns=['Switch1', 'Switch2',
    'IsRegulated', 'CoordX1', 'CoordY1', 'SymbolSize', 'SymbolAngle'])
51
52  Trafo=pd.read_excel('CircuitoTFG.xlsx', sheet_name='Trafo2Winding',
    index_col=0, decimal='.')
53  Trafo=Trafo.drop(columns=['Switch1', 'Switch2', 'IsRegulated',
    'CoordX1', 'CoordY1', 'SymbolSize', 'SymbolAngle'])

```

Además, los transformadores cuentan con un código de librería como el que se muestra en la Figura 3.6. La información que se puede extraer del código de librería es la siguiente:

1. **Término en amarillo:** Indica las fases del transformador. “A”, “B” o “C”

para transformadores monofásicos, “AB”, “BC” o “AC” para transformadores bifásicos y “ABC” para transformadores trifásicos

2. **Término en verde:** Magnitud de la potencia aparente en kVA.
3. **Término en celeste:** Tensión en el devanado de alta tensión en kV.
4. **Término en rosa:** Tensión en los devanados de baja tensión en kV.
5. **Término en naranja:** Indica la configuración del transformador. Solo aplica para transformadores trifásicos.

Figura 3.6

Ejemplo de código de librería de un transformador

ABC_112.5_34.5_kV_240.kV_1_Estrella_Fase_Partida

Nota. Información original del Instituto Costarricense de Electricidad.

3.5.1. Transformadores: Formato de OpenDSS

Para dar formato de acuerdo con OpenDSS a los transformadores, se hizo uso de los modelos que se indican en Laboratorio de Investigación en Potencia y Energía (s.f.-b), los cuales tienen el siguiente formato:

```
//Transformadores trifásicos estrella - delta
~ New transformer.775243_T phases=3 windings=2
~ %noloadloss=0.36 %imag=1.00
~ buses=[MT_AREA_332440_177098.1.2.3.4 BT_AREA_332440_177098.1.2.3]
~ conns=[wye delta] kvs=[34.5 0.24] kvas=[75.0 75.0]
~ Xh1=5.11 %Rs=[0.77 0.77]

//Transformadores bifásicos estrella abierta - delta abierta
```

```

New transformer.775228_T_A phases=1 windings=2
~ %noloadloss=0.34 %imag=1.00
~ buses=[MT_AREA_333655_260457.1.0 BT_AREA_333655_260457.2.1]
~ conns=[wye wye] kvs=[19.91858428704209 0.24]
~ kvas=[50.0 50.0] Xh1=4.91 %Rs=[0.68 0.68]
New transformer.775228_T_B phases=1 windings=2
~ %noloadloss=0.34 %imag=1.00
~ buses=[MT_AREA_333655_260457.2.0 BT_AREA_333655_260457.3.2]
~ conns=[wye wye] kvs=[19.91858428704209 0.24]
~ kvas=[50.0 50.0] Xh1=4.91 %Rs=[0.68 0.68]

//Transformadores monofásicos de tres devanados
New transformer.200047_T phases=1 windings=3
~ %noloadloss=0.36 %imag=1.00
~ buses=[MT_AREA_318032_552256.2.0 BT_AREA_318032_552256.1.0
BT_AREA_318032_552256.0.2]
~ conns=[wye wye wye] kvs=[19.91858428704209 0.12 0.12]
~ kvas=[25.0 25.0 25.0] Xh1=5.65 Xht=5.65
~ Xlt=3.77 %Rs=[0.81 1.62 1.62]

```

Se resalta el caso de los transformadores bifásicos estrella abierta - delta abierta que se representan como dos transformadores de una fase con dos devanados.

Respecto al código empleado, se utilizó el programa `CreadorInputsOpenDSS.py`. En este, primero, se inicia cargando la información de los transformadores en dos *dataframe* llamados *TrafoMonoOrig* y *TrafoBiOrig* (líneas 908-909). Debido a que la información de los transformadores monofásicos y bifásicos se encuentra junta en un solo *dataframe* se separan mediante un ciclo *for* (líneas 913-918). En el caso de transformadores bifásicos se duplican, ya que se requieren dos transformadores para representar un transformador bifásico en OpenDSS.

```
908 TrafoAsymOrig=pd.read_excel('CircuitoLimpio.xlsx',
```

```

        sheet_name='Trafo2WindingAsym', decimal='.')
909 TrafoOrig=pd.read_excel('CircuitoLimpio.xlsx',
        sheet_name='Trafo2Winding', decimal='.')
910
911 TrafoMonoOrig=pd.DataFrame()
912 TrafoBiOrig=pd.DataFrame()
913 for k in range(len(TrafoAsymOrig)):
914     if TrafoAsymOrig.iloc[k,6].startswith('AB') == True or
        TrafoAsymOrig.iloc[k,6].startswith('AC') == True or
        TrafoAsymOrig.iloc[k,6].startswith('BC') == True:
915         TrafoBiOrig = TrafoBiOrig.append(TrafoAsymOrig.iloc[k])
916         TrafoBiOrig = TrafoBiOrig.append(TrafoAsymOrig.iloc[k])
917     else:
918         TrafoMonoOrig = TrafoMonoOrig.append(TrafoAsymOrig.iloc[k])

```

Después, mediante un ciclo *for* (líneas 920-924), se le agrega una de las fases presentes en código de librería a cada uno de los transformadores que componen un transformador bifásico, de forma que no existan dos identificadores iguales, como se muestra a continuación:

```

//Identificador en la información del ICE
775228_T

//Código de librería
AB_50.0_34.5_kV_.240_kV_1

//Identificadores de los transformadores en OpenDSS
que componen un transformador bifásico
775228_T_A
775228_T_B

```

De modo que, si el contador del ciclo es par (línea 921), agrega al identificador la

primera fase del código de librería (línea 922). En caso de no cumplirse la condición, agrega la segunda fase del código de librería al identificador (líneas 923-924).

```

920 for k in range(len(TrafoBiOrig)):
921     if k%2 == 0:
922         TrafoBiOrig.iat[k,0]=
           TrafoBiOrig.iloc[k,0]+'_'+TrafoBiOrig.iloc[k,6][0]
923     else:
924         TrafoBiOrig.iat[k,0]=
           TrafoBiOrig.iloc[k,0]+'_'+TrafoBiOrig.iloc[k,6][1]

```

Luego, se crean *dataframe* vacíos (líneas 926-928) para cada tipo de transformador presente en la red de distribución. a los cuales se les añadirá la información del circuito, de acuerdo con el formato de OpenDSS.

```

926 Trafo=pd.DataFrame(columns=['Name', 'phases', 'windings',
           '%noloadloss', '%imag', 'buses', 'conns', 'kvs', 'kvas',
           'Xh1', '%Rs'], index=range(len(TrafoOrig)))
927 TrafoMono=pd.DataFrame(columns=['Name', 'phases', 'windings',
           '%noloadloss', '%imag', 'buses', 'conns', 'kvs', 'kvas',
           'Xh1', '%Rs'], index=range(len(TrafoMonoOrig)))
928 TrafoBi=pd.DataFrame(columns=['Name', 'phases', 'windings',
           '%noloadloss', '%imag', 'buses', 'conns', 'kvs', 'kvas',
           'Xh1', '%Rs'], index=range(len(TrafoBiOrig)))

```

Para el caso de los transformadores monofásicos de tres devanados, se inicia con el comando *New Transformer*, seguido del identificador correspondiente (línea 931), después, se indica la cantidad de fases (línea 932), así como la cantidad de devanados (línea 933). Luego, se definen los parámetros *%Noloadloss* y *%Imag* llamando el valor correspondiente de los diccionarios *noloadloss_list1F* y *imag_list1F*, acorde a la potencia del transformador (línea 934-935). Luego, se establecen las barras a las que van conectado cada devanado (líneas 936-941), de la siguiente forma:

- Devanado primario: Se conecta a la barra de media tensión. En el caso de los nodos, si es alimentado por la fase “A”, se conecta a los nodos “.1.0”; si es alimentado por la fase “B”, se conecta a los nodos “.2.0”, y si es alimentado por la fase “C”, se conecta a los nodos “.3.0”.
- Devanado secundario: Se conecta a la barra de baja tensión en los nodos “.1.0”.
- Devanado terciario: Se conecta a la barra de baja tensión en los nodos “.0.2”.

Seguidamente, se indica la configuración de los devanados (línea 942). Luego, se establecen las tensiones de cada uno de los devanados (línea 943). Posteriormente, se define la potencia de cada uno de los devanados (línea 944), las cuales varían según el transformador. Después, se indican los parámetros *XHL*, *XHT* y *XLT* llamando el valor correspondiente del diccionario *react_list1F3W* acorde a las tensiones de los devanados y la potencia del transformador (línea 945). Del mismo modo, se establece el parámetro *%Rs* llamando el valor correspondiente del diccionario *res_list1F3W*, de acuerdo con las tensiones de los devanados y la potencia del transformador (línea 946). Finalmente, se exporta la información a un archivo con extensión *.dss* (línea 947).

```

930 for k in range(len(TrafoMonoOrig)):
931     TrafoMono.iat[k,0]=
           ('New transformer.'+TrafoMonoOrig.iloc[k,0])
932     TrafoMono.iat[k,1]='phases=1'
933     TrafoMono.iat[k,2]='windings=3'
934     TrafoMono.iat[k,3]=
           noloadloss_list1F[str(int(TrafoMonoOrig.iloc[k,5]))]
935     TrafoMono.iat[k,4]=
           imag_list1F[str(int(TrafoMonoOrig.iloc[k,5]))]
936     if TrafoMonoOrig.iloc[k,6].startswith('A'):
937         TrafoMono.iat[k,5]='buses=[
           +TrafoMonoOrig.iloc[k,1]+' .1.0 '
           +TrafoMonoOrig.iloc[k,2]+' .1.0 '

```

```

        +TrafoMonoOrig.iloc[k,2]+' .0.2] '
938     if TrafoMonoOrig.iloc[k,6].startswith('B'):
939         TrafoMono.iat[k,5]='buses=['
            +TrafoMonoOrig.iloc[k,1]+' .2.0 '
            +TrafoMonoOrig.iloc[k,2]+' .1.0 '
            +TrafoMonoOrig.iloc[k,2]+' .0.2] '
940     if TrafoMonoOrig.iloc[k,6].startswith('C'):
941         TrafoMono.iat[k,5]='buses=['
            +TrafoMonoOrig.iloc[k,1]+' .3.0 '
            +TrafoMonoOrig.iloc[k,2]+' .1.0 '
            +TrafoMonoOrig.iloc[k,2]+' .0.2] '
942     TrafoMono.iat[k,6]='conns=[wye wye wye] '
943     TrafoMono.iat[k,7]='kvs=['
        +str(TrafoMonoOrig.iloc[k,3]/math.sqrt(3))+
        ' '+str(TrafoMonoOrig.iloc[k,4]/2)
        +' '+str(TrafoMonoOrig.iloc[k,4]/2)+' '']')
944     TrafoMono.iat[k,8]='kvas=['+
        str(TrafoMonoOrig.iloc[k,5])
        +' '+str(TrafoMonoOrig.iloc[k,5])
        +' '+str(TrafoMonoOrig.iloc[k,5])+']')
945     TrafoMono.iat[k,9]=
        react_list1F3W[str(TrafoMonoOrig.iloc[k,3])]
        [str(TrafoMonoOrig.iloc[k,4])]
        [str(int(TrafoMonoOrig.iloc[k,5]))]
946     TrafoMono.iat[k,10]=
        res_list1F3W[str(TrafoMonoOrig.iloc[k,3])]
        [str(TrafoMonoOrig.iloc[k,4])]
        [str(int(TrafoMonoOrig.iloc[k,5]))]
947 TrafoMono.to_csv('Trafo1F.dss', sep=' ', index=False,
        header=False, quoting=csv.QUOTE_NONE, escapechar=' ')

```

Para el caso de los transformadores bifásicos en estrella abierta - delta abierta, se inicia con el comando *New Transformer*, seguido del identificador correspondiente (línea 950), seguidamente, se define la cantidad de fases (línea 951). Además, se establece la cantidad de devanados (línea 952). Luego, se indican los parámetros *%Noloadloss* y *%Imag* llamando el valor correspondiente de los diccionarios *noloadloss_list1F* y *imag_list1F*, acorde a la potencia del transformador (línea 953-954).

Después, se deben establecer las barras a las que van conectado cada devanado (líneas 956-969), lo cual depende del transformador del banco de transformadores y las fases que los alimentan, como se muestra a continuación:

Primer transformador

- Devanado primario: Se conecta a la barra de media tensión correspondiente. En el caso de los nodos, si es alimentado por la fase “A”, se conecta a los nodos “.1.0”; si es alimentado por la fase “B”, se conecta a los nodos “.2.0”, y si es alimentado por la fase “C”, se conecta a los nodos “.3.0”.
- Devanado secundario: Se conecta a la barra de baja tensión en los nodos “.2.1”.

Segundo transformador

- Devanado primario: Se conecta a la barra de media tensión correspondiente. En el caso de los nodos, si es alimentado por la fase “A”, se conecta a los nodos “.1.0”; si es alimentado por la fase “B”, se conecta a los nodos “.2.0”, y si es alimentado por la fase “C”, se conecta a los nodos “.3.0”.
- Devanado secundario: Se conecta a la barra de baja tensión en los nodos “.3.2”.

Luego, se define la configuración de los devanados (línea 970). Después, se indican las tensiones de cada uno de los devanados (línea 971). Posteriormente, se establece la potencia de cada uno de los devanados (línea 972), las cuales varían según el transformador. Seguidamente, se definen los parámetros *XHL* y *%Rs* llamando los valores correspondientes del diccionario *imp_list1F2W*, acorde a las tensiones de los devanados y la potencia del transformador (líneas 973-974). Finalmente, se exporta la información de los transformadores bifásicos a un archivo .dss para su posterior uso (línea 975).

```

949 for k in range(len(TrafoBiOrig)):
950     TrafoBi.iat[k,0]=
           ('New transformer.'+TrafoBiOrig.iloc[k,0])
951     TrafoBi.iat[k,1]='phases=1'
952     TrafoBi.iat[k,2]='windings=2'
953     TrafoBi.iat[k,3]=
           noloadloss_list1F[str(int(TrafoBiOrig.iloc[k,5]))]
954     TrafoBi.iat[k,4]=
           imag_list1F[str(int(TrafoBiOrig.iloc[k,5]))]
956     if k%2 == 0:
957         if TrafoBiOrig.iloc[k,0].endswith('A'):
958             TrafoBi.iat[k,5]='buses=[\'
           +TrafoBiOrig.iloc[k,1]+\' .1.0 \'
           +TrafoBiOrig.iloc[k,2]+\' .2.1\']'
959         if TrafoBiOrig.iloc[k,0].endswith('B'):
960             TrafoBi.iat[k,5]='buses=[\'
           +TrafoBiOrig.iloc[k,1]+\' .2.0 \'
           +TrafoBiOrig.iloc[k,2]+\' .2.1\']'
961         if TrafoBiOrig.iloc[k,0].endswith('C'):
962             TrafoBi.iat[k,5]='buses=[\'
           +TrafoBiOrig.iloc[k,1]+\' .3.0 \'
           +TrafoBiOrig.iloc[k,2]+\' .2.1\']'
963     else:
964         if TrafoBiOrig.iloc[k,0].endswith('A'):
965             TrafoBi.iat[k,5]='buses=[\'
           +TrafoBiOrig.iloc[k,1]+\' .1.0 \'
           +TrafoBiOrig.iloc[k,2]+\' .3.2\']'
966         if TrafoBiOrig.iloc[k,0].endswith('B'):
967             TrafoBi.iat[k,5]='buses=[\'
           +TrafoBiOrig.iloc[k,1]+\' .2.0 \'

```

```

+TrafoBiOrig.iloc[k,2]+' .3.2] '
968     if TrafoBi.iloc[k,0].endswith('C'):
969         TrafoBi.iat[k,5]='buses=[
+TrafoBiOrig.iloc[k,1]+' .3.0 '
+TrafoBiOrig.iloc[k,2]+' .3.2] '
970     TrafoBi.iat[k,6]='conns=[wye wye] '
971     TrafoBi.iat[k,7]='(kvs=[
+str(TrafoBiOrig.iloc[k,3]/math.sqrt(3))
+' '+str(TrafoBiOrig.iloc[k,4])+']')
972     TrafoBi.iat[k,8]='(kvas=[
+str(TrafoBiOrig.iloc[k,5])
+' '+str(TrafoBiOrig.iloc[k,5])+']')
973     TrafoBi.iat[k,9]=
imp_list1F2W[str(TrafoBiOrig.iloc[0,3])]
[str(TrafoBiOrig.iloc[0,4])]
[str(int(TrafoBiOrig.iloc[0,5]))][:8]
974     TrafoBi.iat[k,10]=
imp_list1F2W[str(TrafoBiOrig.iloc[0,3])]
[str(TrafoBiOrig.iloc[0,4])]
[str(int(TrafoBiOrig.iloc[0,5]))][9:]
975 TrafoBi.to_csv('Trafo2F.dss', sep=' ', index=False,
header=False, quoting=csv.QUOTE_NONE, escapechar=' ')

```

Para el caso de los transformadores trifásicos en estrella - delta, se inicia con el comando *New Transformer*, seguido del identificador correspondiente (línea 978), seguidamente, se indica la cantidad de fases (línea 979), así como la cantidad de devanados (línea 980). Luego se definen los parámetros *%Noloadloss* y *%Imag* llamando el valor correspondiente de los diccionarios *noloadloss_list3F* y *imag_list3F* acorde a la potencia del transformador (línea 981-986). Se debe mencionar que, debido a que en los diccionarios no existe información para transformadores de 900 kVA, se utiliza la información para la potencia más cercana correspondiente a 1000 kVA.

Para las barras, el primer devanado se conecta a la barra de media tensión en los nodos “.1.2.3.4” y el segundo devanado a la barra de baja tensión en los nodos “.1.2.3” (línea 987). Luego, se establece la configuración de los devanados (línea 988). Después, se indican las tensiones de cada uno de los devanados (línea 989). Posteriormente, se define la potencia de cada uno de los devanados (línea 990), las cuales varían según el transformador. Seguidamente, se establecen los parámetros *XHL* y *%Rs* llamando los valores correspondientes del diccionario *imp_list3F*, acorde a las tensiones de los devanados y la potencia del transformador (líneas 991-996). Finalmente, se exporta la información de los transformadores trifásicos a un archivo .dss para su posterior uso (línea 997).

```

977 for k in range(len(TrafoOrig)):
978     Trafo.iat[k,0]='New transformer.'+TrafoOrig.iloc[k,0]
979     Trafo.iat[k,1]='phases=3'
980     Trafo.iat[k,2]='windings=2'
981     if TrafoOrig.iloc[k,5] == 900:
982         Trafo.iat[k,3]=noloadloss_list3F['1000']
983         Trafo.iat[k,4]=imag_list3F['1000']
984     else:
985         Trafo.iat[k,3]=
986             noloadloss_list3F[str(int(TrafoOrig.iloc[k,5]))]
987         Trafo.iat[k,4]=
988             imag_list3F[str(int(TrafoOrig.iloc[k,5]))]
989     Trafo.iat[k,5]='buses=['+TrafoOrig.iloc[k,1]+' .1.2.3.4 '
990         +TrafoOrig.iloc[k,2]+' .1.2.3] '
991     Trafo.iat[k,6]='conns=[wye delta] '
992     Trafo.iat[k,7]='kvs=['+str(TrafoOrig.iloc[k,3])+ ' '
993         +str(TrafoOrig.iloc[k,4])+ ']' '
994     Trafo.iat[k,8]='kvas=['+str(TrafoOrig.iloc[k,5])+ ' '
995         +str(TrafoOrig.iloc[k,5])+ ']' '
996     if TrafoOrig.iloc[k,5] == 900:

```

```

992     Trafo.iat[k,9]=
        imp_list3F[str(TrafoOrig.iloc[k,3])]['1000'][:8]
993     Trafo.iat[k,10]=
        imp_list3F[str(TrafoOrig.iloc[k,3])]['1000'][9:]
994     else:
995         Trafo.iat[k,9]=imp_list3F[str(TrafoOrig.iloc[k,3])]
            [str(int(TrafoOrig.iloc[k,5]))[:8]
996         Trafo.iat[k,10]=imp_list3F[str(TrafoOrig.iloc[k,3])]
            [str(int(TrafoOrig.iloc[k,5]))[9:]
997 Trafo.to_csv('Trafo3F.dss', sep=' ', index=False,
            header=False, quoting=csv.QUOTE_NONE, escapechar=' ')

```

3.6. Cargas

En OpenDSS las cargas se representan como elementos de conversión de potencia definidos por su potencia de las tres siguientes formas: potencia real y factor de potencia, potencia real y potencia reactiva o mediante la magnitud de la potencia aparente y el factor de potencia (Dugan & Montenegro, 2020b). Para crear una carga en OpenDSS, se hace uso del siguiente comando:

```

New load.1443075__8 bus1=BT_AREA_317669_656624 kV=0.24 model=1
~ conn=wye kW=1.25 pF=0.95 status=variable phases=2 daily=Perfil

```

Este comienza con *New Load*, seguido del identificador de la carga. Posteriormente, se introducen los siguientes parámetros (Dugan & Montenegro, 2020b):

1. *Bus1*: Corresponde a la barra a la cual se conectará la carga.
2. *kV*: Tensión nominal de la carga en kV. Para cargas bifásicas, trifásicas así como otras cargas en conexión delta se indica la tensión línea - línea, para otras cargas en conexión estrella se indica la tensión línea - neutro.

3. *Model*: Indica la forma en que la carga variará con la tensión. Recibe un entero del 1 al 8. Entre los que se incluyen: modelo 1 de potencia real y potencia reactiva constante, modelo 2 de impedancia constante y modelo 3 de potencia real constante y potencia reactiva cuadrática.
4. *Conn*: Conexión de la carga. Utiliza “wye” para conexiones en estrella o “delta” para conexiones en delta.
5. *kW*: Potencia real nominal de la carga para todas las fases.
6. *Pf*: Factor de potencia nominal de la carga.
7. *Status*: Indica si la carga es modificada por multiplicadores. Utiliza “variable” cuando se aplican los multiplicadores o “fixed” cuando no se aplican.
8. *Phases*: Cantidad de fases de la carga.
9. *Daily*: Nombre del perfil de carga diario. Solo aplica para cargas variables.

Referente al perfil de carga utilizado por las cargas variables, estos se crean en OpenDSS mediante el comando *New Loadshape* como se muestra a continuación:

```
New Loadshape.Perfil npts=144 minterval=10 mult=(file=loadshape_1.txt)
```

Luego del comando, se coloca el identificador del perfil, seguido de los parámetros, los cuales corresponden a (Dugan & Montenegro, 2020b):

1. *Npts*: Cantidad de puntos utilizados para generar la curva.
2. *mInterval*: Intervalo en minutos. Para el caso de horas utilizar *Interval*, y para el caso de segundos utilizar *sInterval*.
3. *Mult*: Vector que contiene los multiplicadores de la carga, del tamaño de *Npts*. Los valores se pueden importar desde un archivo al utilizar el comando *file* e indicando el nombre del archivo con su extensión, este archivo debe localizarse en la misma carpeta.

Continuando con los perfiles de carga, el Laboratorio de Investigación en Potencia y Energía (s.f.-a) brindó 6 de estos perfiles, que van desde una demanda baja hasta una demanda alta.

En cuanto a la información brindada por el ICE, se cuenta con 5805 cargas. Respecto a la minería de datos solo se aplicó una eliminación de columnas con información no requerida por OpenDSS. Para eliminar las columnas, se utilizó el programa `LimpiadorExcel.py`, donde primero se importa la información de las cargas a un dataframe (línea 55), luego, haciendo uso la función `drop`, se eliminan las columnas que no se requieren en OpenDSS (línea 56), después, se reordenan las columnas (línea 57) para que estas coincidan con el orden de los parámetros solicitados por OpenDSS y, finalmente, se exporta el *dataframe* a un archivo Excel para su posterior uso.

```
55 Load=pd.read_excel('CircuitoTFG.xlsx', sheet_name='Load',
    index_col=0, decimal='.')
56 Load=Load.drop(columns=['VelandeK1', 'LfType',
    'CoordX1', 'CoordY1'])
57 Load=Load[['Name', 'Node1', 'Un', 'Tipo', 'Unit', 'E',
    'CosPhi', 'Switch1', 'Phase']]
```

3.6.1. Cargas: Formato de OpenDSS

Para dar formato a las cargas, se utilizó el programa `CreadorInputsOpenDSS.py`, donde primero se importa la información de las cargas a un dataframe llamado *LoadOrig* (línea 999), posteriormente, se crean dos *dataframes* vacíos, el primero para dar formato como cargas fijas y el segundo para dar formato como cargas variables (líneas 1000-1001).

```
999 LoadOrig=pd.read_excel('CircuitoLimpio.xlsx',
    sheet_name='Load', decimal='.')
1000 Load=pd.DataFrame(columns=['Name', 'Bus1', 'kV', 'Model',
    'Conn', 'kW', 'PF', 'Status', 'Phases'],
    index=range(len(LoadOrig)))
```

```

1001     LoadVar=pd.DataFrame(columns=['Name', 'Bus1', 'kV',
    'Model', 'Conn', 'kW', 'PF', 'Status', 'Phases',
    'Daily'],index=range(len(LoadOrig)))

```

Posteriormente, mediante ciclos *for* se da formato a ambos *dataframe*, donde primero se indica el comando *New Load*, seguido del indicador (líneas 1004 y 1016), luego, se define la barra a la que va conectada la carga (líneas 1005 y 1017), seguidamente, se establece la tensión nominal de la carga (líneas 1006 y 1018), después, se establece el parámetro *Model* (líneas 1007 y 1019), posteriormente, se establece la conexión de las cargas (líneas 1008 y 1020).

Luego, se establece la potencia activa de la carga (líneas 1009 y 1021), debido a que el dato suministrado por el ICE corresponde a la energía consumida en un mes en kWh, se divide entre la cantidad de horas en un mes para obtener la potencia real en kW. Considerando un mes de 30 días se tendría un total de 720 horas.

Después, se indica el factor de potencia de las cargas (líneas 1010 y 1022). Seguidamente, se define el parámetro *Status* que en el caso del *dataframe* de cargas fijas corresponde a “fixed” (línea 1011) y en el caso del *dataframe* de cargas variables es “variable” (línea 1023). Luego, se establece el número de fases (líneas 1012 y 1024). Finalmente, para el *dataframe* de cargas variables, se indica el *Loadshape* que se debe utilizar para el modelado de la demanda diaria (línea 1025).

Cuando se les ha dado formato a las cargas, tanto en demanda constante como en demanda variable, se exportan los *dataframe* a archivos con extensión .dss para su posterior uso (líneas 1013 y 1027).

```

1003     for k in range(len(LoadOrig)):
1004         Load.iat[k,0]='New load.'+LoadOrig.iloc[k,0])
1005         Load.iat[k,1]='bus1='+LoadOrig.iloc[k,1])
1006         Load.iat[k,2]='kV='+str(LoadOrig.iloc[k,2])
1007         Load.iat[k,3]='model=1'
1008         Load.iat[k,4]='conn=weye'
1009         Load.iat[k,5]='kW='+str(LoadOrig.iloc[k,5]/720))

```

```

1010     Load.iat[k,6]='pF='+str(LoadOrig.iloc[k,6]))
1011     Load.iat[k,7]='status=fixed'
1012     Load.iat[k,8]='phases=2'
1013 Load.to_csv('Load.dss', sep=' ', index=False, header=False,
             quoting=csv.QUOTE_NONE, escapechar=' ')
1014
1015 for k in range(len(LoadOrig)):
1016     LoadVar.iat[k,0]='New load.'+LoadOrig.iloc[k,0])
1017     LoadVar.iat[k,1]='bus1'+LoadOrig.iloc[k,1])
1018     LoadVar.iat[k,2]='kV'+str(LoadOrig.iloc[k,2]))
1019     LoadVar.iat[k,3]='model=1'
1020     LoadVar.iat[k,4]='conn=wye'
1021     LoadVar.iat[k,5]='kW'+str(LoadOrig.iloc[k,5]/720))
1022     LoadVar.iat[k,6]='pF'+str(LoadOrig.iloc[k,6]))
1023     LoadVar.iat[k,7]='status=variable'
1024     LoadVar.iat[k,8]='phases=2'
1025     LoadVar.iat[k,9]='daily=Perfil'
1026
1027 LoadVar.to_csv('LoadVar.dss', sep=' ', index=False,
             header=False, quoting=csv.QUOTE_NONE, escapechar=' ')

```

3.7. Regulador de tensión

Respecto a la información suministrada por el ICE, la red de distribución solo cuenta con un regulador de tensión tipo A, por lo que su implementación se realizó de forma manual en OpenDSS, sin hacer uso de Python. Para el modelado del regulador de tensión se utilizó como referencia el documento *Estudios de Capacidad de Alojamiento de DER en Circuitos de Distribución, Modelado y Simulación de Reguladores en Línea y Capacitores* de la Autoridad Reguladora de los Servicios Públicos (2023), ajustando parámetros para que coincidieran acorde a los datos de la red de distribución asignada.

De forma que se obtuvo el siguiente código:

```
//Jumpers de entrada
New Reactor.Jumper_Reg_E1 phases=1 bus1=MT_AREA_341077_802343.1
~ bus2=MT_AREA_341077_802343_1.1 X=0.0001 R=0.0001
New Reactor.Jumper_Reg_E2 phases=1 bus1=MT_AREA_341077_802343.2
~ bus2=MT_AREA_341077_802343_2.1 X=0.0001 R=0.0001
New Reactor.Jumper_Reg_E3 phases=1 bus1=MT_AREA_341077_802343.3
~ bus2=MT_AREA_341077_802343_3.1 X=0.0001 R=0.0001

//Unidades transformadoras
New Transformer.Reg_1 phases=1 windings=2
~ bank=reg0 xhl=0.01 %loadloss=0.1
~ wdg=1 Bus=MT_AREA_341077_802343_1.1.0 kV=19.9186 kVA=90.90
~ wdg=2 Bus=MT_AREA_341077_802343_1.2.1 kV=1.99 kVA=90.90
~ Maxtap=1.0 Mintap=-1.0 tap=0 numtaps=32

New Transformer.Reg_2 phases=1 windings=2
~ bank=reg0 xhl=0.01 %loadloss=0.1
~ wdg=1 Bus=MT_AREA_341077_802343_2.1.0 kV=19.9186 kVA=90.90
~ wdg=2 Bus=MT_AREA_341077_802343_2.2.1 kV=1.99 kVA=90.90
~ Maxtap=1.0 Mintap=-1.0 tap=0 numtaps=32

New Transformer.Reg_3 phases=1 windings=2
~ bank=reg0 xhl=0.01 %loadloss=0.1
~ wdg=1 Bus=MT_AREA_341077_802343_3.1.0 kV=19.9186 kVA=90.90
~ wdg=2 Bus=MT_AREA_341077_802343_3.2.1 kV=1.99 kVA=90.90
~ Maxtap=1.0 Mintap=-1.0 tap=0 numtaps=32

//Jumpers de salida
New Reactor.Jumper_Reg_S1 phases=1 bus1=MT_AREA_341077_802343_1.2
```

```

~ bus2=MT_AREA_341076_739692.1 X=0.0001 R=0.0001
New Reactor.Jumper_Reg_S2 phases=1 bus1=MT_AREA_341077_802343_2.2
~ bus2=MT_AREA_341076_739692.2 X=0.0001 R=0.0001
New Reactor.Jumper_Reg_S3 phases=1 bus1=MT_AREA_341077_802343_3.2
~ bus2=MT_AREA_341076_739692.3 X=0.0001 R=0.0001

//Controladores del regulador
New regcontrol.RegCont_1 transformer=Reg_1 winding=2
~ bus=MT_AREA_341077_802343_1.2 vreg=120.0 band=3.0
~ ptratio=166.0 maxtapchange=1
New regcontrol.RegCont_2 transformer=Reg_2 winding=2
~ bus=MT_AREA_341077_802343_2.2 vreg=120.0 band=3.0
~ ptratio=166.0 maxtapchange=1
New regcontrol.RegCont_3 transformer=Reg_3 winding=2
~ bus=MT_AREA_341077_802343_3.2 vreg=120.0 band=3.0
~ ptratio=166.0 maxtapchange=1

//Monitores del transformador
New monitor.MonReg_1 Element=transformer.Reg_1 Terminal=1 Mode=1
New monitor.MonReg_2 Element=transformer.Reg_2 Terminal=1 Mode=1
New monitor.MonReg_3 Element=transformer.Reg_3 Terminal=1 Mode=1

//Se crea un medidor de energia en la alimentacion
New energymeter.alimentacion Line.1247375__0

```

Primero, haciendo uso de *jumpers* de baja impedancia, que se crean con el comando *New Reactor*, se conecta la barra de entrada a tres sub-barras, una para cada fase. Luego, se realiza el mismo procedimiento para la barra de salida. Después, se crean tres transformadores monofásicos conectados en medio de las sub-barras, a estos transformadores se les colocan los reguladores de tensión generados con el comando *New RegControl*, los cuales, con el parámetro *Winding*, se les indica a cual devanado miden

y controlan la tensión, con *Vreg* se establece la tensión de configuración del regulador en V , con *Band* se define el ancho de banda del regulador en V , con *Ptatio* se indica la razón de transformación del transformador de potencial y, finalmente, con *Maxtapchange* se define el cambio máximo de paso por iteración de control (Dugan & Montenegro, 2020b). Por último, se crean equipos de medición mediante los comandos *New Monitor* y *New EnergyMeter*, los cuales se utilizarán para medir la tensión respecto a la distancia desde la fuente de alimentación.

3.8. OpenDSS-G

Para ingresar la información de la red de distribución a OpenDSS, se debe importar el archivo maestro al programa, para esto se hace uso de la herramienta “OpenDSS to OpenDSS-G Translator”, la cual se puede acceder mediante los botones mostrados en la Figura 3.7.

Esto desplegará la interfaz de la herramienta como la que se muestra en la Figura 3.8, donde, al pulsar el botón de la carpeta, se desplegará una pantalla con la cual seleccionar el archivo maestro a importar. Con la ruta del archivo maestro seleccionada, se puede generar una vista previa con el botón “Preview”, la cual se visualizará en el recuadro de la derecha, además, esta vista se puede modificar con las tres opciones de “Graphical Options”, finalmente, al pulsar “Start”, se importará el archivo maestro a OpenDSS-G, donde se solicitará indicar una dirección en la cual guardar los datos importados. También se debe mencionar que en la ventana “Translation Status” se muestran mensajes referentes al proceso de importación.

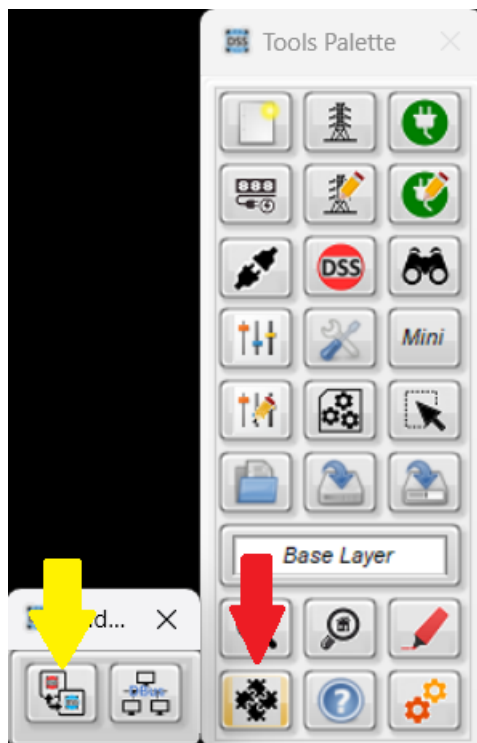
Luego de guardado los datos, en OpenDSS-G se desplegará el circuito en su interfaz y es posible iniciar el modo de simulación con el botón de engranajes de la esquina inferior derecha de la interfaz “Tools Pallete”, como se muestra en la Figura 3.9.

Luego de iniciado el modo de simulación, el menú “Tools Pallete” se actualiza desplegando nuevos botones como se muestra en la Figura 3.10. Empezando desde izquierda a derecha y desde arriba a abajo, las nuevas opciones corresponden a (Dugan & Montenegro, 2020a):

1. *Run Simulation*: Se ejecuta la simulación.
2. *Graphics*: Permite generar gráficos sobre información relevante del circuito.
3. *GIS options*: Brinda acceso a la herramienta de GIS.
4. *Program control*: Permite compilar el circuito así como establecer el tiempo de la simulación.
5. *Reports*: Genera reportes de tensión, corriente, potencia y pérdidas.
6. *Automation*: Brinda acceso a las herramientas de simulación automatizada.
7. *Generate Faults*: Permite generar fallas en el modelo para sus respectivos análisis.
8. *OpenDSS Command*: Permite ingresar comandos de OpenDSS.
9. *Find Element*: Permite ubicar y centrar elementos del circuito.
10. *Analysis*: Brinda acceso a un menú con diferentes estudios y modelos análisis de la red.
11. *Configuration*: Accede a la configuración del programa.
12. *Mini Visualizer*: Vuelve a abrir el menú “Circuit Navigator” en caso de cerrarse.
13. *Select Ops*: Abre un menú con las opciones de selección de los elementos
14. *Zoom*: Despliega un menú con las opciones de zoom.
15. *Inspector*: Permite seleccionar una sección del circuito y visualizar las líneas, los elementos de suministro de potencia o los elementos de consumo de potencia.
16. *Heats Maps*: Despliega las opciones disponibles de mapas de calor.
17. *Meters*: Muestra opciones referentes a la visualización de los elementos de medición.
18. *OpenDSS Help*: Permite acceder a las opciones de ayuda de OpenDSS.
19. *Enable Constructor Mode*: Cambia al modo construcción de circuitos.

Figura 3.7

Botones para acceder a la herramienta “OpenDSS to OpenDSS-G Translator”



Nota. Primero se debe pulsar el botón señalado por la flecha roja y posteriormente el botón señalado por la flecha amarilla.

Figura 3.8

Interfaz de la herramienta “OpenDSS to OpenDSS-G Translator”

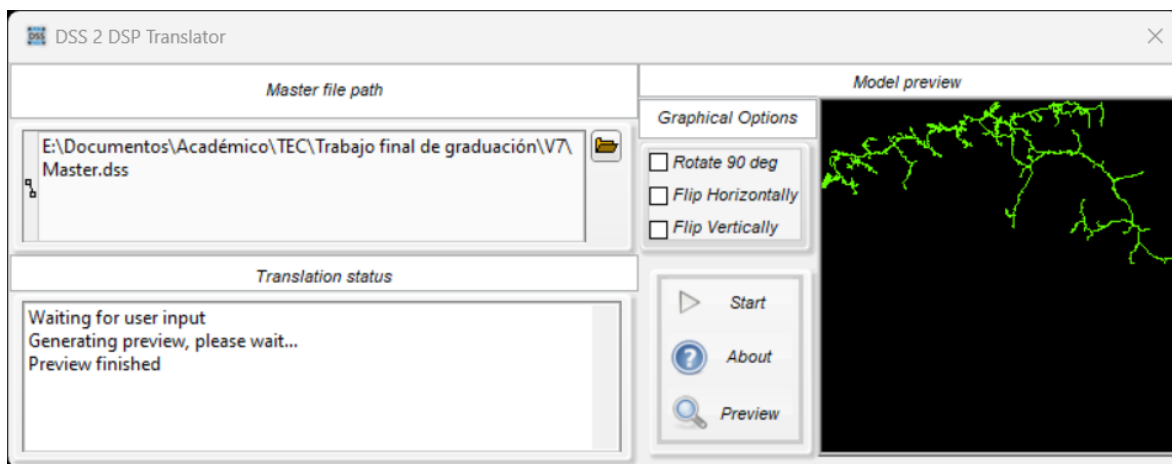


Figura 3.9

Visualización de la red de distribución en OpenDSS-G

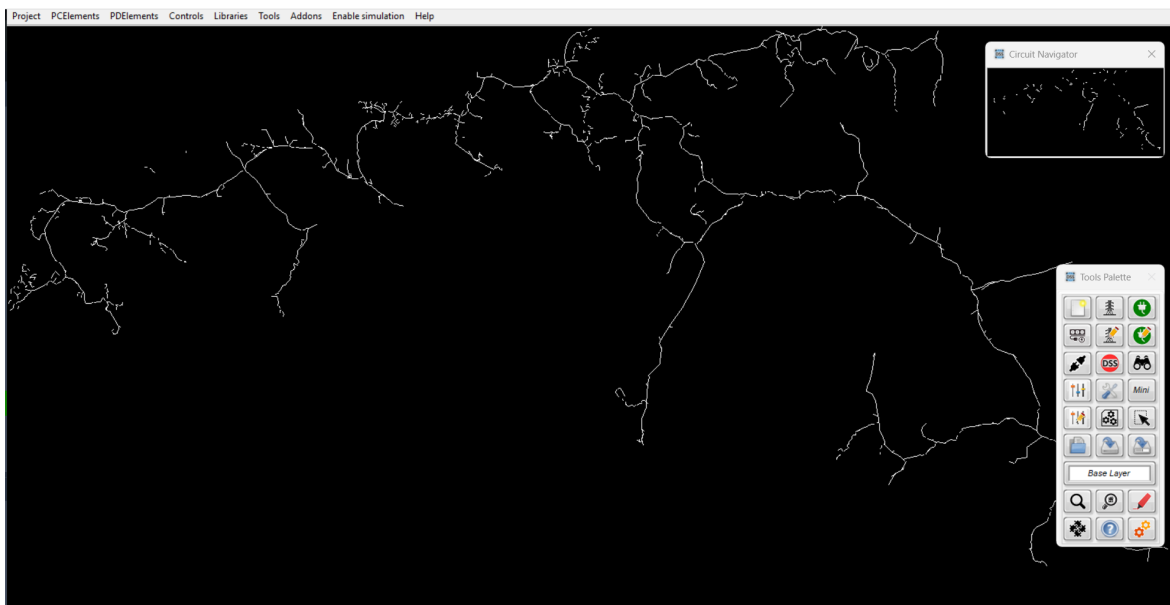


Figura 3.10

“Tools Pallette” en modo simulación



Capítulo 4

Resultados de investigación

Con la implementación de los datos de la red tanto en OpenDSS como en OpenDSS-G, es posible empezar a extraer información del circuito en forma de reportes, gráficos y mapas de calor, mientras se consideran distintos perfiles de demanda. Lo anterior con el fin de observar el comportamiento de la red de distribución ante diferentes situaciones, de forma que se genere información útil sobre esta, comprendiendo su estado actual, su capacidad de crecimiento, la posibilidad de implementación de generación distribuida, entre otros aspectos.

4.1. Reportes

Tanto OpenDSS como OpenDSS-G generan reportes sobre las potencias, corrientes, tensiones y pérdidas de la red. Respecto a los reportes de potencia, estos se pueden generar en kVA o MVA, mostrando solamente la potencia en las terminales de los elementos (Figura 4.1) o incluyendo también la potencia en los nodos de las barras (Figura 4.2). En los reportes de pérdidas, se pueden visualizar las pérdidas de cada línea y transformador, así como un resumen de estos datos al final del reporte (Figura 4.8). En el caso de la tensión, se pueden visualizar los valores de tensión en las barras (Figura 4.3), en los nodos de las barras (Figura 4.4) y en los elementos que componen el circuito (Figura 4.5). En el caso de las corrientes, se pueden generar dos reportes, uno por terminales (Figura 4.6) y el otro por los nodos de los buses (Figura 4.7).

Por otra parte, OpenDSS despliega un resumen de los datos de los circuitos donde se puede apreciar los tiempos de simulación, la tensión máxima y mínima en p.u., la potencia real y reactiva, así como sus pérdidas, tanto en magnitud como en valor relativo al total de potencia. En la Figura 4.9 se aprecian los resúmenes para cargas fijas y para los 6 perfiles de carga utilizados. Se puede apreciar en las cargas variables que, conforme aumenta la demanda, disminuyen las pérdidas activas relativas al total de potencia, hasta alcanzar un valor mínimo que en este caso corresponde al perfil 3; posteriormente, las pérdidas relativas empiezan a aumentar conforme aumenta la demanda. Además, se observa que, conforme aumenta la demanda, disminuyen las pérdidas de potencia reactiva hasta que las pérdidas reactivas cambian de signo y vuelven a aumentar.

Además, se debe mencionar que OpenDSS incluye reportes de elementos aislados o desconectados de la red, reportes de elementos en anillos, reporte de elementos por tipo, así como un diagrama de la topología de la red de distribución. Estos reportes se pueden utilizar para asegurar que la implementación de los datos al *software* se está realizando de forma correcta.

Por último, la importancia de estos reportes radica en el hecho de poder consultar la información específica de uno o de varios elementos, así como visualizar el comportamiento de alguna variable en todos los elementos o barras del circuito, de forma que se puedan detectar comportamientos anómalos que en los gráficos podrían perderse.

4.2. Gráficos

En cuanto a los gráficos, mediante los medidores, se pueden graficar los perfiles de tensión de las fases, como se aprecia en las Figuras 4.10, 4.11, 4.12 y 4.13. Se puede observar que, conforme más se aleja de la subestación, la tensión disminuye; además, se observa el efecto del regulador aproximadamente a la distancia de 37 km y este vuelve a elevar la tensión de las fases. También, se observa que una de las fases experimenta una mayor caída de tensión, posiblemente debido al hecho de que sea la fase más cargada del circuito. Además, esta fase tanto en el caso de carga fija, así como en el caso de

demanda alta presenta una caída de tensión fuera de los niveles aceptables, de acuerdo con el programa. Por último, cabe mencionar que, conforme aumenta la carga, mayor es la caída de tensión, debido a la distancia desde la alimentación.

También es posible graficar el comportamiento de la tensión, corriente y potencias de las fases a lo largo de un periodo de tiempo, con el fin de observar el comportamiento de estas variables acorde a la demanda. En las Figuras 4.14, 4.15, 4.16 y 4.17 se muestran el comportamiento a lo largo de un día de la tensión, corriente, potencia real y potencia reactiva de la fuente de alimentación, respectivamente. En el caso de la tensión, esta tiene un comportamiento inverso a la potencia, teniendo un valor mínimo cuando la potencia alcanza un valor pico. Por otra parte, se observa que la potencia demandada por las fases no es balanceada, lo cual concordaría con lo dicho por Kersting y Kerestes (2023), que las redes de distribución en su gran mayoría son circuitos desbalanceados. Además, es posible visualizar las tensiones, corrientes y potencias de un elemento específico, como se visualiza en la Figura 4.18, de forma que sea más sencilla la interpretación de los valores a comparación de los reportes.

Finalmente, la importancia de los gráficos radica en la capacidad de condensar la información del circuito de manera gráfica, pudiendo interpretar el compartimiento completo de la red de distribución de forma rápida a comparación de un reporte.

4.3. Mapas de calor

OpenDSS-G es capaz de crear mapas de calor de la red de distribución, de forma que se puedan observar características y el comportamiento de la red en un espacio geográfico. Entre los mapas de calor que se pueden generar se incluyen:

- **Infraestructura sobrecargada:** Colorea los elementos de suministro de potencia acorde a la sobrecarga que experimentan, siendo rojo los elementos sobrecargados y azul los elementos que no presentan sobrecarga. En la Figura 4.19 se observa el mapa de sobrecargas de la red de distribución asignada, donde todos los elementos se encuentran dentro del rango aceptable.

- Nivel de tensión:** Este mapa indica mediante colores el nivel de tensión en un rango de 1,05 p.u. a 9,5 p.u. para los elementos de suministro de potencia. En la Figura 4.20 se observa el mapa de nivel de tensión para el caso asignado, donde la mayoría de elementos se encuentra en celeste, correspondiente a un valor entre un 1 y 0,95 p.u., además de ciertas áreas en verde correspondientes a un valor aproximado de 1 p.u.
- Flujo de potencia:** Con este es mapa, es posible visualizar la dirección del flujo de potencia mediante flechas. En la Figura 4.21 se observa la dirección del flujo de potencia en una porción del circuito cercana a la fuente de alimentación, donde se puede apreciar la dirección radial del flujo de potencia.
- Elementos aislados:** Con este mapa, se indica qué elementos se encuentran energizados coloreándolos de rojo, así como los elementos que se encuentran desconectados coloreándolos de verde. En la Figura 4.22 se observa que los elementos se encuentran conectados a la fuente de alimentación, las pequeñas zonas en amarillo y verde corresponden a problemas de coloreado del programa, ya que, mediante un reporte de elementos aislados, el cual se visualiza en la Figura 4.23, se verifica que todos los elementos se encuentran conectados al circuito.

Figura 4.1

Reporte de potencias en las terminales de los elementos de la red

SYMMETRICAL COMPONENT POWERS BY CIRCUIT ELEMENT (first 3 phases)								Excess Power			
Element	Term	P1(kw)	Q1(kvar)	P2	Q2	P0	Q0	P_Norm	Q_Norm	P_Emerg	Q_Emerg
"Vsource.SOURCE"	1	-2076.7	40.5	0.2	0.9	-0.2	-0.5				
"Vsource.SOURCE"	2	0.0	0.0	0.0	0.0	0.0	0.0				
"Line.83623_0"	1	1420.5	103.2	-0.9	-3.7	1.5	-2.9	0.0	0.0	0.0	0.0
"Line.83623_0"	2	-1420.4	-113.2	0.9	3.7	-1.5	2.9				
"Line.83623_1"	1	1420.4	113.2	-0.9	-3.7	1.5	-2.9	0.0	0.0	0.0	0.0
"Line.83623_1"	2	-1420.4	-122.6	0.9	3.7	-1.5	2.9				
"Line.83623_2"	1	1420.4	122.6	-0.9	-3.7	1.5	-2.9	0.0	0.0	0.0	0.0
"Line.83623_2"	2	-1420.4	-138.1	0.9	3.7	-1.5	2.9				
"Line.134028_0"	1	15.7	-33.8	-0.0	-0.0	-0.0	-0.0	0.0	0.0	0.0	0.0
"Line.134028_0"	2	-15.7	27.4	0.0	0.0	0.0	0.0				
"Line.134028_1"	1	15.7	-27.4	-0.0	-0.0	-0.0	-0.0	0.0	0.0	0.0	0.0
"Line.134028_1"	2	-15.7	22.9	0.0	0.0	0.0	0.0				

Figura 4.2

Reporte de potencias en los nodos de las barras

CIRCUIT ELEMENT POWER FLOW						
(Power Flow into element from indicated Bus)						
Power Delivery Elements						
Bus	Phase	kW	+j	kvar	kVA	PF
ELEMENT = "Vsource.SOURCE"						
MT_AREA_361240_501964	1	-1032.2	+j	-110.4	1038.1	0.9943
MT_AREA_361240_501964	2	-460.5	+j	30.5	461.5	-0.9978
MT_AREA_361240_501964	3	-583.9	+j	120.7	596.3	-0.9793
TERMINAL TOTAL		-2076.7	+j	40.9	2077.1	-0.9998
MT_AREA_361240_501964	0	0.0	+j	0.0	0.0	1.0000
MT_AREA_361240_501964	0	0.0	+j	0.0	0.0	1.0000
MT_AREA_361240_501964	0	0.0	+j	0.0	0.0	1.0000
TERMINAL TOTAL		0.0	+j	0.0	0.0	1.0000
ELEMENT = "Line.83623_0"						
MT_AREA_337180_89974	1	703.5	+j	96.3	710.1	0.9908
MT_AREA_337180_89974	2	408.8	+j	45.8	411.4	0.9938
MT_AREA_337180_89974	3	308.8	+j	-45.4	312.1	-0.9893
TERMINAL TOTAL		1421.1	+j	96.7	1424.4	0.9977
MT_SUB_337115_989955	1	-703.5	+j	-99.6	710.5	0.9901
MT_SUB_337115_989955	2	-408.8	+j	-49.1	411.8	0.9929
MT_SUB_337115_989955	3	-308.8	+j	42.1	311.6	-0.9908
TERMINAL TOTAL		-1421.1	+j	-106.6	1425.1	0.9972
ELEMENT = "Line.83623_1"						
MT_SUB_337115_989955	1	703.5	+j	99.6	710.5	0.9901
MT_SUB_337115_989955	2	408.8	+j	49.1	411.8	0.9929
MT_SUB_337115_989955	3	308.8	+j	-42.1	311.6	-0.9908
TERMINAL TOTAL		1421.1	+j	106.6	1425.1	0.9972
MT_SUB_337047_882161	1	-703.5	+j	-102.7	710.9	0.9895
MT_SUB_337047_882161	2	-408.8	+j	-52.3	412.1	0.9919
MT_SUB_337047_882161	3	-308.8	+j	39.0	311.2	-0.9921
TERMINAL TOTAL		-1421.0	+j	-116.0	1425.8	0.9967

Figura 4.3

Reportes de tensión línea - línea y línea - neutro en las barras

SYMMETRICAL COMPONENT PHASE-PHASE VOLTAGES BY BUS (for 3-phase buses)							
Bus	Mag:	V1 (kV)	p.u.	V2 (kV)	%V2/V1	V0 (kV)	%V0/V1
mt_area_361240_501964	34.46	0.999	0.04941	0.1434	4.898E-015	1.421E-014	
mt_area_337180_89974	34.69	1.006	0.3081	0.888	1.819E-015	5.243E-015	
mt_sub_337115_989955	34.69	1.006	0.3083	0.8885	4.067E-015	1.172E-014	
mt_sub_337047_882161	34.69	1.006	0.3084	0.889	4.067E-015	1.172E-014	
mt_area_336969_205437	34.69	1.006	0.3087	0.8898	3.279E-015	9.453E-015	
mt_area_349830_565934	34.77	1.008	0.2945	0.8472	3.638E-015	1.046E-014	
mt_sub_349908_285582	34.77	1.008	0.2945	0.8472	3.638E-015	1.046E-014	
mt_area_349964_281652	34.77	1.008	0.2945	0.8472	3.75E-015	1.079E-014	
mt_area_349969_131709	34.77	1.008	0.2945	0.8472	2.034E-015	5.85E-015	
mt_area_349998_870329	34.77	1.008	0.2945	0.8472	4.067E-015	1.17E-014	
mt_area_322678_480299	34.49	0.9997	0.4039	1.171	3.279E-015	9.507E-015	
mt_sub_322678_746038	34.49	0.9997	0.4039	1.171	2.572E-015	7.458E-015	
mt_sub_322682_812912	34.49	0.9997	0.404	1.171	3.279E-015	9.507E-015	
mt_sub_322687_520638	34.49	0.9997	0.404	1.171	2.572E-015	7.458E-015	
mt_sub_322692_854732	34.49	0.9997	0.404	1.171	4.067E-015	1.179E-014	
mt_sub_322698_798782	34.49	0.9997	0.404	1.171	3.279E-015	9.507E-015	
mt_sub_322705_334501	34.49	0.9997	0.404	1.171	3.279E-015	9.507E-015	
SYMMETRICAL COMPONENT VOLTAGES BY BUS (for 3-phase buses)							
Bus	Mag:	V1 (kV)	p.u.	V2 (kV)	%V2/V1	V0 (kV)	%V0/V1
mt_area_361240_501964	19.9	0.999	0.02853	0.1434	0.02609	0.1311	
mt_area_337180_89974	20.03	1.006	0.1779	0.888	0.1943	0.9699	
mt_sub_337115_989955	20.03	1.006	0.178	0.8885	0.1943	0.9699	
mt_sub_337047_882161	20.03	1.006	0.1781	0.889	0.1943	0.9698	
mt_area_336969_205437	20.03	1.006	0.1782	0.8898	0.1942	0.9698	
mt_area_349830_565934	20.07	1.008	0.17	0.8472	0.1895	0.9443	
mt_sub_349908_285582	20.07	1.008	0.17	0.8472	0.1895	0.9443	
mt_area_349964_281652	20.07	1.008	0.17	0.8472	0.1895	0.9443	
mt_area_349969_131709	20.07	1.008	0.17	0.8472	0.1895	0.9443	
mt_area_349998_870329	20.07	1.008	0.17	0.8472	0.1895	0.9443	
mt_area_322678_480299	19.91	0.9997	0.2332	1.171	0.278	1.396	
mt_sub_322678_746038	19.91	0.9997	0.2332	1.171	0.278	1.396	
mt_sub_322682_812912	19.91	0.9997	0.2332	1.171	0.278	1.396	
mt_sub_322687_520638	19.91	0.9997	0.2332	1.171	0.278	1.396	
mt_sub_322692_854732	19.91	0.9997	0.2332	1.171	0.278	1.396	
mt_sub_322698_798782	19.91	0.9997	0.2333	1.171	0.278	1.396	
mt_sub_322705_334501	19.91	0.9997	0.2333	1.171	0.278	1.396	

Nota. El reporte de arriba corresponde a tensiones línea - línea y el reporte de abajo a tensiones línea - neutro.

Figura 4.4

Reporte de tensión en los nodos de las barras

LINE-GROUND and LINE-LINE VOLTAGES BY BUS & NODE									
Bus	Node	VLN (kV)	Angle	pu	Base kV	Node-Node	VLL (kV)	Angle	pu
MT_AREA_361240_501964 .	1	19.892 /_	-0.3	0.99868	34.500	1-2	34.416 /_	29.8	0.99757
-	2	19.863 /_	-120.2	0.99719	34.500	2-3	34.497 /_	-90.2	0.9999
-	3	19.94 /_	119.7	1.0011	34.500	3-1	34.481 /_	149.7	0.99946
-	4	1.9301 /_	-34.9	0.0969	34.500				
-	5	0.87823 /_	-80.8	0.044091	34.500				
-	6	2.7851 /_	132.2	0.13982	34.500				
MT_AREA_337180_89974 ..	1	19.953 /_	-2.8	1.0017	34.500	1-2	34.394 /_	28.1	0.99694
-	2	20.091 /_	-121.2	1.0087	34.500	2-3	34.784 /_	-91.3	1.0082
-	3	20.053 /_	118.7	1.0068	34.500	3-1	34.906 /_	147.9	1.0118
MT_SUB_337115_989955 ..	1	19.952 /_	-2.8	1.0017	34.500	1-2	34.393 /_	28.1	0.99691
-	2	20.091 /_	-121.2	1.0086	34.500	2-3	34.784 /_	-91.3	1.0082
-	3	20.053 /_	118.7	1.0067	34.500	3-1	34.905 /_	147.9	1.0117
MT_SUB_337047_882161 ..	1	19.952 /_	-2.8	1.0017	34.500	1-2	34.393 /_	28.1	0.99688
-	2	20.09 /_	-121.2	1.0086	34.500	2-3	34.783 /_	-91.3	1.0082
-	3	20.053 /_	118.7	1.0067	34.500	3-1	34.905 /_	147.9	1.0117
MT_AREA_336969_205437 .	1	19.951 /_	-2.8	1.0016	34.500	1-2	34.391 /_	28.1	0.99684
-	2	20.09 /_	-121.2	1.0086	34.500	2-3	34.782 /_	-91.3	1.0082
-	3	20.052 /_	118.7	1.0067	34.500	3-1	34.903 /_	147.9	1.0117
MT_AREA_349830_565934 .	1	20.025 /_	-2.7	1.0054	34.500	1-2	34.485 /_	28.2	0.99956
-	2	20.117 /_	-121.1	1.01	34.500	2-3	34.828 /_	-91.2	1.0095
-	3	20.077 /_	118.8	1.008	34.500	3-1	34.984 /_	148.0	1.014
MT_SUB_349908_285582 ..	1	20.025 /_	-2.7	1.0054	34.500	1-2	34.485 /_	28.2	0.99956
-	2	20.117 /_	-121.1	1.01	34.500	2-3	34.828 /_	-91.2	1.0095
-	3	20.077 /_	118.8	1.008	34.500	3-1	34.984 /_	148.0	1.014

Figura 4.5*Reporte de tensión en los elementos de la red*

NODE-GROUND VOLTAGES BY CIRCUIT ELEMENT				
Power Delivery Elements				
Bus	(node ref)	Phase	Magnitude, kV (pu)	Angle
ELEMENT = "Vsource.SOURCE"				
MT_AREA_361240_501964	(1)	1	19.892 (0.9987) /_	-0.3
MT_AREA_361240_501964	(2)	2	19.863 (0.9972) /_	-120.2
MT_AREA_361240_501964	(3)	3	19.94 (1.001) /_	119.7

MT_AREA_361240_501964	(0)	0	0 (0) /_	0.0
MT_AREA_361240_501964	(0)	0	0 (0) /_	0.0
MT_AREA_361240_501964	(0)	0	0 (0) /_	0.0
ELEMENT = "Line.83623__0"				
MT_AREA_337180_89974	(4)	1	19.953 (1.002) /_	-2.8
MT_AREA_337180_89974	(5)	2	20.091 (1.009) /_	-121.2
MT_AREA_337180_89974	(6)	3	20.053 (1.007) /_	118.7

MT_SUB_337115_989955	(7)	1	19.952 (1.002) /_	-2.8
MT_SUB_337115_989955	(8)	2	20.091 (1.009) /_	-121.2
MT_SUB_337115_989955	(9)	3	20.053 (1.007) /_	118.7
ELEMENT = "Line.83623__1"				
MT_SUB_337115_989955	(7)	1	19.952 (1.002) /_	-2.8
MT_SUB_337115_989955	(8)	2	20.091 (1.009) /_	-121.2
MT_SUB_337115_989955	(9)	3	20.053 (1.007) /_	118.7

MT_SUB_337047_882161	(10)	1	19.952 (1.002) /_	-2.8
MT_SUB_337047_882161	(11)	2	20.09 (1.009) /_	-121.2
MT_SUB_337047_882161	(12)	3	20.053 (1.007) /_	118.7

Figura 4.6

Reporte de corriente en las terminales de los elementos de la red

SYMMETRICAL COMPONENT CURRENTS BY CIRCUIT ELEMENT (first 3 phases)								
Element	Term	I1	I2	%I2/I1	I0	%I0/I1	%Normal	%Emergency
"VSOURCE.SOURCE"	1	34.796	10.977	31.55	7.3917	21.24	0.00	0.00
-	2	34.796	10.977	31.55	7.3917	21.24	0.00	0.00
"LINE.83623__0"	1	23.7	7.0664	29.82	5.5695	23.50	8.90	5.93
-	2	23.713	7.065	29.79	5.568	23.48	0.00	0.00
"LINE.83623__1"	1	23.713	7.065	29.79	5.568	23.48	8.90	5.94
-	2	23.726	7.0636	29.77	5.5667	23.46	0.00	0.00
"LINE.83623__2"	1	23.726	7.0636	29.77	5.5667	23.46	8.91	5.94
-	2	23.75	7.0614	29.73	5.5645	23.43	0.00	0.00
"LINE.134028__0"	1	0.61878	0.023208	3.75	0.055701	9.00	0.17	0.11
-	2	0.52427	0.022312	4.26	0.05512	10.51	0.00	0.00
"LINE.134028__1"	1	0.52427	0.022312	4.26	0.05512	10.51	0.15	0.10
-	2	0.46172	0.02169	4.70	0.054724	11.85	0.00	0.00
"LINE.134028__3"	1	0.46188	0.021954	4.75	0.057994	12.56	0.13	0.09
-	2	0.45659	0.0219	4.80	0.057958	12.69	0.00	0.00
"LINE.134028__5"	1	0.45746	0.021439	4.69	0.059112	12.92	0.13	0.09
-	2	0.42563	0.02111	4.96	0.058889	13.84	0.00	0.00
"LINE.136034__0"	1	3.3833	1.4362	42.45	0.54719	16.17	1.32	0.88
-	2	3.371	1.4361	42.60	0.54721	16.23	0.00	0.00
"LINE.136034__1"	1	3.371	1.4361	42.60	0.54721	16.23	1.31	0.87
-	2	3.3555	1.4359	42.79	0.54725	16.31	0.00	0.00
"LINE.136034__2"	1	3.3555	1.4359	42.79	0.54725	16.31	1.31	0.87
-	2	3.3401	1.4357	42.98	0.54728	16.39	0.00	0.00
"LINE.136034__3"	1	3.3401	1.4357	42.98	0.54728	16.39	1.30	0.87
-	2	3.3246	1.4355	43.18	0.54732	16.46	0.00	0.00
"LINE.136034__4"	1	3.3246	1.4355	43.18	0.54732	16.46	1.30	0.87
-	2	3.3091	1.4353	43.38	0.54735	16.54	0.00	0.00
"LINE.136034__5"	1	3.3091	1.4353	43.38	0.54735	16.54	1.30	0.86
-	2	3.2937	1.4352	43.57	0.54739	16.62	0.00	0.00

Figura 4.7

Reporte de corriente en los nodos de las barras

```

CIRCUIT ELEMENT CURRENTS
(Currents into element from indicated bus)

Power Delivery Elements

```

Bus	Phase	Magnitude, A	Angle	(Real)	+j	(Imag)
ELEMENT = "Vsource.SOURCE"						
MT_AREA_361240_501964	1	52.187 /_	173.6 =	-51.866	+j	5.7762
MT_AREA_361240_501964	2	23.237 /_	63.6 =	10.326	+j	20.817
MT_AREA_361240_501964	3	29.904 /_	-48.7 =	19.756	+j	-22.448

MT_AREA_361240_501964	0	52.187 /_	-6.4 =	51.866	+j	-5.7762
MT_AREA_361240_501964	0	23.237 /_	-116.4 =	-10.326	+j	-20.817
MT_AREA_361240_501964	0	29.904 /_	131.3 =	-19.756	+j	22.448
ELEMENT = "Line.83623_0"						
MT_AREA_337180_89974	1	35.588 /_	-10.6 =	34.981	+j	-6.5475
MT_AREA_337180_89974	2	20.476 /_	-127.6 =	-12.488	+j	-16.226
MT_AREA_337180_89974	3	15.563 /_	127.1 =	-9.3811	+j	12.418
MT_AREA_337180_89974	Resid	16.708 /_	141.7 =	-13.112	+j	10.356

MT_SUB_337115_989955	1	35.611 /_	169.1 =	-34.973	+j	6.7129
MT_SUB_337115_989955	2	20.495 /_	52.0 =	12.631	+j	16.14
MT_SUB_337115_989955	3	15.539 /_	-53.5 =	9.2351	+j	-12.497
MT_SUB_337115_989955	Resid	16.704 /_	-38.3 =	13.107	+j	-10.356
ELEMENT = "Line.83623_1"						
MT_SUB_337115_989955	1	35.611 /_	-10.9 =	34.973	+j	-6.7129
MT_SUB_337115_989955	2	20.495 /_	-128.0 =	-12.631	+j	-16.14
MT_SUB_337115_989955	3	15.539 /_	126.5 =	-9.2351	+j	12.497
MT_SUB_337115_989955	Resid	16.704 /_	141.7 =	-13.107	+j	10.356

MT_SUB_337047_882161	1	35.634 /_	168.9 =	-34.965	+j	6.8688
MT_SUB_337047_882161	2	20.514 /_	51.5 =	12.765	+j	16.059
MT_SUB_337047_882161	3	15.519 /_	-54.1 =	9.0975	+j	-12.573
MT_SUB_337047_882161	Resid	16.7 /_	-38.3 =	13.102	+j	-10.355

Figura 4.8

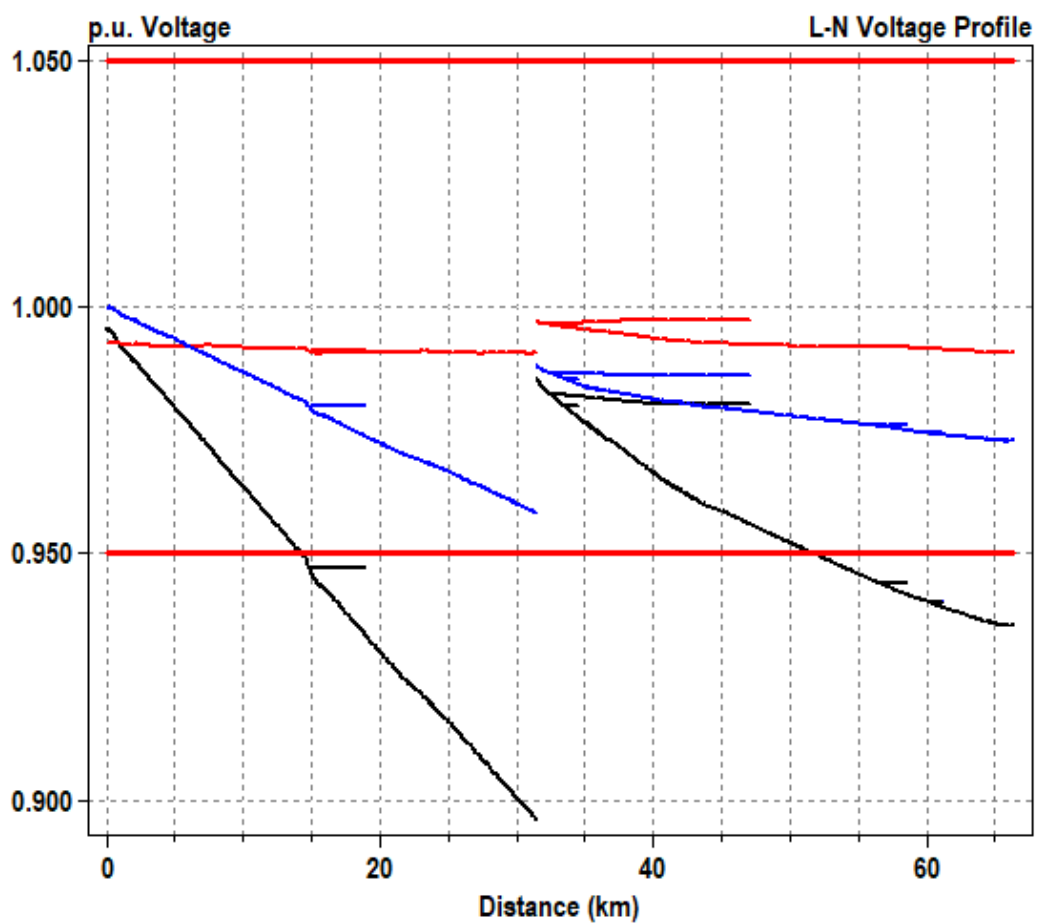
Reporte de pérdidas en las líneas y transformadores

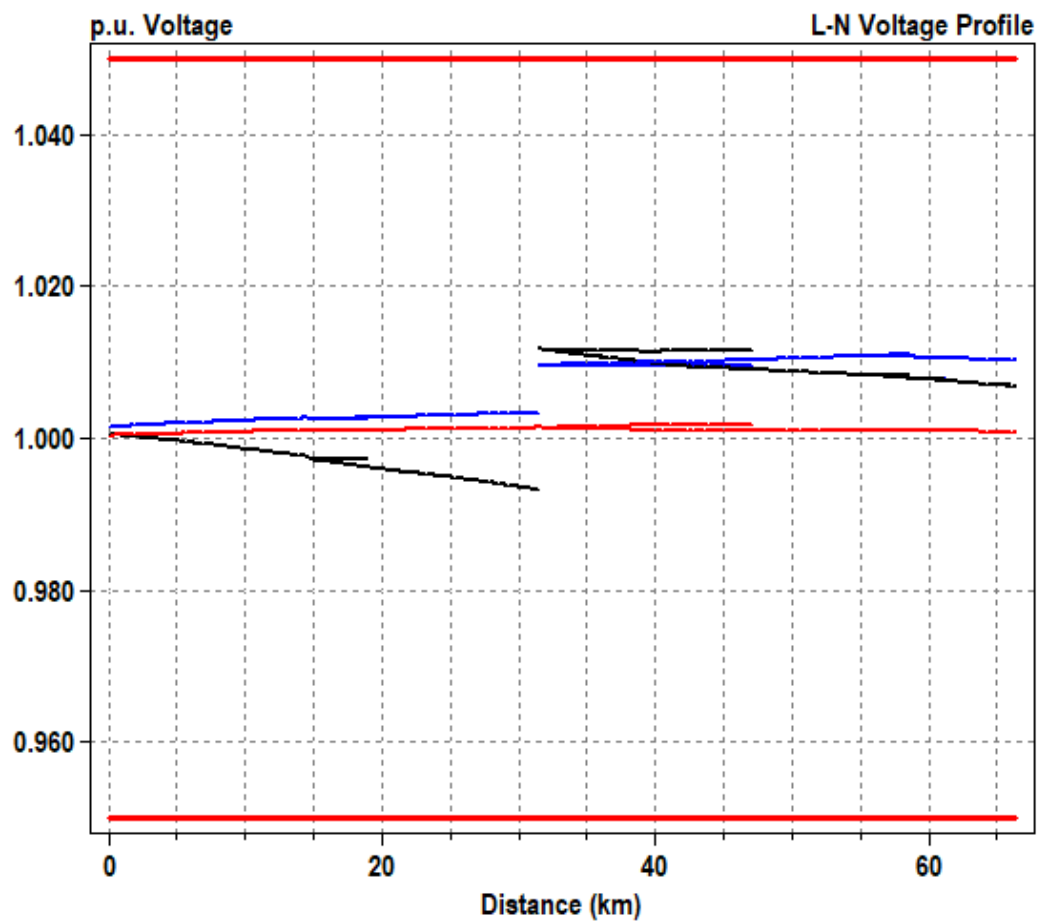
LOSSES REPORT			
Power Delivery Element Loss Report			
Element	kW Losses	% of Power	kvar Losses
"Line.83623__0"	0.03181,	0.00	-9.9648
"Line.83623__1"	0.03001,	0.00	-9.39271
"Line.83623__2"	0.04967,	0.00	-15.5284
"Line.134028__0"	0.00004,	0.0	-6.40382
"Line.134028__1"	0.00002,	0.0	-4.443
"Line.134028__3"	0.00000,	0.0	-0.384827
"Line.134028__5"	0.00001,	0.0	-2.35867
"Line.136034__0"	0.00019,	0.0	-0.743488
"Line.136034__1"	0.00023,	0.0	-0.937436
"Line.136034__2"	0.00023,	0.0	-0.93744
"Line.136034__3"	0.00023,	0.0	-0.937442
"Line.136034__4"	0.00023,	0.0	-0.937446
"Line.136034__5"	0.00023,	0.0	-0.937448
"Line.136034__6"	0.00023,	0.0	-0.937451
"Line.136034__7"	0.00051,	0.0	-2.15861
"Line.136034__8"	0.00050,	0.0	-2.15862
"Line.136034__9"	0.00049,	0.0	-2.15864
"Line.136034__10"	0.00017,	0.0	-0.737431
"Line.136034__11"	0.00017,	0.0	-0.737432
"Transformer.REG_1"	0.03239,	0.06	0.0161923
"Transformer.REG_2"	0.00033,	0.0	0.01377
"Transformer.REG_3"	0.00211,	0.02	0.0137122
"Reactor.JUMPER_REG_S1"	0.00023,	0.0	0.000234992
"Reactor.JUMPER_REG_S2"	0.00005,	0.0	4.83222E-005
"Reactor.JUMPER_REG_S3"	0.00008,	0.0	7.75008E-005
LINE LOSSES=	75.3 kW		
TRANSFORMER LOSSES=	112.9 kW		
TOTAL LOSSES=	188.2 kW		
TOTAL LOAD POWER =	1888.5 kW		
Percent Losses for Circuit =	9.96 %		

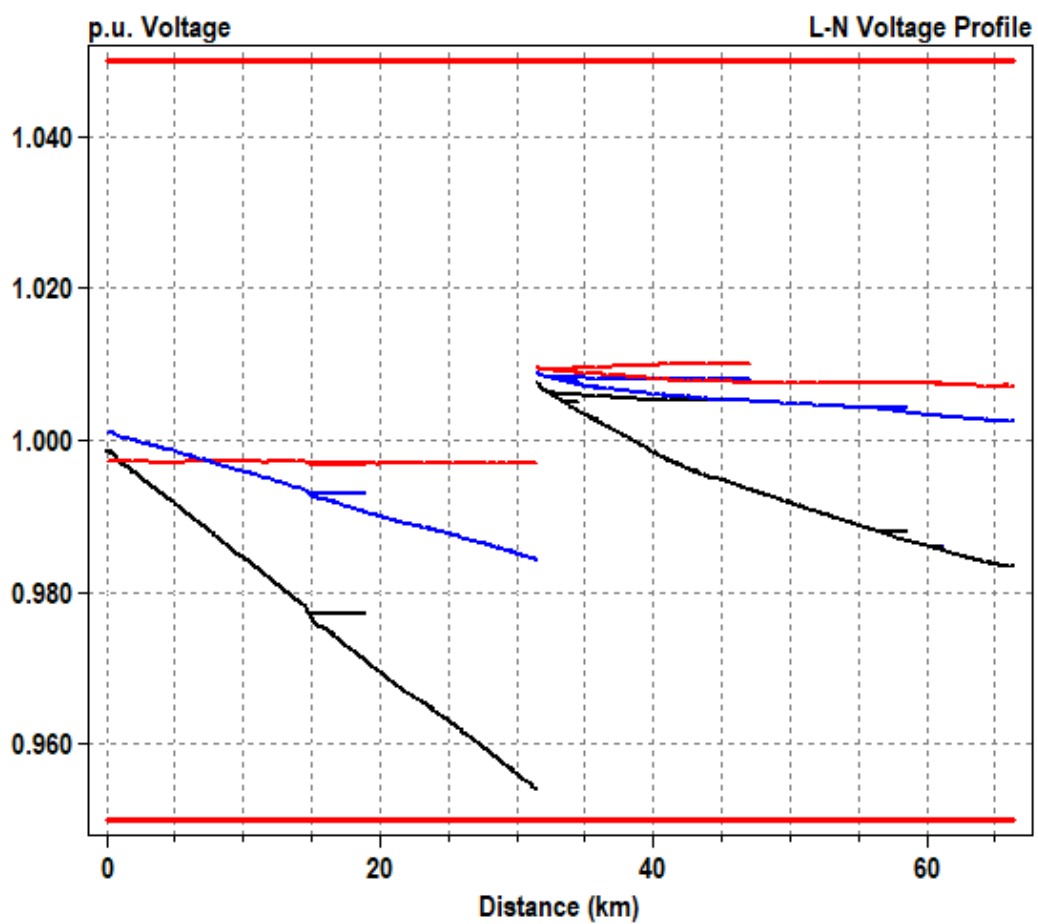
Figura 4.9

Resumen de OpenDSS para los diferente perfiles de carga

Carga fija	Perfil 1	Perfil 2	Perfil 3
<p>- Circuit Summary -</p> <p>Year = 0 Hour = 0 Max pu. voltage = 1.4819 Min pu. voltage = 0.13963 Total Active Power: 3.94208 MW Total Reactive Power: 0.809842 Mvar Total Active Losses: 0.412039 MW, (10.45 %) Total Reactive Losses: -0.350427 Mvar Frequency = 60 Hz Mode = Snap Control Mode = STATIC Load Model = PowerFlow</p>	<p>- Circuit Summary -</p> <p>Year = 0 Hour = 24 Max pu. voltage = 1.5103 Min pu. voltage = 0.13985 Total Active Power: 0.705508 MW Total Reactive Power: -0.564318 Mvar Total Active Losses: 0.119245 MW, (16.9 %) Total Reactive Losses: -0.757 Mvar Frequency = 60 Hz Mode = Daily Control Mode = STATIC Load Model = PowerFlow</p>	<p>- Circuit Summary -</p> <p>Year = 0 Hour = 24 Max pu. voltage = 1.5063 Min pu. voltage = 0.13984 Total Active Power: 1.2587 MW Total Reactive Power: -0.358463 Mvar Total Active Losses: 0.136924 MW, (10.88 %) Total Reactive Losses: -0.727167 Mvar Frequency = 60 Hz Mode = Daily Control Mode = STATIC Load Model = PowerFlow</p>	<p>- Circuit Summary -</p> <p>Year = 0 Hour = 24 Max pu. voltage = 1.5122 Min pu. voltage = 0.13981 Total Active Power: 2.07667 MW Total Reactive Power: -0.0408927 Mvar Total Active Losses: 0.183151 MW, (9.06 %) Total Reactive Losses: -0.661621 Mvar Frequency = 60 Hz Mode = Daily Control Mode = STATIC Load Model = PowerFlow</p>
<p>- Circuit Summary -</p> <p>Year = 0 Hour = 24 Max pu. voltage = 1.5086 Min pu. voltage = 0.13959 Total Active Power: 4.1667 MW Total Reactive Power: 0.897384 Mvar Total Active Losses: 0.443061 MW, (10.63 %) Total Reactive Losses: -0.326505 Mvar Frequency = 60 Hz Mode = Daily Control Mode = STATIC Load Model = PowerFlow</p>	<p>- Circuit Summary -</p> <p>Year = 0 Hour = 24 Max pu. voltage = 1.4993 Min pu. voltage = 0.13952 Total Active Power: 4.64565 MW Total Reactive Power: 1.14254 Mvar Total Active Losses: 0.526429 MW, (11.33 %) Total Reactive Losses: -0.211377 Mvar Frequency = 60 Hz Mode = Daily Control Mode = STATIC Load Model = PowerFlow</p>	<p>- Circuit Summary -</p> <p>Year = 0 Hour = 24 Max pu. voltage = 1.4896 Min pu. voltage = 0.13911 Total Active Power: 6.63511 MW Total Reactive Power: 2.21663 Mvar Total Active Losses: 0.962758 MW, (14.51 %) Total Reactive Losses: 0.352204 Mvar Frequency = 60 Hz Mode = Daily Control Mode = STATIC Load Model = PowerFlow</p>	

*Figura 4.10**Perfil de tensión con cargas constantes*

*Figura 4.11**Perfil de tensión con demanda baja (LoadShape_1)*

*Figura 4.12**Perfil de tensión con demanda media (LoadShape_3)*

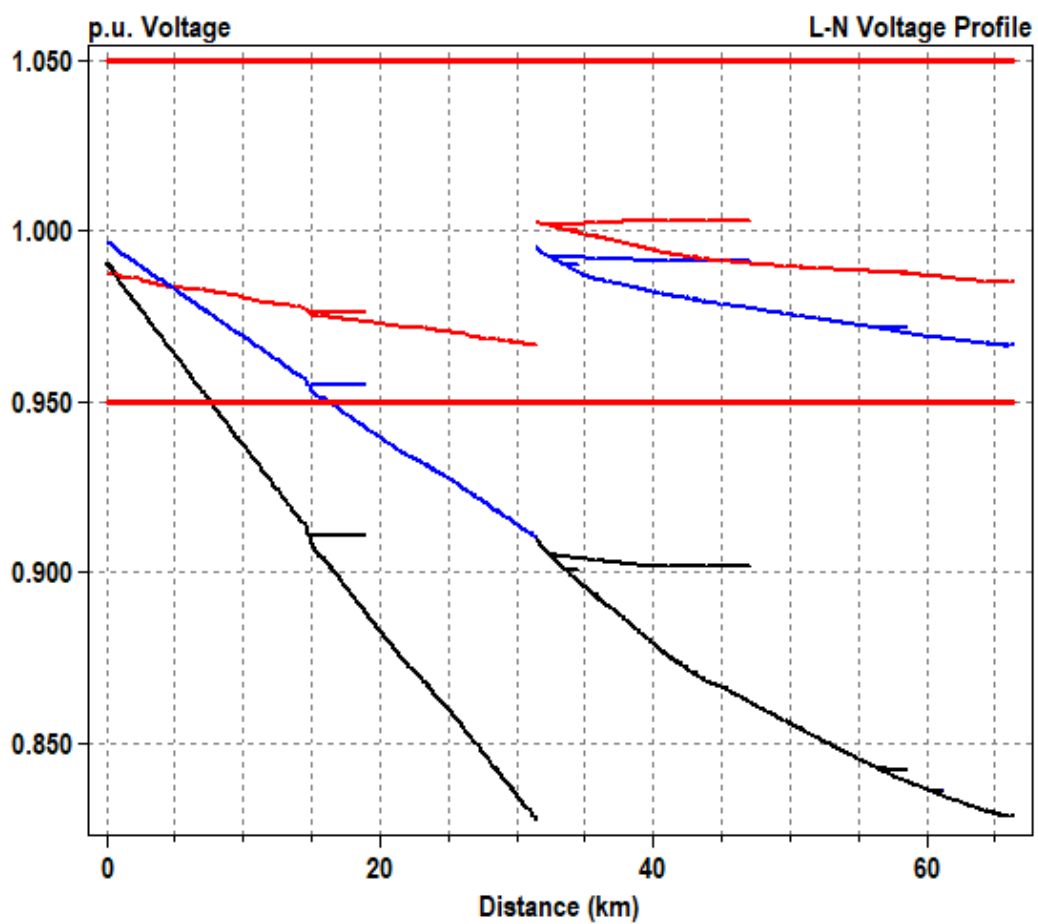
*Figura 4.13**Perfil de tensión con demanda alta (LoadShape_6)*

Figura 4.14

Tensión en la fuente de alimentación a lo largo de un día considerando una demanda media (LoadShape_3)

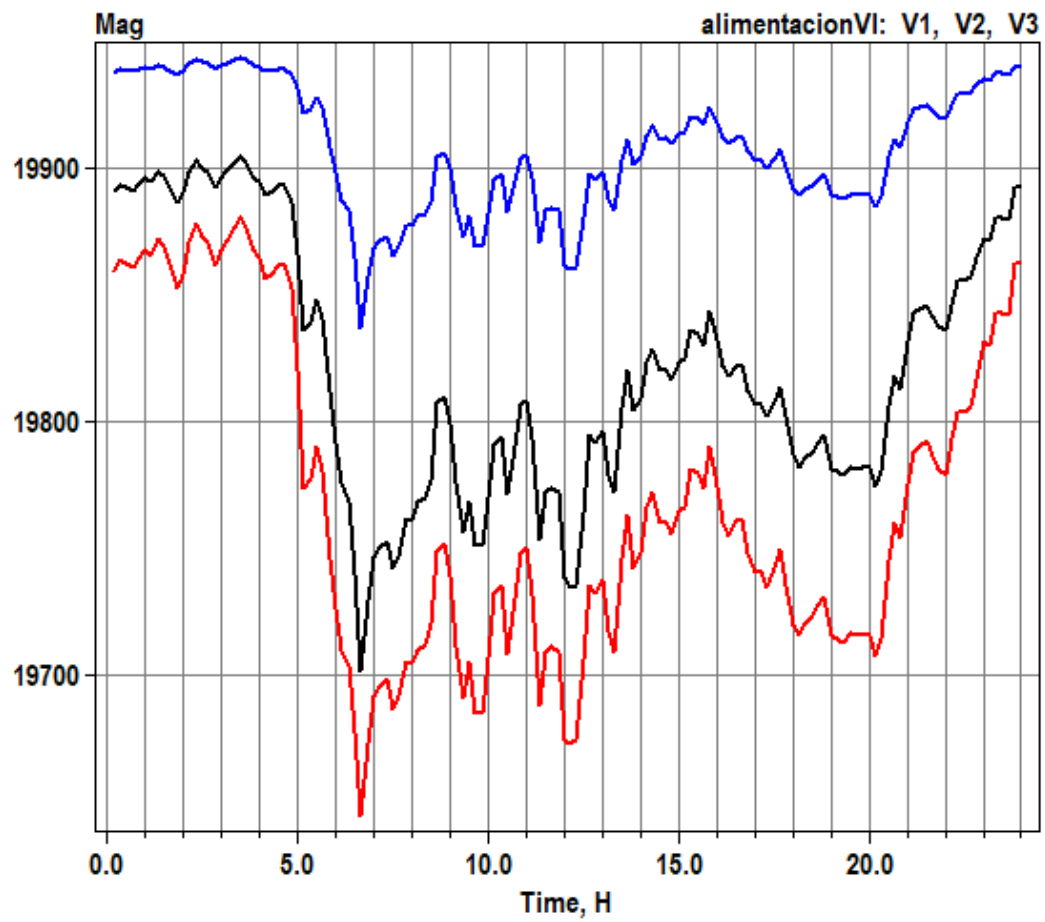


Figura 4.15

Corriente en la fuente de alimentación a lo largo de un día considerando una demanda media (LoadShape_3)

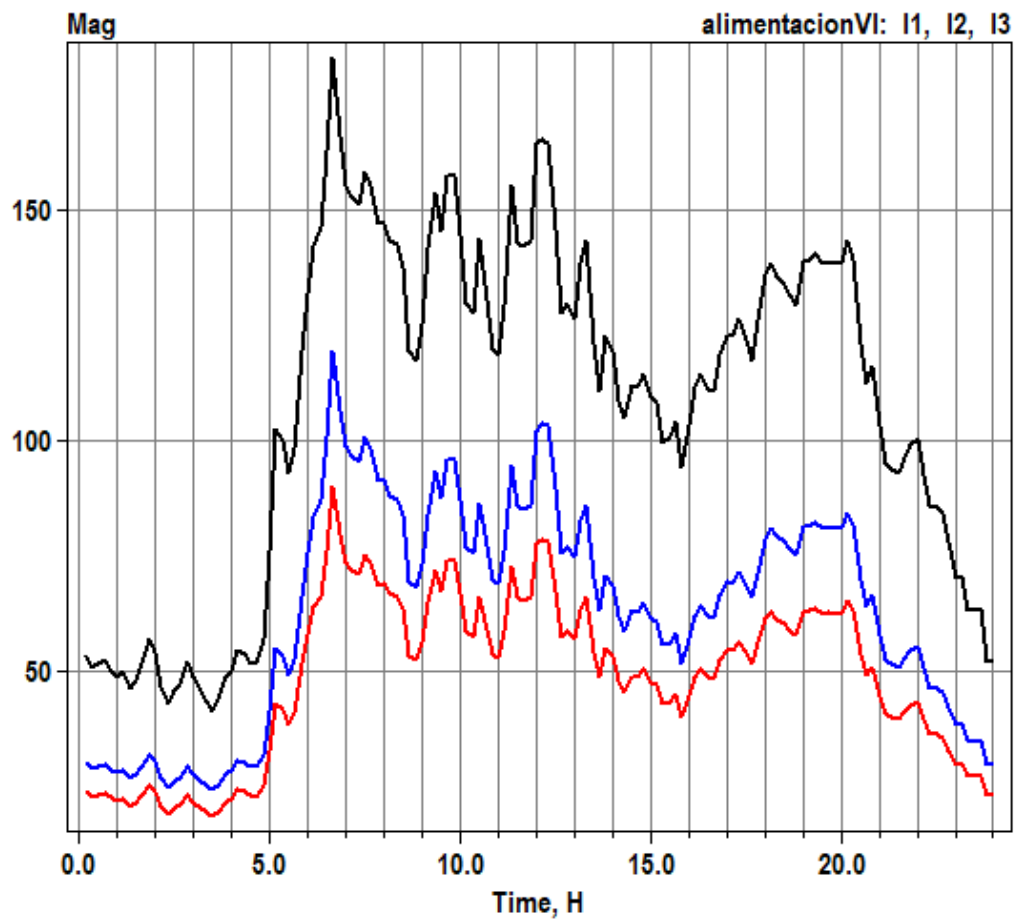


Figura 4.16

Potencia real en la fuente de alimentación a lo largo de un día considerando una demanda media (LoadShape_3)

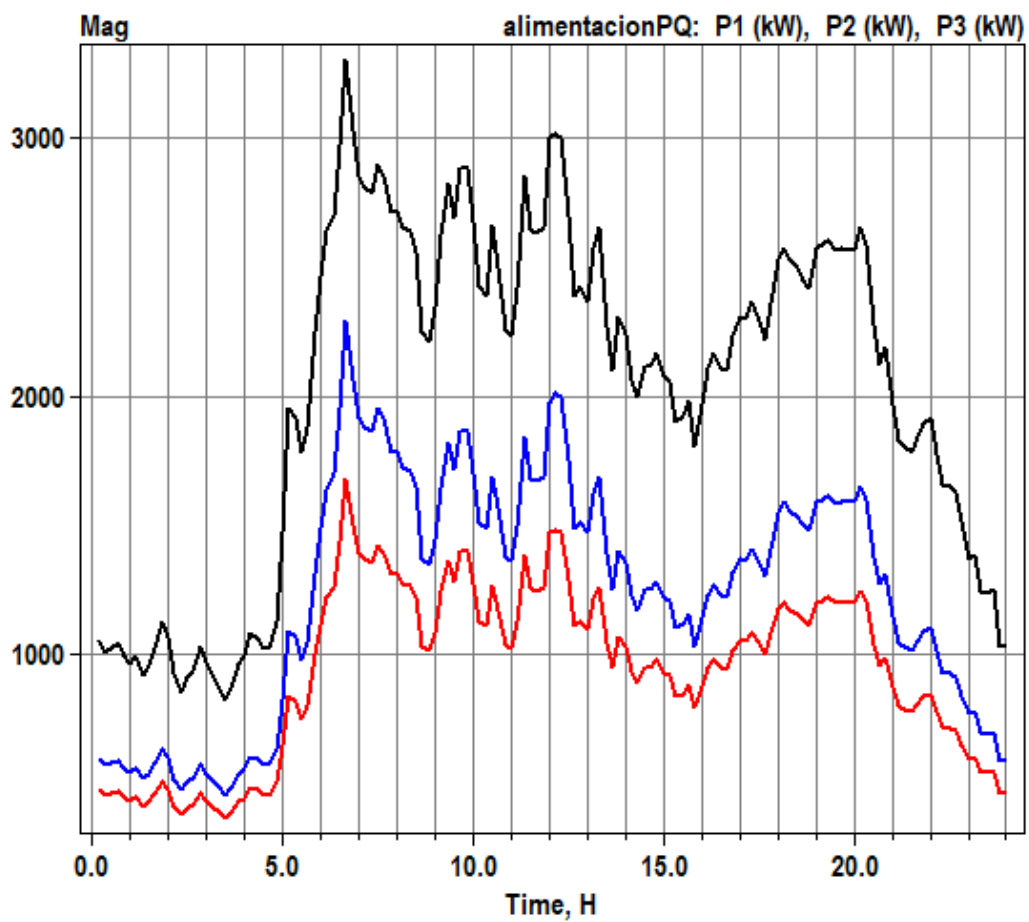


Figura 4.17

Potencia reactiva en la fuente de alimentación a lo largo de un día considerando una demanda media (LoadShape_3)

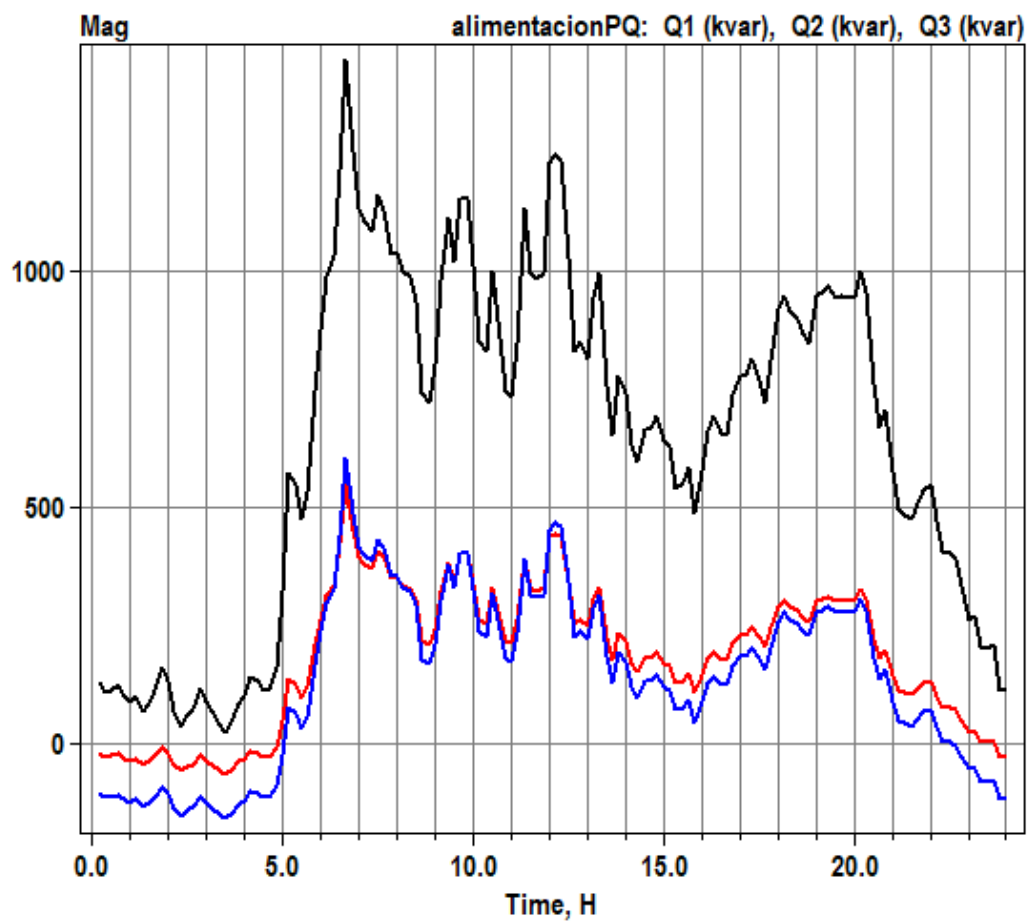


Figura 4.18

Visualización de potencias de un elemento en OpenDSS

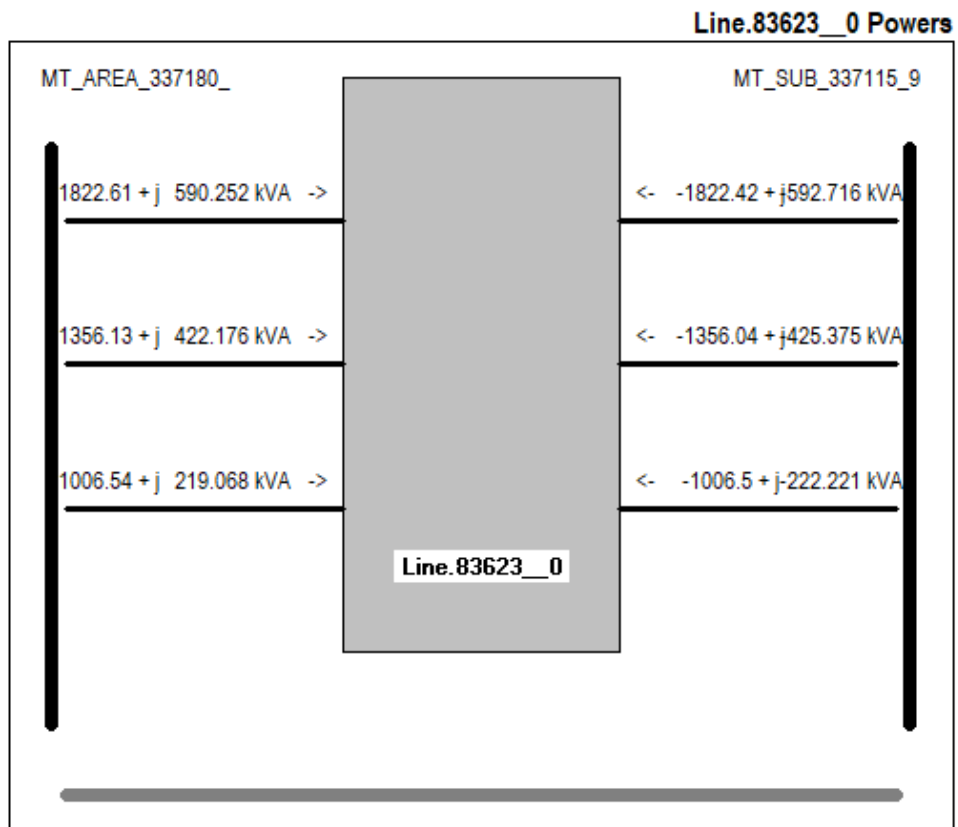
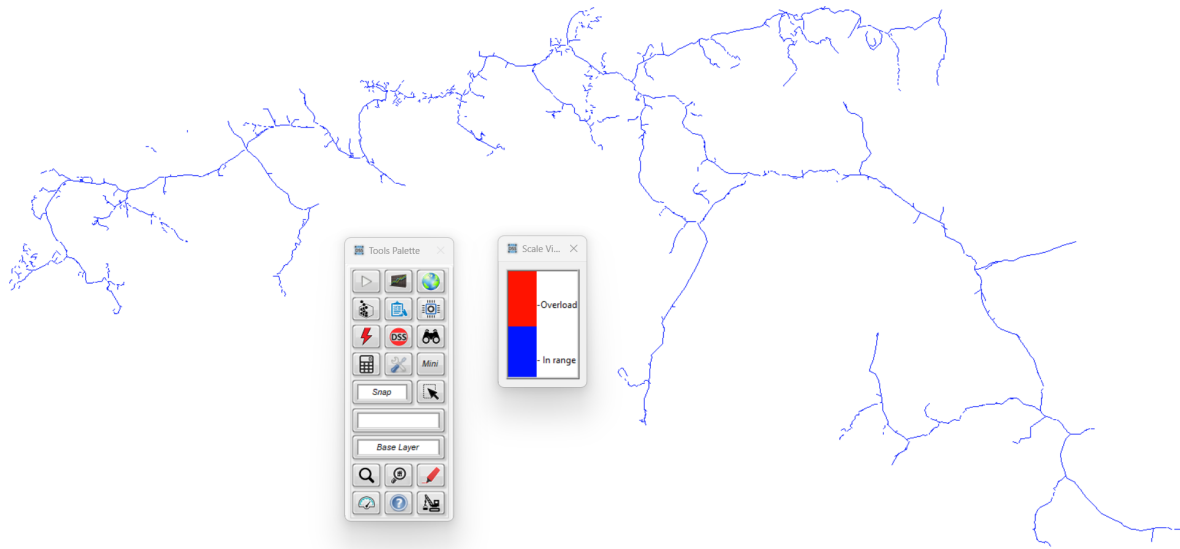


Figura 4.19

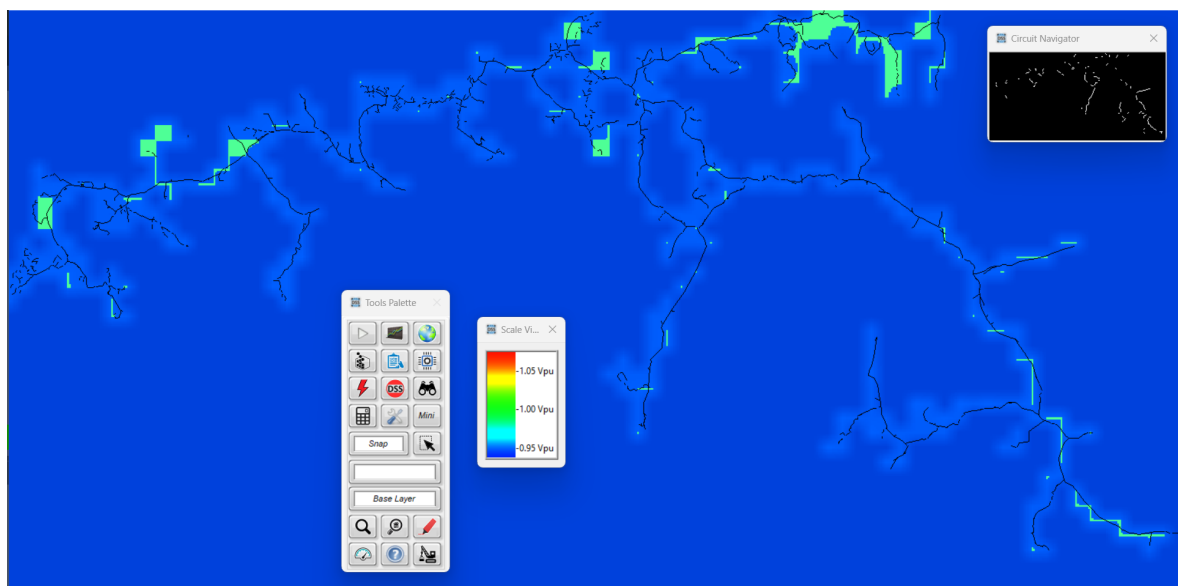
Mapa de sobrecargas de la red de distribución



Nota. Se puede observar que los elementos del circuito no se encuentran sobrecargados.

Figura 4.20

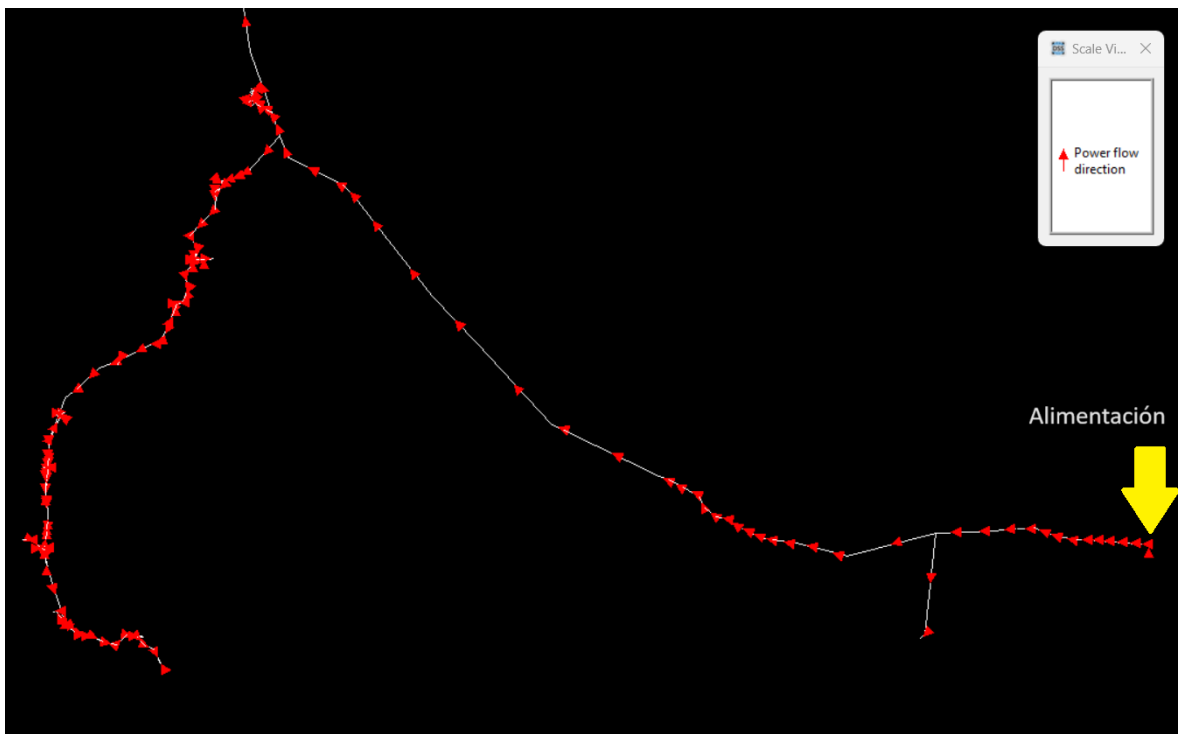
Mapa de nivel de tensión de la red de distribución



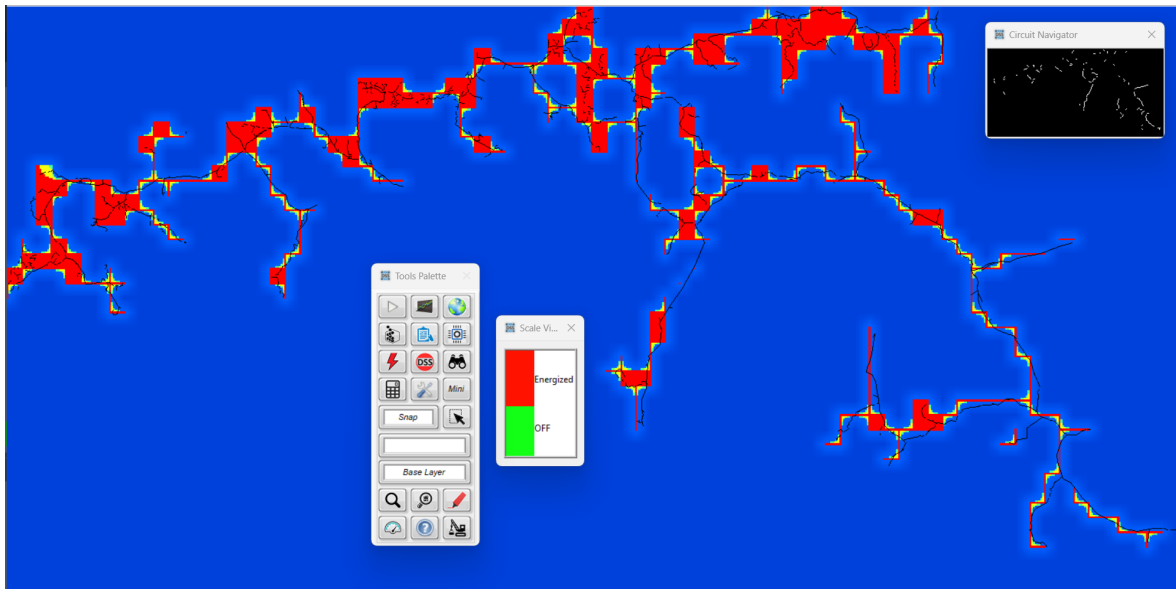
Nota. Se aprecia que la mayoría de elementos se encuentran con una tensión de entre 1 p.u. y 0,95 p.u.

Figura 4.21

Mapa de flujo de potencia de la red de distribución en un área cercana a la alimentación



Nota. Se observa la dirección radial del flujo de potencia desde la fuente de alimentación.

Figura 4.22*Mapa de elementos aislados*

Nota. Se pueden apreciar en rojo los elementos energizados de la red. Las pequeñas zonas en amarillo se deben a problemas de coloreado del programa.

Figura 4.23*Reporte de elementos aislados*

```

ISOLATED CIRCUIT ELEMENT REPORT

*** THE FOLLOWING BUSES HAVE NO CONNECTION TO THE SOURCE ***

***** THE FOLLOWING SUB NETWORKS ARE ISOLATED *****

***** THE FOLLOWING ENABLED ELEMENTS ARE ISOLATED *****

*** THE FOLLOWING BUSES ARE NOT CONNECTED TO ANY POWER DELIVERY ELEMENT ***

```

Nota. Se observa que no aparece ningún elemento aislado, por lo que todos los elementos de la red están conectados a la fuente de alimentación.

Capítulo 5

Conclusiones y recomendaciones

5.1. Conclusiones

Con respecto al primer objetivo específico:

- Se realiza una minería de datos mediante Python con la cual eliminar datos innecesarios para la implementación de la red en OpenDSS.
- Mediante una minería de datos, se eliminan y corrigen datos que generaban problemas con la importación del circuito a OpenDSS.

Acorde al segundo objetivo específico:

- Se logra crear un programa en Python con el cual dar formato de acuerdo con OpenDSS y con cada uno de los elementos que componen la red de distribución.
- Se generan dos archivos maestros para la implementación de la red en el *software* de simulación y con los cuales estudiar el comportamiento de la red de distribución con cargas fijas y con cargas variables.

Según el tercer objetivo específico:

- Se importan los datos de la red al *software* de OpenDSS para su posterior uso en simulaciones.

- Se importa el circuito a OpenDSS-G dentro de un sistema de información geográfico.

Referente al cuarto objetivo específico:

- Se realizan simulaciones con cargas fijas y con 6 perfiles de demanda que van desde demanda baja hasta demanda alta.
- Se generan reportes de potencia, corriente y tensión de la red, con las cuales visualizar información tanto específica de ciertos elementos como el comportamiento general de la red.
- Se generan gráficos de perfiles de tensión, así como del comportamiento de la tensión, corriente y potencia respecto a la demanda, de forma que se pueda extraer información de la red de distribución.
- Se generan mapas de calor con los cuales visualizar el comportamiento de la red de distribución dentro de un espacio geográfico.
- Durante la generación, implementación y análisis de los archivos del circuito, se presentaron dificultades debido al manejo tosco de OpenDSS y OpenDSS-G así como con sus características poco amigables con el usuario como es el caso de la herramienta de zoom con la cual no es sencilla navegar por el circuito e identificar elementos, por lo que el manejo de estos programas dificulta las labores de simulación y análisis a comparación de otro software como es ETAP y CYME los cuales brindan herramientas más sencillas y cómodas para estas labores.

5.2. Recomendaciones

Según el primer objetivo:

- Para la generación de los *CodeLine*, implementar un programa que genere de forma automática estos para cada tipo de línea, eliminando el trabajo manual y reduciendo el tiempo de procesado.

- Al generar o actualizar las bases de datos de los elementos de los circuitos, evitar incluir información no requerida por OpenDSS para disminuir la minería de datos requerida.

Con respecto al segundo objetivo:

- Respecto a la programación, optimizar el código con funciones y comandos más eficientes, lo que disminuiría el tiempo que tarda el proceso en dar formato a los datos.
- Respecto al uso de Python, existe una librería de OpenDSS llamada OpenDSS-Direct.py, por lo que un estudio de la librería podría revelar las capacidades de esta para facilitar las labores de simulación y generación de archivos.

Referente al tercer objetivo:

- En caso de agregar elementos a la simulación para pruebas, crear archivos nuevos para estos elementos y llamarlos desde el archivo maestro, con el fin de no modificar los archivos originales y facilitar las simulaciones.
- En caso de implementar un nuevo circuito, utilizar los reportes de elementos aislados y de anillos, de forma que se puedan identificar errores durante este proceso.
- Hacer uso de versiones en los archivos, ya que en caso de generar algún error con la actualización de los datos del circuito, se cuente con una versión anterior que sirva como copia de seguridad.
- Mantener actualizados los circuitos en OpenDSS, de forma que las simulaciones den resultados acordes a los sistemas físicos.

Acorde al cuarto objetivo:

- Implementar perfiles de demanda propios de cada carga, con el fin de realizar una simulación más cercana al comportamiento real de la red de distribución.

- Analizar y escoger cuál de los modelos de carga presentes en OpenDSS se ajustan mejor para cada carga de la red de distribución, de forma que los resultados de la simulación se acercarán más al sistema físico.
- Verificar y validar los resultados generados por los programas de simulación con mediciones de los equipos físicos o con otro programa de simulación.
- Para futuros trabajos relacionados, realizar la implementación, simulación y estudio de un circuito que incluya elementos de generación distribuida.

Bibliografía

- Arguello, A., Gómez, G., Quirós-Tortós, J., & Valverde, G. (2018). Distribution Network Element Model Parameters: Creation of Database. *2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII)*, 1-6. <https://doi.org/10.1109/CONCAPAN.2018.8596604>
- Asamblea Legislativa. (2021). *Ley N°10086. Promoción y regulación de recursos energéticos distribuidos partir de fuentes renovables*. Consultado el 27 de noviembre de 2023, desde http://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=96064
- Autoridad Reguladora de los Servicios Públicos. (2015). *Supervisión de la comercialización del suministro eléctrico en baja y media tensión (norma técnica regulatoria AR-NT-SUCOM)*. Consultado el 14 de abril de 2024, desde https://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?nValor1=1&nValor2=79376
- Autoridad Reguladora de los Servicios Públicos. (2019). *Subestaciones eléctricas*. Consultado el 15 de abril de 2024, desde <https://aresep.go.cr/datos-abiertos/subestaciones-electricas/>
- Autoridad Reguladora de los Servicios Públicos. (2023). *Estudios de Capacidad de Alojamiento de DER en Circuitos de Distribución, Modelado y Simulación de Reguladores en Línea y Capacitores*.
- Carta González, J. A., Calero Pérez, R., Colmenar Santos, A., & Castro Gil, M. A. (2009). *Centrales de energías renovables: Generación eléctrica con energías renovables* (2^a ed.). Pearson UNED Editorial.
- Chapman, S. J. (2012). *Máquinas eléctricas* (5^a ed.). McGraw-Hill Education.

- Dugan, R. C., & Montenegro, D. (2020a). *Program on Technology Innovation: OpenDSS-G, A Graphical View of OpenDSS*. Electric Power Research Institute.
- Dugan, R. C., & Montenegro, D. (2020b). *Reference Guide The Open Distribution System Simulator™ (OpenDSS)*. Electric Power Research Institute.
- European Petroleum Survey Group. (s.f.). *About the EPSG Dataset*. Consultado el 22 de abril de 2024, desde <https://epsg.org/home.html>
- Furones, A. M. (2011). *Sistema y marco de referencia terrestre, sistemas de coordenadas*. Universidad Politécnica de Valencia.
- Gallego, L. A., Padilha-Feltrin, A., & López Lezama, J. M. (2008). Flujo de potencia en sistemas de distribución considerando incertidumbre en la demanda y en los parámetros de los alimentadores. *Scientia Et Technica*, 2(39). <https://doi.org/https://doi.org/10.22517/23447214.3139>
- Garcés-Ruiz, A. (2022). Power Flow in Unbalanced Three-Phase Power Distribution Networks Using Matlab: Theory, analysis, and quasi-dynamic simulation. *Ingeniería*, 27(3), 1-25. <https://doi.org/https://doi-org.ezproxy.itcr.ac.cr/10.14483/23448393.19252>
- Giménez Gutiérrez, W. F. (2000). *Modelización de interruptores eléctricos de potencia* [Tesis doctoral, Universitat Politècnica de Catalunya].
- Gómez-Ramírez, G. A., Luévano-Reyes, I. A., Meza, C., & García-Santander, L. (2021). Demand Response Improvement using Storage Power Systems: Case study of Honduras. *2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 1-6. <https://doi.org/10.1109/CHILECON54041.2021.9703062>
- Gómez-Ramírez, G. A., Luévano-Reyes, I. A., Mora-Jiménez, G., García-Santander, L., Laskano, M. Z., & Meza, C. (2022). Increasing Distribution Network Capacity through Storage in Central American Countries: A Case Study. *2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, 1-6. <https://doi.org/10.1109/ICA-ACCA56767.2022.10006043>

- Gómez-Ramírez, G. A. (2016). Evolución y tendencias de índices de confiabilidad en sistemas eléctricos de potencia. *Revista Tecnología en Marcha*, 29(2), 3-13.
- Gómez-Ramírez, G. A., Meza, C., Mora-Jiménez, G., Morales, J. R. R., & García-Santander, L. (2023). The Central American Power System: Achievements, Challenges, and Opportunities for a Green Transition. *Energies*, 16(11). <https://doi.org/10.3390/en16114328>
- Gómez-Ramírez, G. A., Mora-Jiménez, G., & Meza, C. (2023). Simulación del sistema de interconexión eléctrica de los países de América Central usando ETAP. *Revista Tecnología en Marcha*, 36(2), Pág. 50-58. <https://doi.org/10.18845/tm.v36i2.6007>
- Grupo ICE. (2021). *Historia*. Consultado el 27 de noviembre de 2023, desde <https://www.grupoice.com/wps/portal/ICE/quienessomos/quienes-somos/historia>
- Grupo ICE. (2022). *Grupo ICE*. Consultado el 27 de noviembre de 2023, desde <https://www.grupoice.com/wps/portal/ICE/quienessomos/quienes-somos/grupoice-inf>
- Grupo ICE. (2023). *Principios corporativos*. Consultado el 27 de noviembre de 2023, desde <https://www.grupoice.com/wps/portal/ICE/quienessomos/quienes-somos/principios-corporativos>
- Hashemi-Dezaki, H., Askarian-Abyaneh, H., & Haeri-Khiavi, H. (2015). Reliability optimization of electrical distribution systems using internal loops to minimize energy not-supplied (ENS). *Journal of applied research and technology*, 13(3), 416-424.
- Imprenta Nacional. (2023). Procedimiento de capacidad de penetración de DER por circuito de distribución que se integran con las redes de distribución del SEN. *La Gaceta*.
- Instituto Costarricense de Electricidad. (2015). *Costa Rica: Matriz eléctrica. Un modelo sostenible, único en el mundo*. https://www.grupoice.com/wps/wcm/connect/8823524c-7cc7-4cef-abde-a1f06e14da0e/matriz_folleto_web2.pdf?MOD=AJPERES&CVID=18SK4gG

- Instituto Costarricense de Electricidad. (2020). *Costa Rica: Matriz eléctrica. Un modelo sostenible, único en el mundo*. <https://www.grupoice.com/wps/wcm/connect/19b209b1-049b-4cb4-bf4f-ca7170ce2749/Matriz+el%C3%A9ctrica+2020.pdf?MOD=AJPERES&CACHEID=ROOTWORKSPACE-19b209b1-049b-4cb4-bf4f-ca7170ce2749-o6SDoG0>
- Instituto Costarricense de Electricidad. (2021). *Plan de Expansión de la Transmisión 2021-2031*. https://www.grupoice.com/wps/wcm/connect/ed6734d2-f0d8-4a5b-9c11-6a96a5d263af/PLAN+DE+EXPANSI%C3%93N+DE+TRANSMISI%C3%93N_2021-2031_compressed.pdf?MOD=AJPERES&CVID=nJBsHZ.
- Instituto Costarricense de Electricidad. (2023). *Proyecto: Control y Aseguramiento de la calidad de los datos del SIRDE, Configuraciones MT Aérea y Subterránea*.
- Instituto Geográfico Nacional. (2016). *Sistema de Referencia Geodésico de Costa Rica (NTIG_CR01₀1,2016)*. Consultado el 22 de abril de 2024, desde https://www.snitcr.go.cr/ico_normativa?id=NTIG_CR01_01.2016
- Issicaba, D., & Coelho, J. (2016). Evaluation of the forward-backward sweep load flow method using the contraction mapping principle. *International Journal of Electrical and Computer Engineering*, 6(6). <https://doi.org/http://doi.org/10.11591/ijece.v6i6.pp3229-3237>
- Kersting, W. H., & Kerestes, R. J. (2023). *Distribution System Modeling and Analysis with MATLAB® and WindMil®* (5^a ed.). CRC Press.
- Laboratorio de Investigación en Potencia y Energía. (s.f.-a). *Herramientas de Simulación de Redes Eléctricas de Distribución. Modelado y Simulación de la Demanda*. Universidad de Costa Rica.
- Laboratorio de Investigación en Potencia y Energía. (s.f.-b). *Herramientas de Simulación de Redes Eléctricas de Distribución. Modelado y Simulación de Transformadores*. Universidad de Costa Rica.
- Leal Parra, J. (2016). *Manual de uso del programa OpenDSS para el curso de distribución y transmisión de potencia* [Tesis]. Universidad de Costa Rica.

- Llano, L. E., & Moreno, J. (2014). On-line analysis of fault events in power transmission systems using SOE, fuzzy logic and expert systems. *Revista técnica de la Facultad de Ingeniería*, 36(2).
- Madrigal, L. M. (2024). *Generación eléctrica renovable en Costa Rica llegó a 94,91 % durante 2023 pese a fenómeno de El Niño*. <https://delfino.cr/2024/02/generacion-electrica-renovable-en-costa-rica-llego-a-9491-durante-2023-pese-a-fenomeno-de-el-nino>
- Morales de la Cruz, V. M. (2018). *Análisis de sobretensiones transitorias en líneas de transmisión a través del método J Martí* [Tesis de maestría]. Benemérita Universidad Autónoma de Puebla.
- Nolasco Valenzuela, J. S. (2018). *Python Aplicaciones prácticas*. Ra-Ma.
- Pandas. (s.f.). *Pandas*. Consultado el 22 de abril de 2024, desde <https://pandas.pydata.org/>
- Presidencia de la Republica de Costa Rica. (2021). *Firmada ley que regula la generación de energía distribuida en el país*. <https://www.presidencia.go.cr/comunicados/2021/12/firmada-ley-que-regula-la-generacion-de-energia-distribuida-en-el-pais/#:~:text=La%20ley%2010086%20establece%20las,de%20fuentes%20de%20energ%C3%ADa%20renovables>
- PROJ Contributors. (2024). *PROJ*. Consultado el 22 de abril de 2024, desde <https://proj.org/en/9.4/>
- Python Software Foundation. (2024). *math — Mathematical functions*. Consultado el 22 de abril de 2024, desde <https://docs.python.org/3/library/math.html>
- Quintero, C. M., López-Lezama, J. M., & Muñoz-Galeano, N. (2019). Técnicas para aumentar la Seguridad de la Operación en Condiciones de Red Degradada a través de Modificaciones Topológicas en Sistemas de Potencia. *Información Tecnológica*, 30(6), 363-376. <https://doi.org/http://dx.doi.org/10.4067/S0718-07642019000600363>
- Spatial Reference. (s.f.-a). *EPSG:4326*. Consultado el 22 de abril de 2024, desde <https://spatialreference.org/ref/epsg/4326/>

- Spatial Reference. (s.f.-b). *EPSG:5367*. Consultado el 22 de abril de 2024, desde <https://spatialreference.org/ref/epsg/5367/>
- Tabitas-Aguilar, M. I., Chacón-Reyna, J. M., Orta-Pesina, H., Mercado-Hernández, R., Flores-Suarez, A. E., Ponce-García, G., Rebollar-Telles, E. A., Fernández-Salas, I., Rodríguez-Castro, V. A., Quiroz-Martínez, H., Saucedo-Garza, J. S., & Varela-Echavarría, A. (2018). Análisis de la Distribución Espacial y Temporal de los Casos de Dengue (2009–2012) Mediante un Sistema de Información Geográfica en el Área Metropolitana de Monterrey, NL, México. *Southwestern Entomologist*, *43*(3), 761-771. <https://doi.org/https://doi-org.ezproxy.itcr.ac.cr/10.3958/059.043.0321>
- Tuta, Ó. F., & Hincapié, R. A. (2011). Coordinación óptima de dispositivos de protección en sistemas de distribución con presencia de generación distribuida. *Scientia Et Technica*, *16*(49), 14-19.
- Zhaoyu, W. (s.f.). *Distribution System Protection*. Consultado el 20 de abril de 2024, desde <https://wzy.ece.iastate.edu/Courses/EE555/08%20Chapter%2011%20Distribution%20System%20Protection.pdf>

Apéndice A

Código empleado

A.1. LimpiadorExcel.py

```
1  import pandas as pd
2
3  Busbar=pd.read_excel('CircuitoTFG.xlsx', index_col=0, decimal='.')
4
5  Busbar=Busbar.drop(columns=['AliasName', 'CoordX2', 'CoordY2'])
6  headers=Busbar.columns
7  for k in range(2, (len(headers)-2)):
8      Busbar=Busbar.drop(columns=[headers[k]])
9  Busbarcor=Busbar.copy()
10 for k in range (len(Busbarcor)):
11     Busbarcor.iat[k,2]= float(str(Busbarcor.iloc[k,2])[:11])
12     Busbarcor.iat[k,3]= float(str(Busbarcor.iloc[k,3])[:11])
13
14 Filtro=Busbarcor[Busbarcor.duplicated(
15     subset=['Un', 'CoordX1', 'CoordY1'], keep=False) == True]
16 BusRep=pd.DataFrame(columns=['Bus1', 'Bus2'], index=range(len(Filtro)))
17 for k in range(len(Filtro)):
18     for c in range(k+1, len(Filtro)):
19         if Filtro.iloc[k, 3] == Filtro.iloc[c, 3] and
20             Filtro.iloc[k, 2] == Filtro.iloc[c, 2]:
21             BusRep.iat[k,0]=Filtro.iat[k,0]
22             BusRep.iat[k,1]=Filtro.iat[c,0]
23
24 BusRep=BusRep.dropna()
25 for k in range(len(BusRep)):
26     Bus = Busbar[Busbar['Name'] == BusRep.iloc[k,1]]
27     BusInd=Bus.index
28     BusInd=BusInd.to_list()
29     Busbar=Busbar.drop(BusInd[0], axis=0)
30
31 LineAsym=pd.read_excel('CircuitoTFG.xlsx', sheet_name='LineAsym',
32     decimal='.')
33 LineAsym=LineAsym.drop(columns=['Index', 'Switch1', 'Switch2',
34     'AliasName1', 'LineType', 'Phase', 'Un'])
35
36 LimpBus1=LineAsym.merge(BusRep, left_on='Node1', right_on='Bus2')
37 for k in range(len(LimpBus1)):
38     Bus = LineAsym[LineAsym['Node1'] == LimpBus1.iloc[k,6]]
39     Bus = Bus[Bus['Node2'] == LimpBus1.iloc[k,1]]
```

```

36     BusInd=Bus.index
37     BusInd=BusInd.to_list()
38     LineAsym.iat[BusInd[0],0]=LimpBus1.iat[k,5]
39
40 LimpBus2=LineAsym.merge(BusRep, left_on='Node2', right_on='Bus2')
41 for k in range(len(LimpBus2)):
42     Bus = LineAsym[LineAsym['Node2'] == LimpBus2.iloc[k,6]]
43     Bus = Bus[Bus['Node1'] == LimpBus2.iloc[k,0]]
44     BusInd=Bus.index
45     BusInd=BusInd.to_list()
46     LineAsym.iat[BusInd[0],1]=LimpBus2.iat[k,5]
47 LineAsym=LineAsym[['Name', 'Node1', 'Node2', 'LibraryType', 'Length']]
48
49 TrafoAsym=pd.read_excel('CircuitoTFG.xlsx',
sheet_name='Trafo2WindingAsym', index_col=0, decimal='.')
50 TrafoAsym=TrafoAsym.drop(columns=['Switch1', 'Switch2',
'IsRegulated', 'CoordX1', 'CoordY1', 'SymbolSize', 'SymbolAngle'])
51
52 Trafo=pd.read_excel('CircuitoTFG.xlsx', sheet_name='Trafo2Winding',
index_col=0, decimal='.')
53 Trafo=Trafo.drop(columns=['Switch1', 'Switch2', 'IsRegulated',
'CoordX1', 'CoordY1', 'SymbolSize', 'SymbolAngle'])
54
55 Load=pd.read_excel('CircuitoTFG.xlsx', sheet_name='Load',
index_col=0, decimal='.')
56 Load=Load.drop(columns=['VelanderK1', 'LfType',
'CoordX1', 'CoordY1'])
57 Load=Load[['Name', 'Node1', 'Un', 'Tipo', 'Unit', 'E',
'CosPhi', 'Switch1', 'Phase']]
58
59 with pd.ExcelWriter('CircuitoLimpio.xlsx') as writer:
60     Busbar.to_excel(writer, sheet_name='Busbar', index=False)
61     LineAsym.to_excel(writer, sheet_name='LineAsym',
index=False)
62     TrafoAsym.to_excel(writer, sheet_name='Trafo2WindingAsym',
index=False)
63     Trafo.to_excel(writer, sheet_name='Trafo2Winding',
index=False)
64     Load.to_excel(writer, sheet_name='Load', index=False)

```

A.2. CreadorInputsOpenDSS.py

```

1  import pandas as pd
2  import csv
3  import math
4  from pyproj import Transformer
5
6  BusOrig=pd.read_excel('CircuitoLimpio.xlsx', decimal='.')
7  BusCoords=BusOrig.drop(columns=['Un'])
8
9  transformer = Transformer.from_crs("EPSG:5367", "EPSG:4326")
10 for k in range(0,len(BusOrig)):
11     Conv=transformer.transform(BusCoords.iloc[k, 2],
BusCoords.iloc[k, 1])

```

```

12     BusCoords.iat[k,0]=(BusCoords.iat[k,0]+",")
13     BusCoords.iat[k,1]=(str(Conv[1])+",")
14     BusCoords.iat[k,2]=Conv[0]
15 BusCoords.to_csv('BusCoords.dss', sep=' ', index=False,
16 header=False)
17
17 LineAsym=pd.read_excel('CircuitoLimpio.xlsx',
18 sheet_name='LineAsym', decimal='.')
18 Lista_Lineas_MT_3F=pd.DataFrame()
19 Lista_Lineas_MT_2F=pd.DataFrame()
20 Lista_Lineas_MT_1F=pd.DataFrame()
21 Lista_Lineas_BT=pd.DataFrame()
22
23 comp1='MT_ABC'
24 comp2='MT_SUB_ABC'
25 comp3='MT_A:'
26 comp4='MT_B:'
27 comp5='MT_C:'
28 comp6='MT_SUB_A:'
29 comp7='MT_SUB_B:'
30 comp8='MT_SUB_C:'
31 comp9='MT'
32 comp10='MT_AB:'
33 comp11='MT_SUB_AB:'
34 comp12='MT_AC:'
35 comp13='MT_SUB_AC:'
36 comp14='MT_BC:'
37 comp15='MT_SUB_BC:'
38
39 for k in range(len(LineAsym)):
40     if LineAsym.iloc[k,3][0:6] == comp1 or
41        LineAsym.iloc[k,3][0:10] == comp2:
42         if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
43             Lista_Lineas_MT_3F =
44                 Lista_Lineas_MT_3F.append(LineAsym.iloc[k])
45         elif LineAsym.iloc[k,3][0:5] == comp3
46            or LineAsym.iloc[k,3][0:5] == comp4
47            or LineAsym.iloc[k,3][0:5] == comp5
48            or LineAsym.iloc[k,3][0:9] == comp6
49            or LineAsym.iloc[k,3][0:9] == comp7
50            or LineAsym.iloc[k,3][0:9] == comp8:
51             if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
52                 Lista_Lineas_MT_1F =
53                     Lista_Lineas_MT_1F.append(LineAsym.iloc[k])
54             elif LineAsym.iloc[k,3][0:2] == comp9:
55                 if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
56                     Lista_Lineas_MT_2F =
57                         Lista_Lineas_MT_2F.append(LineAsym.iloc[k])
58             else:
59                 if LineAsym.iloc[k,1] != LineAsym.iloc[k,2]:
60                     Lista_Lineas_BT =
61                         Lista_Lineas_BT.append(LineAsym.iloc[k])
62
63 for k in range(len(Lista_Lineas_MT_3F)):
64     Lista_Lineas_MT_3F.iat[k,0]=
65     ('New line.'+Lista_Lineas_MT_3F.iat[k,0])

```

```

55     Lista_Lineas_MT_3F.iat[k,1]=
        ('bus1='+Lista_Lineas_MT_3F.iat[k,1])
56     Lista_Lineas_MT_3F.iat[k,2]=
        ('bus2='+Lista_Lineas_MT_3F.iat[k,2])
57     Lista_Lineas_MT_3F.iat[k,3]=
        ('LineCode='+Lista_Lineas_MT_3F.iat[k,3].replace(' ','_'))
58     Lista_Lineas_MT_3F.iat[k,4]=
        ('length='+str(Lista_Lineas_MT_3F.iat[k,4]))
59     Lista_Lineas_MT_3F.to_csv('Lista_Lineas_MT_3F.dss', sep=' ',
        index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
60
61     for k in range(len(Lista_Lineas_MT_2F)):
62         if Lista_Lineas_MT_2F.iloc[k,3][0:6] == comp10 or
            Lista_Lineas_MT_2F.iloc[k,3][0:10] == comp11:
63             Lista_Lineas_MT_2F.iat[k,1]=
                ('bus1='+Lista_Lineas_MT_2F.iat[k,1]+' .1.2')
64             Lista_Lineas_MT_2F.iat[k,2]=
                ('bus2='+Lista_Lineas_MT_2F.iat[k,2]+' .1.2')
65         elif Lista_Lineas_MT_2F.iloc[k,3][0:6] == comp12 or
            Lista_Lineas_MT_2F.iloc[k,3][0:10] == comp12:
66             Lista_Lineas_MT_2F.iat[k,1]=
                ('bus1='+Lista_Lineas_MT_2F.iat[k,1]+' .1.3')
67             Lista_Lineas_MT_2F.iat[k,2]=
                ('bus2='+Lista_Lineas_MT_2F.iat[k,2]+' .1.3')
68         elif Lista_Lineas_MT_2F.iloc[k,3][0:6] == comp14 or
            Lista_Lineas_MT_2F.iloc[k,3][0:10] == comp15:
69             Lista_Lineas_MT_2F.iat[k,1]=
                ('bus1='+Lista_Lineas_MT_2F.iat[k,1]+' .2.3')
70             Lista_Lineas_MT_2F.iat[k,2]=
                ('bus2='+Lista_Lineas_MT_2F.iat[k,2]+' .2.3')
71         Lista_Lineas_MT_2F.iat[k,0]=
            ('New line.'+Lista_Lineas_MT_2F.iat[k,0])
72         Lista_Lineas_MT_2F.iat[k,3]=
            ('LineCode='+Lista_Lineas_MT_2F.iat[k,3].replace(' ','_'))
73         Lista_Lineas_MT_2F.iat[k,4]=
            ('length='+str(Lista_Lineas_MT_2F.iat[k,4]))
74     Lista_Lineas_MT_2F.to_csv('Lista_Lineas_MT_2F.dss', sep=' ',
        index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
75
76     for k in range(len(Lista_Lineas_MT_1F)):
77         if Lista_Lineas_MT_1F.iloc[k,3][0:5] == comp3 or
            Lista_Lineas_MT_1F.iloc[k,3][0:9] == comp6:
78             Lista_Lineas_MT_1F.iat[k,1]=
                ('bus1='+Lista_Lineas_MT_1F.iat[k,1]+' .1')
79             Lista_Lineas_MT_1F.iat[k,2]=
                ('bus2='+Lista_Lineas_MT_1F.iat[k,2]+' .1')
80         elif Lista_Lineas_MT_1F.iloc[k,3][0:5] == comp4 or
            Lista_Lineas_MT_1F.iloc[k,3][0:9] == comp7:
81             Lista_Lineas_MT_1F.iat[k,1]=
                ('bus1='+Lista_Lineas_MT_1F.iat[k,1]+' .2')
82             Lista_Lineas_MT_1F.iat[k,2]=
                ('bus2='+Lista_Lineas_MT_1F.iat[k,2]+' .2')
83         elif Lista_Lineas_MT_1F.iloc[k,3][0:5] == comp5 or
            Lista_Lineas_MT_1F.iloc[k,3][0:9] == comp8:
84             Lista_Lineas_MT_1F.iat[k,1]=
                ('bus1='+Lista_Lineas_MT_1F.iat[k,1]+' .3')

```

```

85     Lista_Lineas_MT_1F.iat[k,2]=
      ('bus2='+Lista_Lineas_MT_1F.iat[k,2]+' .3')
86     Lista_Lineas_MT_1F.iat[k,0]=
      ('New line.'+Lista_Lineas_MT_1F.iat[k,0])
87     Lista_Lineas_MT_1F.iat[k,3]=
      ('LineCode='+Lista_Lineas_MT_1F.iat[k,3].replace(' ','_'))
88     Lista_Lineas_MT_1F.iat[k,4]=
      ('length='+str(Lista_Lineas_MT_1F.iat[k,4]))
89     Lista_Lineas_MT_1F.to_csv('Lista_Lineas_MT_1F.dss', sep=' ',
      index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
90
91 for k in range(len(Lista_Lineas_BT)):
92     Lista_Lineas_BT.iat[k,0]=
      ('New line.'+Lista_Lineas_BT.iat[k,0])
93     Lista_Lineas_BT_Servicio.iat[k,1]=
      ('bus1='+Lista_Lineas_BT.iat[k,1])
94     Lista_Lineas_BT_Servicio.iat[k,2]=
      ('bus2='+Lista_Lineas_BT.iat[k,2])
95     Lista_Lineas_BT.iat[k,3]=
      ('LineCode='+Lista_Lineas_BT.iat[k,3].replace(' ','_'))
96     Lista_Lineas_BT.iat[k,4]=
      ('length='+str(Lista_Lineas_BT.iat[k,4]))
97     Lista_Lineas_BT.to_csv('Lista_Lineas_BT.dss', sep=' ',
      index=False, header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
98
99 react_list1F3W={ # List of reactances for single
      phase three winding transformers
100     '2.4':{
101         '0.12': {
102             '3': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
103             '5': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
104             '10': 'Xhl=1.54 Xht=1.54 Xlt=1.03',
105             '15': 'Xhl=1.19 Xht=1.19 Xlt=0.80',
106             '25': 'Xhl=1.49 Xht=1.49 Xlt=1.00',
107             '37': 'Xhl=1.71 Xht=1.71 Xlt=1.14',
108             '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
109             '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
110             '100': 'Xhl=2.14 Xht=2.14 Xlt=1.43',
111             '167': 'Xhl=2.35 Xht=2.35 Xlt=1.57',
112             '250': 'Xhl=3.17 Xht=3.17 Xlt=2.11',
113             '333': 'Xhl=3.52 Xht=3.52 Xlt=2.35',
114             '500': 'Xhl=3.61 Xht=3.61 Xlt=2.40'},
115         '0.24':{
116             '3': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
117             '5': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
118             '10': 'Xhl=1.40 Xht=1.40 Xlt=0.94',
119             '15': 'Xhl=1.19 Xht=1.19 Xlt=0.79',
120             '25': 'Xhl=1.49 Xht=1.49 Xlt=0.99',
121             '37': 'Xhl=1.70 Xht=1.70 Xlt=1.14',
122             '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
123             '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
124             '100': 'Xhl=1.94 Xht=1.94 Xlt=1.30',
125             '167': 'Xhl=2.15 Xht=2.15 Xlt=1.43',
126             '250': 'Xhl=2.69 Xht=2.69 Xlt=1.79',
127             '333': 'Xhl=2.66 Xht=2.66 Xlt=1.78',
128             '500': 'Xhl=2.76 Xht=2.76 Xlt=1.84'}}}

```

```

129 '4.16':{
130     '0.12': {
131         '3': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
132         '5': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
133         '10': 'Xhl=1.54 Xht=1.54 Xlt=1.03',
134         '15': 'Xhl=1.19 Xht=1.19 Xlt=0.80',
135         '25': 'Xhl=1.49 Xht=1.49 Xlt=1.00',
136         '37': 'Xhl=1.71 Xht=1.71 Xlt=1.14',
137         '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
138         '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
139         '100': 'Xhl=2.14 Xht=2.14 Xlt=1.43',
140         '167': 'Xhl=2.35 Xht=2.35 Xlt=1.57',
141         '250': 'Xhl=3.17 Xht=3.17 Xlt=2.11',
142         '333': 'Xhl=3.52 Xht=3.52 Xlt=2.35',
143         '500': 'Xhl=3.61 Xht=3.61 Xlt=2.40'}},
144     '0.24':{
145         '3': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
146         '5': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
147         '10': 'Xhl=1.40 Xht=1.40 Xlt=0.94',
148         '15': 'Xhl=1.19 Xht=1.19 Xlt=0.79',
149         '25': 'Xhl=1.49 Xht=1.49 Xlt=0.99',
150         '37': 'Xhl=1.70 Xht=1.70 Xlt=1.14',
151         '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
152         '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
153         '100': 'Xhl=1.94 Xht=1.94 Xlt=1.30',
154         '167': 'Xhl=2.15 Xht=2.15 Xlt=1.43',
155         '250': 'Xhl=2.69 Xht=2.69 Xlt=1.79',
156         '333': 'Xhl=2.66 Xht=2.66 Xlt=1.78',
157         '500': 'Xhl=2.76 Xht=2.76 Xlt=1.84'}},
158 '7.2':{
159     '0.12': {
160         '3': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
161         '5': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
162         '10': 'Xhl=1.54 Xht=1.54 Xlt=1.03',
163         '15': 'Xhl=1.19 Xht=1.19 Xlt=0.80',
164         '25': 'Xhl=1.49 Xht=1.49 Xlt=1.00',
165         '37': 'Xhl=1.71 Xht=1.71 Xlt=1.14',
166         '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
167         '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
168         '100': 'Xhl=2.14 Xht=2.14 Xlt=1.43',
169         '167': 'Xhl=2.35 Xht=2.35 Xlt=1.57',
170         '250': 'Xhl=3.17 Xht=3.17 Xlt=2.11',
171         '333': 'Xhl=3.52 Xht=3.52 Xlt=2.35',
172         '500': 'Xhl=3.61 Xht=3.61 Xlt=2.40'}},
173     '0.24':{
174         '3': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
175         '5': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
176         '10': 'Xhl=1.40 Xht=1.40 Xlt=0.94',
177         '15': 'Xhl=1.19 Xht=1.19 Xlt=0.79',
178         '25': 'Xhl=1.49 Xht=1.49 Xlt=0.99',
179         '37': 'Xhl=1.70 Xht=1.70 Xlt=1.14',
180         '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
181         '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
182         '100': 'Xhl=1.94 Xht=1.94 Xlt=1.30',
183         '167': 'Xhl=2.15 Xht=2.15 Xlt=1.43',
184         '250': 'Xhl=2.69 Xht=2.69 Xlt=1.79',

```

```

185          '333': 'Xhl=2.66 Xht=2.66 Xlt=1.78',
186          '500': 'Xhl=2.76 Xht=2.76 Xlt=1.84'}},
187
188    '13.8':{
189      '0.12': {
190        '3': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
191        '5': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
192        '10': 'Xhl=1.54 Xht=1.54 Xlt=1.03',
193        '15': 'Xhl=1.19 Xht=1.19 Xlt=0.80',
194        '25': 'Xhl=1.49 Xht=1.49 Xlt=1.00',
195        '37': 'Xhl=1.71 Xht=1.71 Xlt=1.14',
196        '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
197        '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
198        '100': 'Xhl=2.14 Xht=2.14 Xlt=1.43',
199        '167': 'Xhl=2.35 Xht=2.35 Xlt=1.57',
200        '250': 'Xhl=3.17 Xht=3.17 Xlt=2.11',
201        '333': 'Xhl=3.52 Xht=3.52 Xlt=2.35',
202        '500': 'Xhl=3.61 Xht=3.61 Xlt=2.40'}},
203      '0.24':{
204        '3': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
205        '5': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
206        '10': 'Xhl=1.40 Xht=1.40 Xlt=0.94',
207        '15': 'Xhl=1.19 Xht=1.19 Xlt=0.79',
208        '25': 'Xhl=1.49 Xht=1.49 Xlt=0.99',
209        '37': 'Xhl=1.70 Xht=1.70 Xlt=1.14',
210        '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
211        '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
212        '100': 'Xhl=1.94 Xht=1.94 Xlt=1.30',
213        '167': 'Xhl=2.15 Xht=2.15 Xlt=1.43',
214        '250': 'Xhl=2.69 Xht=2.69 Xlt=1.79',
215        '333': 'Xhl=2.66 Xht=2.66 Xlt=1.78',
216        '500': 'Xhl=2.76 Xht=2.76 Xlt=1.84'}},
217
218    '24.9':{
219      '0.12': {
220        '3': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
221        '5': 'Xhl=0.82 Xht=0.82 Xlt=0.55',
222        '10': 'Xhl=1.54 Xht=1.54 Xlt=1.03',
223        '15': 'Xhl=1.19 Xht=1.19 Xlt=0.80',
224        '25': 'Xhl=1.49 Xht=1.49 Xlt=1.00',
225        '37': 'Xhl=1.71 Xht=1.71 Xlt=1.14',
226        '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',
227        '75': 'Xhl=1.80 Xht=1.80 Xlt=1.20',
228        '100': 'Xhl=2.14 Xht=2.14 Xlt=1.43',
229        '167': 'Xhl=2.35 Xht=2.35 Xlt=1.57',
230        '250': 'Xhl=3.17 Xht=3.17 Xlt=2.11',
231        '333': 'Xhl=3.52 Xht=3.52 Xlt=2.35',
232        '500': 'Xhl=3.61 Xht=3.61 Xlt=2.40'}},
233      '0.24':{
234        '3': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
235        '5': 'Xhl=1.31 Xht=1.31 Xlt=0.87',
236        '10': 'Xhl=1.4 Xht=1.4 Xlt=0.94',
237        '15': 'Xhl=1.19 Xht=1.19 Xlt=0.79',
238        '25': 'Xhl=1.49 Xht=1.49 Xlt=0.99',
239        '37': 'Xhl=1.7 Xht=1.7 Xlt=1.14',
240        '50': 'Xhl=1.86 Xht=1.86 Xlt=1.24',

```

```

241          '75': 'Xhl=1.8 Xht=1.8 Xlt=1.2',
242          '100': 'Xhl=1.94 Xht=1.94 Xlt=1.3',
243          '167': 'Xhl=2.15 Xht=2.15 Xlt=1.43',
244          '250': 'Xhl=2.69 Xht=2.69 Xlt=1.79',
245          '333': 'Xhl=2.66 Xht=2.66 Xlt=1.78',
246          '500': 'Xhl=2.76 Xht=2.76 Xlt=1.84'}},
247  '34.5':{
248      '0.12': {
249          '3': 'Xhl=4.92 Xht=4.92 Xlt=3.28',
250          '5': 'Xhl=4.92 Xht=4.92 Xlt=3.28',
251          '10': 'Xhl=5.34 Xht=5.34 Xlt=3.56',
252          '15': 'Xhl=5.52 Xht=5.52 Xlt=3.68',
253          '25': 'Xhl=5.65 Xht=5.65 Xlt=3.77',
254          '37': 'Xhl=5.80 Xht=5.80 Xlt=3.87',
255          '50': 'Xhl=5.90 Xht=5.90 Xlt=3.93',
256          '75': 'Xhl=5.94 Xht=5.94 Xlt=3.96',
257          '100': 'Xhl=5.97 Xht=5.97 Xlt=3.98',
258          '167': 'Xhl=6.04 Xht=6.04 Xlt=4.03',
259          '250': 'Xhl=6.08 Xht=6.08 Xlt=4.05',
260          '333': 'Xhl=6.10 Xht=6.10 Xlt=4.07',
261          '500': 'Xhl=6.12 Xht=6.12 Xlt=4.08',
262          '667': 'Xhl=6.12 Xht=6.12 Xlt=4.08',
263          '833': 'Xhl=6.12 Xht=6.12 Xlt=4.08'}},
264      '0.24':{
265          '3': 'Xhl=4.92 Xht=4.92 Xlt=3.28',
266          '5': 'Xhl=4.92 Xht=4.92 Xlt=3.28',
267          '10': 'Xhl=5.34 Xht=5.34 Xlt=3.56',
268          '15': 'Xhl=5.52 Xht=5.52 Xlt=3.68',
269          '25': 'Xhl=5.65 Xht=5.65 Xlt=3.77',
270          '37': 'Xhl=5.8 Xht=5.8 Xlt=3.86',
271          '50': 'Xhl=5.89 Xht=5.89 Xlt=3.93',
272          '75': 'Xhl=5.94 Xht=5.94 Xlt=3.96',
273          '100': 'Xhl=5.98 Xht=5.98 Xlt=3.98',
274          '167': 'Xhl=6.04 Xht=6.04 Xlt=4.02',
275          '250': 'Xhl=6.07 Xht=6.07 Xlt=4.05',
276          '333': 'Xhl=6.1 Xht=6.1 Xlt=4.06',
277          '500': 'Xhl=6.12 Xht=6.12 Xlt=4.08',
278          '667': 'Xhl=6.12 Xht=6.12 Xlt=4.08',
279          '833': 'Xhl=6.12 Xht=6.12 Xlt=4.08'}}}
280
281 res_list1F3W={ # List of resistances for single
282 phase three winding transformers
283
284     '2.4':{
285         '0.12': {
286             '3': '%Rs=[0.95 1.90 1.90]',
287             '5': '%Rs=[0.95 1.90 1.90]',
288             '10': '%Rs=[0.90 1.80 1.80]',
289             '15': '%Rs=[0.80 1.60 1.60]',
290             '25': '%Rs=[0.71 1.42 1.42]',
291             '37': '%Rs=[0.65 1.29 1.29]',
292             '50': '%Rs=[0.56 1.12 1.12]',
293             '75': '%Rs=[0.55 1.10 1.10]',
294             '100': '%Rs=[0.54 1.09 1.09]',
295             '167': '%Rs=[0.54 1.07 1.07]',
296             '250': '%Rs=[0.52 1.04 1.04]',

```

```

296         '333': '%Rs=[0.50 1.00 1.00] ',
297         '500': '%Rs=[0.50 1.00 1.00] ',
298         '667': '%Rs=[0.50 1.00 1.00] }',
299     '0.24':{
300         '3': '%Rs=[0.95 1.90 1.90] ',
301         '5': '%Rs=[0.95 1.90 1.90] ',
302         '10': '%Rs=[0.90 1.80 1.80] ',
303         '15': '%Rs=[0.80 1.60 1.60] ',
304         '25': '%Rs=[0.71 1.42 1.42] ',
305         '37': '%Rs=[0.65 1.29 1.29] ',
306         '50': '%Rs=[0.56 1.12 1.12] ',
307         '75': '%Rs=[0.55 1.10 1.10] ',
308         '100': '%Rs=[0.54 1.09 1.09] ',
309         '167': '%Rs=[0.54 1.07 1.07] ',
310         '250': '%Rs=[0.52 1.04 1.04] ',
311         '333': '%Rs=[0.50 1.00 1.00] ',
312         '500': '%Rs=[0.50 1.00 1.00] ',
313         '667': '%Rs=[0.50 1.00 1.00] }},
314
315     '4.16':{
316         '0.12': {
317             '3': '%Rs=[0.95 1.90 1.90] ',
318             '5': '%Rs=[0.95 1.90 1.90] ',
319             '10': '%Rs=[0.90 1.80 1.80] ',
320             '15': '%Rs=[0.80 1.60 1.60] ',
321             '25': '%Rs=[0.71 1.42 1.42] ',
322             '37': '%Rs=[0.65 1.29 1.29] ',
323             '50': '%Rs=[0.56 1.12 1.12] ',
324             '75': '%Rs=[0.55 1.10 1.10] ',
325             '100': '%Rs=[0.54 1.09 1.09] ',
326             '167': '%Rs=[0.54 1.07 1.07] ',
327             '250': '%Rs=[0.52 1.04 1.04] ',
328             '333': '%Rs=[0.50 1.00 1.00] ',
329             '500': '%Rs=[0.50 1.00 1.00] ',
330             '667': '%Rs=[0.50 1.00 1.00] }',
331         '0.24':{
332             '3': '%Rs=[0.95 1.90 1.90] ',
333             '5': '%Rs=[0.95 1.90 1.90] ',
334             '10': '%Rs=[0.90 1.80 1.80] ',
335             '15': '%Rs=[0.80 1.60 1.60] ',
336             '25': '%Rs=[0.71 1.42 1.42] ',
337             '37': '%Rs=[0.65 1.29 1.29] ',
338             '50': '%Rs=[0.56 1.12 1.12] ',
339             '75': '%Rs=[0.55 1.10 1.10] ',
340             '100': '%Rs=[0.54 1.09 1.09] ',
341             '167': '%Rs=[0.54 1.07 1.07] ',
342             '250': '%Rs=[0.52 1.04 1.04] ',
343             '333': '%Rs=[0.50 1.00 1.00] ',
344             '500': '%Rs=[0.50 1.00 1.00] ',
345             '667': '%Rs=[0.50 1.00 1.00] }},
346     '7.2':{
347         '0.12': {
348             '3': '%Rs=[0.95 1.90 1.90] ',
349             '5': '%Rs=[0.95 1.90 1.90] ',
350             '10': '%Rs=[0.90 1.80 1.80] ',
351             '15': '%Rs=[0.80 1.60 1.60] ',

```

```

352         '25': '%Rs=[0.71 1.42 1.42] ',
353         '37': '%Rs=[0.65 1.29 1.29] ',
354         '50': '%Rs=[0.56 1.12 1.12] ',
355         '75': '%Rs=[0.55 1.10 1.10] ',
356         '100': '%Rs=[0.54 1.09 1.09] ',
357         '167': '%Rs=[0.54 1.07 1.07] ',
358         '250': '%Rs=[0.52 1.04 1.04] ',
359         '333': '%Rs=[0.50 1.00 1.00] ',
360         '500': '%Rs=[0.50 1.00 1.00] ',
361         '667': '%Rs=[0.50 1.00 1.00] }',
362     '0.24':{
363         '3': '%Rs=[0.95 1.90 1.90] ',
364         '5': '%Rs=[0.95 1.90 1.90] ',
365         '10': '%Rs=[0.90 1.80 1.80] ',
366         '15': '%Rs=[0.80 1.60 1.60] ',
367         '25': '%Rs=[0.71 1.42 1.42] ',
368         '37': '%Rs=[0.65 1.29 1.29] ',
369         '50': '%Rs=[0.56 1.12 1.12] ',
370         '75': '%Rs=[0.55 1.10 1.10] ',
371         '100': '%Rs=[0.54 1.09 1.09] ',
372         '167': '%Rs=[0.54 1.07 1.07] ',
373         '250': '%Rs=[0.52 1.04 1.04] ',
374         '333': '%Rs=[0.50 1.00 1.00] ',
375         '500': '%Rs=[0.50 1.00 1.00] ',
376         '667': '%Rs=[0.50 1.00 1.00] }},
377     '13.8':{
378         '0.12': {
379             '3' : '%Rs=[0.95 1.90 1.90] ',
380             '5' : '%Rs=[0.95 1.90 1.90] ',
381             '10': '%Rs=[0.90 1.80 1.80] ',
382             '15': '%Rs=[0.80 1.60 1.60] ',
383             '25': '%Rs=[0.71 1.42 1.42] ',
384             '37': '%Rs=[0.65 1.29 1.29] ',
385             '50': '%Rs=[0.56 1.12 1.12] ',
386             '75': '%Rs=[0.55 1.10 1.10] ',
387             '100': '%Rs=[0.54 1.09 1.09] ',
388             '167': '%Rs=[0.54 1.07 1.07] ',
389             '250': '%Rs=[0.52 1.04 1.04] ',
390             '333': '%Rs=[0.50 1.00 1.00] ',
391             '500': '%Rs=[0.50 1.00 1.00] ',
392             '667': '%Rs=[0.50 1.00 1.00] }',
393         '0.24':{
394             '3' : '%Rs=[0.95 1.90 1.90] ',
395             '5' : '%Rs=[0.95 1.90 1.90] ',
396             '10': '%Rs=[0.90 1.80 1.80] ',
397             '15': '%Rs=[0.80 1.60 1.60] ',
398             '25': '%Rs=[0.71 1.42 1.42] ',
399             '37': '%Rs=[0.65 1.29 1.29] ',
400             '50': '%Rs=[0.56 1.12 1.12] ',
401             '75': '%Rs=[0.55 1.10 1.10] ',
402             '100': '%Rs=[0.54 1.09 1.09] ',
403             '167': '%Rs=[0.54 1.07 1.07] ',
404             '250': '%Rs=[0.52 1.04 1.04] ',
405             '333': '%Rs=[0.50 1.00 1.00] ',
406             '500': '%Rs=[0.50 1.00 1.00] ',
407             '667': '%Rs=[0.50 1.00 1.00] }},

```

```

408
409 '24.9':{
410     '0.12': {
411         '3': '%Rs=[1.05 2.10 2.10] ',
412         '5': '%Rs=[1.05 2.10 2.10] ',
413         '10': '%Rs=[0.95 1.90 1.90] ',
414         '15': '%Rs=[0.88 1.77 1.77] ',
415         '25': '%Rs=[0.77 1.54 1.54] ',
416         '37': '%Rs=[0.71 1.41 1.41] ',
417         '50': '%Rs=[0.68 1.35 1.35] ',
418         '75': '%Rs=[0.60 1.20 1.20] ',
419         '100': '%Rs=[0.55 1.11 1.11] ',
420         '167': '%Rs=[0.50 1.00 1.00] ',
421         '250': '%Rs=[0.50 1.00 1.00] ',
422         '333': '%Rs=[0.50 1.00 1.00] ',
423         '667': '%Rs=[0.50 1.00 1.00] ',
424         '500': '%Rs=[0.50 1.00 1.00] }},
425     '0.24':{
426         '3': '%Rs=[1.05 2.10 2.10] ',
427         '5': '%Rs=[1.05 2.10 2.10] ',
428         '10': '%Rs=[0.95 1.90 1.90] ',
429         '15': '%Rs=[0.88 1.77 1.77] ',
430         '25': '%Rs=[0.77 1.54 1.54] ',
431         '37': '%Rs=[0.71 1.41 1.41] ',
432         '50': '%Rs=[0.68 1.35 1.35] ',
433         '75': '%Rs=[0.60 1.20 1.20] ',
434         '100': '%Rs=[0.55 1.11 1.11] ',
435         '167': '%Rs=[0.50 1.00 1.00] ',
436         '250': '%Rs=[0.50 1.00 1.00] ',
437         '333': '%Rs=[0.50 1.00 1.00] ',
438         '667': '%Rs=[0.50 1.00 1.00] ',
439         '500': '%Rs=[0.50 1.00 1.00] }},
440 '34.5':{
441     '0.12': {
442         '3': '%Rs=[1.05 2.10 2.10] ',
443         '5': '%Rs=[1.05 2.10 2.10] ',
444         '10': '%Rs=[0.98 1.95 1.95] ',
445         '15': '%Rs=[0.92 1.83 1.83] ',
446         '25': '%Rs=[0.81 1.62 1.62] ',
447         '37': '%Rs=[0.73 1.45 1.45] ',
448         '50': '%Rs=[0.68 1.36 1.36] ',
449         '75': '%Rs=[0.61 1.21 1.21] ',
450         '100': '%Rs=[0.56 1.12 1.12] ',
451         '167': '%Rs=[0.50 1.00 1.00] ',
452         '250': '%Rs=[0.50 1.00 1.00] ',
453         '333': '%Rs=[0.50 1.00 1.00] ',
454         '500': '%Rs=[0.50 1.00 1.00] ',
455         '667': '%Rs=[0.50 1.00 1.00] ',
456         '833': '%Rs=[0.50 1.00 1.00] }},
457     '0.24':{
458         '3': '%Rs=[1.05 2.10 2.10] ',
459         '5': '%Rs=[1.05 2.10 2.10] ',
460         '10': '%Rs=[0.98 1.95 1.95] ',
461         '15': '%Rs=[0.92 1.83 1.83] ',
462         '25': '%Rs=[0.81 1.62 1.62] ',
463         '37': '%Rs=[0.73 1.45 1.45] ',

```

```

485         '50': '%Rs=[0.68 1.36 1.36]',
486         '75': '%Rs=[0.61 1.21 1.21]',
487         '100': '%Rs=[0.56 1.12 1.12]',
488         '167': '%Rs=[0.50 1.00 1.00]',
489         '250': '%Rs=[0.50 1.00 1.00]',
490         '333': '%Rs=[0.50 1.00 1.00]',
491         '500': '%Rs=[0.50 1.00 1.00]',
492         '667': '%Rs=[0.50 1.00 1.00]',
493         '833': '%Rs=[0.50 1.00 1.00]'}},
494
495     '0.48':{
496         '3': '%Rs=[1.05 2.10 2.10]',
497         '5': '%Rs=[1.05 2.10 2.10]',
498         '10': '%Rs=[0.98 1.95 1.95]',
499         '15': '%Rs=[0.92 1.83 1.83]',
500         '25': '%Rs=[0.81 1.62 1.62]',
501         '37': '%Rs=[0.73 1.45 1.45]',
502         '50': '%Rs=[0.68 1.36 1.36]',
503         '75': '%Rs=[0.61 1.21 1.21]',
504         '100': '%Rs=[0.56 1.12 1.12]',
505         '167': '%Rs=[0.50 1.00 1.00]',
506         '250': '%Rs=[0.50 1.00 1.00]',
507         '333': '%Rs=[0.50 1.00 1.00]',
508         '500': '%Rs=[0.50 1.00 1.00]',
509         '667': '%Rs=[0.50 1.00 1.00]',
510         '833': '%Rs=[0.50 1.00 1.00]'}]}
511
512 imp_list1F2W={
513     '13.8':{# List of impedances for single phase
514         '0.12': {
515             '3': 'Xhl=0.69 %Rs=[0.95 0.95]',
516             '5': 'Xhl=0.69 %Rs=[0.95 0.95]',
517             '10': 'Xhl=1.28 %Rs=[0.90 0.90]',
518             '15': 'Xhl=0.99 %Rs=[0.80 0.80]',
519             '25': 'Xhl=1.24 %Rs=[0.71 0.71]',
520             '37': 'Xhl=1.42 %Rs=[0.65 0.65]',
521             '50': 'Xhl=1.55 %Rs=[0.56 0.56]',
522             '75': 'Xhl=1.50 %Rs=[0.55 0.55]',
523             '100': 'Xhl=1.79 %Rs=[0.54 0.54]',
524             '167': 'Xhl=1.96 %Rs=[0.54 0.54]',
525             '250': 'Xhl=2.64 %Rs=[0.54 0.54]',
526             '333': 'Xhl=2.93 %Rs=[0.54 0.54]',
527             '500': 'Xhl=3.00 %Rs=[0.54 0.54]'}},
528         '0.24':{
529             '3': 'Xhl=1.09 %Rs=[0.95 0.95]',
530             '5': 'Xhl=1.09 %Rs=[0.95 0.95]',
531             '10': 'Xhl=1.17 %Rs=[0.90 0.90]',
532             '15': 'Xhl=0.99 %Rs=[0.80 0.80]',
533             '25': 'Xhl=1.24 %Rs=[0.71 0.71]',
534             '37': 'Xhl=1.42 %Rs=[0.65 0.65]',
535             '50': 'Xhl=1.55 %Rs=[0.56 0.56]',
536             '75': 'Xhl=1.50 %Rs=[0.55 0.55]',
537             '100': 'Xhl=1.62 %Rs=[0.54 0.54]',
538             '167': 'Xhl=1.79 %Rs=[0.54 0.54]',
539             '250': 'Xhl=2.24 %Rs=[0.54 0.54]',

```

```

540          '333': 'Xhl=2.22 %Rs=[0.54 0.54]',
541          '500': 'Xhl=2.3 %Rs=[0.54 0.54]'}},
542
543 '24.9':{
544     '0.12': {
545         '3': 'Xhl=0.69 %Rs=[1.05 1.05]',
546         '5': 'Xhl=0.69 %Rs=[1.05 1.05]',
547         '10': 'Xhl=1.28 %Rs=[0.95 0.95]',
548         '15': 'Xhl=0.99 %Rs=[0.88 0.88]',
549         '25': 'Xhl=1.24 %Rs=[0.77 0.77]',
550         '37': 'Xhl=1.42 %Rs=[0.71 0.71]',
551         '50': 'Xhl=1.55 %Rs=[0.68 0.68]',
552         '75': 'Xhl=1.50 %Rs=[0.60 0.60]',
553         '100': 'Xhl=1.79 %Rs=[0.55 0.55]',
554         '167': 'Xhl=1.96 %Rs=[0.50 0.50]',
555         '250': 'Xhl=2.64 %Rs=[0.50 0.50]',
556         '333': 'Xhl=2.93 %Rs=[0.50 0.50]',
557         '500': 'Xhl=3.00 %Rs=[0.50 0.50]'}},
558     '0.24':{
559         '3': 'Xhl=1.09 %Rs=[1.05 1.05]',
560         '5': 'Xhl=1.09 %Rs=[1.05 1.05]',
561         '10': 'Xhl=1.17 %Rs=[0.95 0.95]',
562         '15': 'Xhl=0.99 %Rs=[0.88 0.88]',
563         '25': 'Xhl=1.24 %Rs=[0.77 0.77]',
564         '37': 'Xhl=1.42 %Rs=[0.71 0.71]',
565         '50': 'Xhl=1.55 %Rs=[0.68 0.68]',
566         '75': 'Xhl=1.50 %Rs=[0.60 0.60]',
567         '100': 'Xhl=1.62 %Rs=[0.55 0.55]',
568         '167': 'Xhl=1.79 %Rs=[0.50 0.50]',
569         '250': 'Xhl=2.24 %Rs=[0.50 0.50]',
570         '333': 'Xhl=2.22 %Rs=[0.50 0.50]',
571         '500': 'Xhl=2.3 %Rs=[0.50 0.50]'}},
572
573 '34.5':{
574     '0.12': {
575         '3': 'Xhl=4.10 %Rs=[1.05 1.05]',
576         '5': 'Xhl=4.10 %Rs=[1.05 1.05]',
577         '10': 'Xhl=4.45 %Rs=[0.98 0.98]',
578         '15': 'Xhl=4.60 %Rs=[0.92 0.92]',
579         '25': 'Xhl=4.71 %Rs=[0.81 0.81]',
580         '37': 'Xhl=4.83 %Rs=[0.73 0.73]',
581         '50': 'Xhl=4.91 %Rs=[0.68 0.68]',
582         '75': 'Xhl=4.95 %Rs=[0.61 0.61]',
583         '100': 'Xhl=4.98 %Rs=[0.56 0.56]',
584         '167': 'Xhl=5.03 %Rs=[0.50 0.50]',
585         '250': 'Xhl=5.06 %Rs=[0.50 0.50]',
586         '333': 'Xhl=5.08 %Rs=[0.50 0.50]',
587         '500': 'Xhl=5.10 %Rs=[0.50 0.50]',
588         '833': 'Xhl=5.10 %Rs=[0.50 0.50]'}},
589
590     '0.24':{
591         '3': 'Xhl=4.10 %Rs=[1.05 1.05]',
592         '5': 'Xhl=4.10 %Rs=[1.05 1.05]',
593         '10': 'Xhl=4.45 %Rs=[0.98 0.98]',
594         '15': 'Xhl=4.60 %Rs=[0.92 0.92]',
595         '25': 'Xhl=4.71 %Rs=[0.81 0.81]',

```

```

596         '37': 'Xhl=4.83 %Rs=[0.73 0.73]',
597         '50': 'Xhl=4.91 %Rs=[0.68 0.68]',
598         '75': 'Xhl=4.95 %Rs=[0.61 0.61]',
599         '100': 'Xhl=4.98 %Rs=[0.56 0.56]',
600         '167': 'Xhl=5.03 %Rs=[0.50 0.50]',
601         '250': 'Xhl=5.06 %Rs=[0.50 0.50]',
602         '333': 'Xhl=5.08 %Rs=[0.50 0.50]',
603         '500': 'Xhl=5.10 %Rs=[0.50 0.50]',
604         '667': 'Xhl=5.10 %Rs=[0.50 0.50]'},
605     '0.48':{
606         '3': 'Xhl=4.10 %Rs=[1.05 1.05]',
607         '5': 'Xhl=4.10 %Rs=[1.05 1.05]',
608         '10': 'Xhl=4.45 %Rs=[0.98 0.98]',
609         '15': 'Xhl=4.60 %Rs=[0.92 0.92]',
610         '25': 'Xhl=4.71 %Rs=[0.81 0.81]',
611         '37': 'Xhl=4.83 %Rs=[0.73 0.73]',
612         '50': 'Xhl=4.91 %Rs=[0.68 0.68]',
613         '75': 'Xhl=4.95 %Rs=[0.61 0.61]',
614         '100': 'Xhl=4.98 %Rs=[0.56 0.56]',
615         '167': 'Xhl=5.03 %Rs=[0.50 0.50]',
616         '250': 'Xhl=5.06 %Rs=[0.50 0.50]',
617         '333': 'Xhl=5.08 %Rs=[0.50 0.50]',
618         '500': 'Xhl=5.10 %Rs=[0.50 0.50]',
619         '667': 'Xhl=5.10 %Rs=[0.50 0.50]'}]}
620
621 imp_list1F_mv_mv={ # List of impedances for single
        phase two winding transformers [MV][LV][Power]
622 '13.8':{
623     '2.4':{}}},
624
625 '34.5':{
626
627     '14.38': {
628         '3': 'Xhl=1 %Rs=[0.1 0.1]',
629         '5': 'Xhl=1 %Rs=[0.1 0.1]',
630         '10': 'Xhl=1 %Rs=[0.1 0.1]',
631         '15': 'Xhl=1 %Rs=[0.1 0.1]',
632         '25': 'Xhl=1 %Rs=[0.1 0.1]',
633         '37': 'Xhl=1 %Rs=[0.1 0.1]',
634         '50': 'Xhl=1 %Rs=[0.1 0.1]',
635         '75': 'Xhl=1 %Rs=[0.1 0.1]',
636         '100': 'Xhl=1 %Rs=[0.1 0.1]',
637         '167': 'Xhl=1 %Rs=[0.1 0.1]',
638         '250': 'Xhl=1 %Rs=[0.1 0.1]',
639         '333': 'Xhl=1 %Rs=[0.1 0.1]',
640         '500': 'Xhl=1 %Rs=[0.1 0.1]'},
641
642     '7.2': {
643         '3': 'Xhl=1 %Rs=[0.1 0.1]',
644         '5': 'Xhl=1 %Rs=[0.1 0.1]',
645         '10': 'Xhl=1 %Rs=[0.1 0.1]',
646         '15': 'Xhl=1 %Rs=[0.1 0.1]',
647         '25': 'Xhl=1 %Rs=[0.1 0.1]',
648         '37': 'Xhl=1 %Rs=[0.1 0.1]',
649         '50': 'Xhl=1 %Rs=[0.1 0.1]',
650         '75': 'Xhl=1 %Rs=[0.1 0.1]',

```

```

651         '100': 'Xhl=1 %Rs=[0.1 0.1]',
652         '167': 'Xhl=1 %Rs=[0.1 0.1]',
653         '250': 'Xhl=1 %Rs=[0.1 0.1]',
654         '333': 'Xhl=1 %Rs=[0.1 0.1]',
655         '500': 'Xhl=1 %Rs=[0.1 0.1]'},
656
657     '7.97': {
658         '3': 'Xhl=1 %Rs=[0.1 0.1]',
659         '5': 'Xhl=1 %Rs=[0.1 0.1]',
660         '10': 'Xhl=1 %Rs=[0.1 0.1]',
661         '15': 'Xhl=1 %Rs=[0.1 0.1]',
662         '25': 'Xhl=1 %Rs=[0.1 0.1]',
663         '37': 'Xhl=1 %Rs=[0.1 0.1]',
664         '50': 'Xhl=1 %Rs=[0.1 0.1]',
665         '75': 'Xhl=1 %Rs=[0.1 0.1]',
666         '100': 'Xhl=1 %Rs=[0.1 0.1]',
667         '167': 'Xhl=1 %Rs=[0.1 0.1]',
668         '250': 'Xhl=1 %Rs=[0.1 0.1]',
669         '333': 'Xhl=1 %Rs=[0.1 0.1]',
670         '500': 'Xhl=1 %Rs=[0.1 0.1]'}},
671
672     '2.4': {
673         '3': 'Xhl=1 %Rs=[0.1 0.1]',
674         '5': 'Xhl=1 %Rs=[0.1 0.1]',
675         '10': 'Xhl=1 %Rs=[0.1 0.1]',
676         '15': 'Xhl=1 %Rs=[0.1 0.1]',
677         '25': 'Xhl=1 %Rs=[0.1 0.1]',
678         '37': 'Xhl=1 %Rs=[0.1 0.1]',
679         '50': 'Xhl=1 %Rs=[0.1 0.1]',
680         '75': 'Xhl=1 %Rs=[0.1 0.1]',
681         '100': 'Xhl=1 %Rs=[0.1 0.1]',
682         '167': 'Xhl=1 %Rs=[0.1 0.1]',
683         '250': 'Xhl=1 %Rs=[0.1 0.1]',
684         '333': 'Xhl=1 %Rs=[0.1 0.1]',
685         '500': 'Xhl=1 %Rs=[0.1 0.1]'}]}}
686
687 noloadloss_list1F={
688     '3': '%noloadloss=0.70',
689     '5': '%noloadloss=0.70',
690     '10': '%noloadloss=0.50',
691     '15': '%noloadloss=0.47',
692     '25': '%noloadloss=0.36',
693     '37': '%noloadloss=0.35',
694     '50': '%noloadloss=0.34',
695     '75': '%noloadloss=0.27',
696     '100': '%noloadloss=0.25',
697     '167': '%noloadloss=0.24',
698     '250': '%noloadloss=0.23',
699     '333': '%noloadloss=0.23',
700     '500': '%noloadloss=0.22',
701     '667': '%noloadloss=0.22',
702     '833': '%noloadloss=0.22'}
703
704 imag_list1F={
705     '3': '%imag=1.00',
706     '5': '%imag=1.00',

```

```

707     '10': '%imag=1.00',
708     '15': '%imag=1.00',
709     '25': '%imag=1.00',
710     '37': '%imag=1.00',
711     '50': '%imag=1.00',
712     '75': '%imag=1.00',
713     '100': '%imag=1.00',
714     '167': '%imag=1.00',
715     '250': '%imag=1.00',
716     '333': '%imag=1.00',
717     '500': '%imag=1.00',
718     '667': '%imag=1.00',
719     '833': '%imag=1.00'}
720
721 imp_list3F={
722
723     '2.4':{
724         '9': 'Xhl=2.84 %Rs=[0.98 0.98]',
725         '30': 'Xhl=2.28 %Rs=[0.83 0.83]',
726         '45': 'Xhl=2.08 %Rs=[0.76 0.76]',
727         '75': 'Xhl=1.65 %Rs=[0.67 0.67]',
728         '112': 'Xhl=1.30 %Rs=[0.61 0.61]',
729         '150': 'Xhl=1.55 %Rs=[0.55 0.55]',
730         '225': 'Xhl=1.55 %Rs=[0.56 0.56]',
731         '300': 'Xhl=1.67 %Rs=[0.50 0.50]',
732         '500': 'Xhl=2.07 %Rs=[0.50 0.50]',
733         '750': 'Xhl=5.59 %Rs=[0.48 0.48]',
734         '1000': 'Xhl=5.61 %Rs=[0.46 0.46]',
735         '1500': 'Xhl=5.59 %Rs=[0.36 0.36]',
736         '2500': 'Xhl=5.59 %Rs=[0.36 0.36]',
737         '3000': 'Xhl=5.59 %Rs=[0.36 0.36]',
738         '3750': 'Xhl=5.59 %Rs=[0.30 0.30]',
739         '5000': 'Xhl=5.59 %Rs=[0.30 0.30]',
740         '7500': 'Xhl=5.59 %Rs=[0.27 0.27]',
741         '10000': 'Xhl=5.59 %Rs=[0.24 0.24]',
742         '15000': 'Xhl=5.59 %Rs=[0.22 0.22]',
743         '20000': 'Xhl=5.59 %Rs=[0.20 0.20]',
744         '25000': 'Xhl=5.59 %Rs=[0.19 0.19]'},
745     '4.16':{
746         '9': 'Xhl=2.84 %Rs=[0.98 0.98]',
747         '30': 'Xhl=2.28 %Rs=[0.83 0.83]',
748         '45': 'Xhl=2.08 %Rs=[0.76 0.76]',
749         '75': 'Xhl=1.65 %Rs=[0.67 0.67]',
750         '112': 'Xhl=1.30 %Rs=[0.61 0.61]',
751         '150': 'Xhl=1.55 %Rs=[0.55 0.55]',
752         '225': 'Xhl=1.55 %Rs=[0.56 0.56]',
753         '300': 'Xhl=1.67 %Rs=[0.50 0.50]',
754         '500': 'Xhl=2.07 %Rs=[0.50 0.50]',
755         '750': 'Xhl=5.59 %Rs=[0.48 0.48]',
756         '1000': 'Xhl=5.61 %Rs=[0.46 0.46]',
757         '1500': 'Xhl=5.59 %Rs=[0.40 0.40]',
758         '2500': 'Xhl=5.59 %Rs=[0.36 0.36]',
759         '3000': 'Xhl=5.59 %Rs=[0.36 0.36]',
760         '3750': 'Xhl=5.59 %Rs=[0.345 0.345]',
761         '5000': 'Xhl=5.59 %Rs=[0.30 0.30]',
762         '7500': 'Xhl=5.59 %Rs=[0.27 0.27]',

```

```

763         '10000': 'Xhl=5.59 %Rs=[0.24 0.24]',
764         '15000': 'Xhl=5.59 %Rs=[0.22 0.22]',
765         '20000': 'Xhl=5.59 %Rs=[0.20 0.20]',
766         '25000': 'Xhl=5.59 %Rs=[0.19 0.19]'},
767
768     '7.2':{
769         '9': 'Xhl=2.84 %Rs=[0.98 0.98]',
770         '30': 'Xhl=2.28 %Rs=[0.83 0.83]',
771         '45': 'Xhl=2.08 %Rs=[0.76 0.76]',
772         '75': 'Xhl=1.65 %Rs=[0.67 0.67]',
773         '112': 'Xhl=1.30 %Rs=[0.61 0.61]',
774         '150': 'Xhl=1.55 %Rs=[0.56 0.56]',
775         '225': 'Xhl=1.55 %Rs=[0.53 0.53]',
776         '300': 'Xhl=1.67 %Rs=[0.50 0.50]',
777         '500': 'Xhl=2.07 %Rs=[0.50 0.50]',
778         '750': 'Xhl=5.59 %Rs=[0.48 0.48]',
779         '1000': 'Xhl=5.61 %Rs=[0.46 0.46]',
780         '1500': 'Xhl=5.59 %Rs=[0.40 0.40]',
781         '2500': 'Xhl=5.59 %Rs=[0.36 0.36]',
782         '3000': 'Xhl=5.59 %Rs=[0.36 0.36]',
783         '3750': 'Xhl=5.59 %Rs=[0.345 0.345]',
784         '5000': 'Xhl=5.59 %Rs=[0.30 0.30]',
785         '7500': 'Xhl=5.59 %Rs=[0.27 0.27]',
786         '10000': 'Xhl=5.59 %Rs=[0.24 0.24]',
787         '15000': 'Xhl=5.59 %Rs=[0.22 0.22]',
788         '20000': 'Xhl=5.59 %Rs=[0.20 0.20]',
789         '25000': 'Xhl=5.59 %Rs=[0.19 0.19]'},
790
791     '13.8':{
792         '9': 'Xhl=2.84 %Rs=[0.98 0.98]',
793         '30': 'Xhl=2.28 %Rs=[0.83 0.83]',
794         '75': 'Xhl=1.65 %Rs=[0.67 0.67]',
795         '112': 'Xhl=1.30 %Rs=[0.61 0.61]',
796         '150': 'Xhl=1.55 %Rs=[0.56 0.56]',
797         '225': 'Xhl=1.55 %Rs=[0.53 0.53]',
798         '300': 'Xhl=1.67 %Rs=[0.52 0.52]',
799         '500': 'Xhl=2.07 %Rs=[0.50 0.50]',
800         '750': 'Xhl=5.59 %Rs=[0.48 0.48]',
801         '1000': 'Xhl=5.61 %Rs=[0.46 0.46]',
802         '1500': 'Xhl=5.59 %Rs=[0.40 0.40]',
803         '2500': 'Xhl=5.59 %Rs=[0.36 0.36]',
804         '3000': 'Xhl=5.59 %Rs=[0.36 0.36]',
805         '3750': 'Xhl=5.59 %Rs=[0.345 0.345]',
806         '5000': 'Xhl=5.59 %Rs=[0.30 0.30]',
807         '7500': 'Xhl=5.59 %Rs=[0.27 0.27]',
808         '10000': 'Xhl=5.59 %Rs=[0.24 0.24]',
809         '15000': 'Xhl=5.59 %Rs=[0.22 0.22]',
810         '20000': 'Xhl=5.59 %Rs=[0.20 0.20]',
811         '25000': 'Xhl=5.59 %Rs=[0.19 0.19]'},
812
813     '24.9':{
814         '9': 'Xhl=2.84 %Rs=[1.03 1.03]',
815         '30': 'Xhl=2.28 %Rs=[0.88 0.88]',
816         '45': 'Xhl=2.08 %Rs=[0.81 0.81]',
817         '75': 'Xhl=1.65 %Rs=[0.72 0.72]',
818         '112': 'Xhl=1.30 %Rs=[0.63 0.63]',

```

```

819         '150': 'Xhl=1.55 %Rs=[0.60 0.60]',
820         '225': 'Xhl=1.55 %Rs=[0.55 0.55]',
821         '300': 'Xhl=1.67 %Rs=[0.52 0.52]',
822         '500': 'Xhl=2.07 %Rs=[0.50 0.50]',
823         '750': 'Xhl=5.59 %Rs=[0.48 0.48]',
824         '1000': 'Xhl=5.49 %Rs=[0.455 0.455]',
825         '1500': 'Xhl=5.50 %Rs=[0.40 0.40]',
826         '2500': 'Xhl=5.51 %Rs=[0.36 0.36]',
827         '3000': 'Xhl=5.51 %Rs=[0.36 0.36]',
828         '3750': 'Xhl=4.95 %Rs=[0.345 0.345]',
829         '5000': 'Xhl=5.62 %Rs=[0.30 0.30]',
830         '7500': 'Xhl=5.66 %Rs=[0.27 0.27]',
831         '10000': 'Xhl=5.70 %Rs=[0.24 0.24]',
832         '15000': 'Xhl=5.75 %Rs=[0.22 0.22]',
833         '20000': 'Xhl=5.78 %Rs=[0.20 0.20]',
834         '25000': 'Xhl=5.81 %Rs=[0.19 0.19]'}},
835
836     '34.5':{
837         '9': 'Xhl=3.96 %Rs=[1.10 1.10]',
838         '30': 'Xhl=4.80 %Rs=[0.93 0.93]',
839         '45': 'Xhl=4.96 %Rs=[0.86 0.86]',
840         '75': 'Xhl=5.11 %Rs=[0.77 0.77]',
841         '112': 'Xhl=5.17 %Rs=[0.66 0.66]',
842         '150': 'Xhl=5.24 %Rs=[0.64 0.64]',
843         '225': 'Xhl=5.30 %Rs=[0.58 0.58]',
844         '300': 'Xhl=5.32 %Rs=[0.54 0.54]',
845         '500': 'Xhl=5.37 %Rs=[0.52 0.52]',
846         '750': 'Xhl=5.40 %Rs=[0.518 0.518]',
847         '1000': 'Xhl=5.49 %Rs=[0.455 0.455]',
848         '1500': 'Xhl=5.50 %Rs=[0.40 0.40]',
849         '2000': 'Xhl=5.51 %Rs=[0.36 0.36]',
850         '2500': 'Xhl=5.51 %Rs=[0.36 0.36]',
851         '3000': 'Xhl=5.51 %Rs=[0.36 0.36]',
852         '3750': 'Xhl=4.95 %Rs=[0.345 0.345]',
853         '5000': 'Xhl=5.62 %Rs=[0.30 0.30]',
854         '7500': 'Xhl=5.66 %Rs=[0.27 0.27]',
855         '10000': 'Xhl=5.70 %Rs=[0.24 0.24]',
856         '15000': 'Xhl=5.75 %Rs=[0.22 0.22]',
857         '20000': 'Xhl=5.78 %Rs=[0.20 0.20]',
858         '25000': 'Xhl=5.81 %Rs=[0.19 0.19]'}}}
859
860     noloadloss_list3F={
861         '9': '%noloadloss=0.53',
862         '30': '%noloadloss=0.47',
863         '45': '%noloadloss=0.41',
864         '75': '%noloadloss=0.36',
865         '112': '%noloadloss=0.33',
866         '150': '%noloadloss=0.30',
867         '225': '%noloadloss=0.28',
868         '300': '%noloadloss=0.25',
869         '500': '%noloadloss=0.24',
870         '750': '%noloadloss=0.23',
871         '1000': '%noloadloss=0.21',
872         '1500': '%noloadloss=0.19',
873         '2000': '%noloadloss=0.17',
874         '2500': '%noloadloss=0.17',

```

```

875     '3000': '%noloadloss=0.17',
876     '3750': '%noloadloss=0.15',
877     '5000': '%noloadloss=0.14',
878     '7500': '%noloadloss=0.12',
879     '10000': '%noloadloss=0.11',
880     '15000': '%noloadloss=0.10',
881     '20000': '%noloadloss=0.09',
882     '25000': '%noloadloss=0.08'}
883
884 imag_list3F={
885     '9': '%imag=1.00',
886     '30': '%imag=1.00',
887     '45': '%imag=1.00',
888     '75': '%imag=1.00',
889     '112': '%imag=1.00',
890     '150': '%imag=1.00',
891     '225': '%imag=1.00',
892     '300': '%imag=1.00',
893     '500': '%imag=1.00',
894     '750': '%imag=1.00',
895     '1000': '%imag=1.00',
896     '1500': '%imag=1.00',
897     '2000': '%imag=1.00',
898     '2500': '%imag=1.00',
899     '3000': '%imag=1.00',
900     '3750': '%imag=1.00',
901     '5000': '%imag=1.00',
902     '7500': '%imag=1.00',
903     '10000': '%imag=1.00',
904     '15000': '%imag=1.00',
905     '20000': '%imag=1.00',
906     '25000': '%imag=1.00'}
907
908 TrafoAsymOrig=pd.read_excel('CircuitoLimpio.xlsx',
sheet_name='Trafo2WindingAsym', decimal='.')
909 TrafoOrig=pd.read_excel('CircuitoLimpio.xlsx',
sheet_name='Trafo2Winding', decimal='.')
910
911 TrafoMonoOrig=pd.DataFrame()
912 TrafoBiOrig=pd.DataFrame()
913 for k in range(len(TrafoAsymOrig)):
914     if TrafoAsymOrig.iloc[k,6].startswith('AB') == True or
TrafoAsymOrig.iloc[k,6].startswith('AC') == True or
TrafoAsymOrig.iloc[k,6].startswith('BC') == True:
915         TrafoBiOrig = TrafoBiOrig.append(TrafoAsymOrig.iloc[k])
916         TrafoBiOrig = TrafoBiOrig.append(TrafoAsymOrig.iloc[k])
917     else:
918         TrafoMonoOrig = TrafoMonoOrig.append(TrafoAsymOrig.iloc[k])
919
920 for k in range(len(TrafoBiOrig)):
921     if k%2 == 0:
922         TrafoBiOrig.iat[k,0]=
TrafoBiOrig.iloc[k,0]+'_'+TrafoBiOrig.iloc[k,6][0]
923     else:
924         TrafoBiOrig.iat[k,0]=
TrafoBiOrig.iloc[k,0]+'_'+TrafoBiOrig.iloc[k,6][1]

```

```

925
926 Trafo=pd.DataFrame(columns=['Name', 'phases', 'windings',
    '%noloadloss', '%imag', 'buses', 'conns', 'kvs', 'kvas',
    'Xhl', '%Rs'], index=range(len(TrafoOrig)))
927 TrafoMono=pd.DataFrame(columns=['Name', 'phases', 'windings',
    '%noloadloss', '%imag', 'buses', 'conns', 'kvs', 'kvas',
    'Xhl', '%Rs'], index=range(len(TrafoMonoOrig)))
928 TrafoBi=pd.DataFrame(columns=['Name', 'phases', 'windings',
    '%noloadloss', '%imag', 'buses', 'conns', 'kvs', 'kvas',
    'Xhl', '%Rs'], index=range(len(TrafoBiOrig)))
929
930 for k in range(len(TrafoMonoOrig)):
931     TrafoMono.iat[k,0]=
    ('New transformer.'+TrafoMonoOrig.iloc[k,0])
932     TrafoMono.iat[k,1]='phases=1'
933     TrafoMono.iat[k,2]='windings=3'
934     TrafoMono.iat[k,3]=
    noloadloss_list1F[str(int(TrafoMonoOrig.iloc[k,5]))]
935     TrafoMono.iat[k,4]=
    imag_list1F[str(int(TrafoMonoOrig.iloc[k,5]))]
936     if TrafoMonoOrig.iloc[k,6].startswith('A'):
937         TrafoMono.iat[k,5]='buses=['
            +TrafoMonoOrig.iloc[k,1]+' .1.0 '
            +TrafoMonoOrig.iloc[k,2]+' .1.0 '
            +TrafoMonoOrig.iloc[k,2]+' .0.2]'
938     if TrafoMonoOrig.iloc[k,6].startswith('B'):
939         TrafoMono.iat[k,5]='buses=['
            +TrafoMonoOrig.iloc[k,1]+' .2.0 '
            +TrafoMonoOrig.iloc[k,2]+' .1.0 '
            +TrafoMonoOrig.iloc[k,2]+' .0.2]'
940     if TrafoMonoOrig.iloc[k,6].startswith('C'):
941         TrafoMono.iat[k,5]='buses=['
            +TrafoMonoOrig.iloc[k,1]+' .3.0 '
            +TrafoMonoOrig.iloc[k,2]+' .1.0 '
            +TrafoMonoOrig.iloc[k,2]+' .0.2]'
942     TrafoMono.iat[k,6]='conns=[wye wye wye]'
943     TrafoMono.iat[k,7]='kvs=['
    +str(TrafoMonoOrig.iloc[k,3]/math.sqrt(3))+
    ' '+str(TrafoMonoOrig.iloc[k,4]/2)
    +' '+str(TrafoMonoOrig.iloc[k,4]/2)+'']')
944     TrafoMono.iat[k,8]='kvas=['+
    str(TrafoMonoOrig.iloc[k,5])
    +' '+str(TrafoMonoOrig.iloc[k,5])
    +' '+str(TrafoMonoOrig.iloc[k,5])+']')
945     TrafoMono.iat[k,9]=
    react_list1F3W[str(TrafoMonoOrig.iloc[k,3])]
    [str(TrafoMonoOrig.iloc[k,4])]
    [str(int(TrafoMonoOrig.iloc[k,5]))]
946     TrafoMono.iat[k,10]=
    res_list1F3W[str(TrafoMonoOrig.iloc[k,3])]
    [str(TrafoMonoOrig.iloc[k,4])]
    [str(int(TrafoMonoOrig.iloc[k,5]))]
947 TrafoMono.to_csv('Trafo1F.dss', sep=' ', index=False,
    header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
948
949 for k in range(len(TrafoBiOrig)):

```

```

950     TrafoBi.iat[k,0]=
        ('New transformer.'+TrafoBiOrig.iloc[k,0])
951     TrafoBi.iat[k,1]='phases=1'
952     TrafoBi.iat[k,2]='windings=2'
953     TrafoBi.iat[k,3]=
        noloadloss_list1F[str(int(TrafoBiOrig.iloc[k,5]))]
954     TrafoBi.iat[k,4]=
        imag_list1F[str(int(TrafoBiOrig.iloc[k,5]))]
956     if k%2 == 0:
957         if TrafoBiOrig.iloc[k,0].endswith('A'):
958             TrafoBi.iat[k,5]='buses=[
                +TrafoBiOrig.iloc[k,1]+' .1.0 '
                +TrafoBiOrig.iloc[k,2]+' .2.1]'
959         if TrafoBiOrig.iloc[k,0].endswith('B'):
960             TrafoBi.iat[k,5]='buses=[
                +TrafoBiOrig.iloc[k,1]+' .2.0 '
                +TrafoBiOrig.iloc[k,2]+' .2.1]'
961         if TrafoBiOrig.iloc[k,0].endswith('C'):
962             TrafoBi.iat[k,5]='buses=[
                +TrafoBiOrig.iloc[k,1]+' .3.0 '
                +TrafoBiOrig.iloc[k,2]+' .2.1]'
963     else:
964         if TrafoBiOrig.iloc[k,0].endswith('A'):
965             TrafoBi.iat[k,5]='buses=[
                +TrafoBiOrig.iloc[k,1]+' .1.0 '
                +TrafoBiOrig.iloc[k,2]+' .3.2]'
966         if TrafoBiOrig.iloc[k,0].endswith('B'):
967             TrafoBi.iat[k,5]='buses=[
                +TrafoBiOrig.iloc[k,1]+' .2.0 '
                +TrafoBiOrig.iloc[k,2]+' .3.2]'
968         if TrafoBiOrig.iloc[k,0].endswith('C'):
969             TrafoBi.iat[k,5]='buses=[
                +TrafoBiOrig.iloc[k,1]+' .3.0 '
                +TrafoBiOrig.iloc[k,2]+' .3.2]'
970     TrafoBi.iat[k,6]='conns=[wye wye]'
971     TrafoBi.iat[k,7]='kvs=[
        +str(TrafoBiOrig.iloc[k,3]/math.sqrt(3))
        +' '+str(TrafoBiOrig.iloc[k,4])+']')
972     TrafoBi.iat[k,8]='kvas=[
        +str(TrafoBiOrig.iloc[k,5])
        +' '+str(TrafoBiOrig.iloc[k,5])+']')
973     TrafoBi.iat[k,9]=
        imp_list1F2W[str(TrafoBiOrig.iloc[0,3])
        [str(TrafoBiOrig.iloc[0,4])]
        [str(int(TrafoBiOrig.iloc[0,5]))][:8]
974     TrafoBi.iat[k,10]=
        imp_list1F2W[str(TrafoBiOrig.iloc[0,3])
        [str(TrafoBiOrig.iloc[0,4])]
        [str(int(TrafoBiOrig.iloc[0,5]))][9:]
975     TrafoBi.to_csv('Trafo2F.dss', sep=' ', index=False,
        header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
976
977 for k in range(len(TrafoOrig)):
978     Trafo.iat[k,0]='New transformer.'+TrafoOrig.iloc[k,0])
979     Trafo.iat[k,1]='phases=3'
980     Trafo.iat[k,2]='windings=2'

```

```

981     if TrafoOrig.iloc[k,5] == 900:
982         Trafo.iat[k,3]=noloadloss_list3F['1000']
983         Trafo.iat[k,4]=imag_list3F['1000']
984     else:
985         Trafo.iat[k,3]=
986             noloadloss_list3F[str(int(TrafoOrig.iloc[k,5]))]
987         Trafo.iat[k,4]=
988             imag_list3F[str(int(TrafoOrig.iloc[k,5]))]
989         Trafo.iat[k,5]='buses=['+TrafoOrig.iloc[k,1]+' .1.2.3.4 '
990             +TrafoOrig.iloc[k,2]+' .1.2.3]'
991         Trafo.iat[k,6]='conns=[wye delta]'
992         Trafo.iat[k,7]='kvs=['+str(TrafoOrig.iloc[k,3])+ ' '
993             +str(TrafoOrig.iloc[k,4])+']')
994         Trafo.iat[k,8]='kvas=['+str(TrafoOrig.iloc[k,5])+ ' '
995             +str(TrafoOrig.iloc[k,5])+']')
996     if TrafoOrig.iloc[k,5] == 900:
997         Trafo.iat[k,9]=
998             imp_list3F[str(TrafoOrig.iloc[k,3])][ '1000' ][:8]
999         Trafo.iat[k,10]=
1000             imp_list3F[str(TrafoOrig.iloc[k,3])][ '1000' ][9:]
1001     else:
1002         Trafo.iat[k,9]=imp_list3F[str(TrafoOrig.iloc[k,3])][
1003             str(int(TrafoOrig.iloc[k,5]))][:8]
1004         Trafo.iat[k,10]=imp_list3F[str(TrafoOrig.iloc[k,3])][
1005             str(int(TrafoOrig.iloc[k,5]))[9:]
1006 Trafo.to_csv('Trafo3F.dss', sep=' ', index=False,
1007             header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
1008
1009     LoadOrig=pd.read_excel('CircuitoLimpio.xlsx',
1010         sheet_name='Load', decimal='.')
1011     Load=pd.DataFrame(columns=['Name', 'Bus1', 'kV', 'Model',
1012         'Conn', 'kW', 'PF', 'Status', 'Phases'],
1013         index=range(len(LoadOrig)))
1014     LoadVar=pd.DataFrame(columns=['Name', 'Bus1', 'kV',
1015         'Model', 'Conn', 'kW', 'PF', 'Status', 'Phases',
1016         'Daily'],index=range(len(LoadOrig)))
1017
1018     for k in range(len(LoadOrig)):
1019         Load.iat[k,0]='New load.'+LoadOrig.iloc[k,0])
1020         Load.iat[k,1]='bus1'+LoadOrig.iloc[k,1])
1021         Load.iat[k,2]='kV'+str(LoadOrig.iloc[k,2])
1022         Load.iat[k,3]='model=1'
1023         Load.iat[k,4]='conn=wye'
1024         Load.iat[k,5]='kW'+str(LoadOrig.iloc[k,5]/720)
1025         Load.iat[k,6]='pF'+str(LoadOrig.iloc[k,6])
1026         Load.iat[k,7]='status=fixed'
1027         Load.iat[k,8]='phases=2'
1028     Load.to_csv('Load.dss', sep=' ', index=False, header=False,
1029         quoting=csv.QUOTE_NONE, escapechar=' ')
1030
1031     for k in range(len(LoadOrig)):
1032         LoadVar.iat[k,0]='New load.'+LoadOrig.iloc[k,0])
1033         LoadVar.iat[k,1]='bus1'+LoadOrig.iloc[k,1])
1034         LoadVar.iat[k,2]='kV'+str(LoadOrig.iloc[k,2])
1035         LoadVar.iat[k,3]='model=1'
1036         LoadVar.iat[k,4]='conn=wye'

```

```
1021     LoadVar.iat[k,5]='kW='+str(LoadOrig.iloc[k,5]/720))
1022     LoadVar.iat[k,6]='pF='+str(LoadOrig.iloc[k,6])
1023     LoadVar.iat[k,7]='status=variable'
1024     LoadVar.iat[k,8]='phases=2'
1025     LoadVar.iat[k,9]='daily=Perfil'
1026
1027     LoadVar.to_csv('LoadVar.dss', sep=' ', index=False,
header=False, quoting=csv.QUOTE_NONE, escapechar=' ')
```

Apéndice B

WireData de conductores

B.1. Conductores AAAC

Calibres referenciados a su equivalente en ACSR.

New WireData.AAAC_6_7STR	Rac=0.785	Runits=kft	GMR=0.0060	GMRunits=ft
DIAM=0.198	Radunits=in	Normamps=107		
New WireData.AAAC_4_7STR	Rac=0.4928	Runits=kft	GMR=0.0076	GMRunits=ft
DIAM=0.250	Radunits=in	Normamps=143		
New WireData.AAAC_2_7STR	Rac=0.3098	Runits=kft	GMR=0.0095	GMRunits=ft
DIAM=0.316	Radunits=in	Normamps=191		
New WireData.AAAC_1/0_7STR	Rac=0.1947	Runits=kft	GMR=0.0120	GMRunits=ft
DIAM=0.398	Radunits=in	Normamps=256		
New WireData.AAAC_2/0_7STR	Rac=0.1545	Runits=kft	GMR=0.0135	GMRunits=ft
DIAM=0.447	Radunits=in	Normamps=296		
New WireData.AAAC_3/0_7STR	Rac=0.1227	Runits=kft	GMR=0.0152	GMRunits=ft
DIAM=0.502	Radunits=in	Normamps=342		
New WireData.AAAC_4/0_7STR	Rac=0.0973	Runits=kft	GMR=0.0170	GMRunits=ft
DIAM=0.563	Radunits=in	Normamps=395		
New WireData.AAAC_266.8_19STR	Rac=0.0769	Runits=kft	GMR=0.0203	GMRunits=ft
DIAM=0.642	Radunits=in	Normamps=460		
New WireData.AAAC_336.4_19STR	Rac=0.0610	Runits=kft	GMR=0.0227	GMRunits=ft
DIAM=0.720	Radunits=in	Normamps=532		
New WireData.AAAC_397.5_19STR	Rac=0.0518	Runits=kft	GMR=0.0247	GMRunits=ft
DIAM=0.783	Radunits=in	Normamps=590		
New WireData.AAAC_477_19STR	Rac=0.0431	Runits=kft	GMR=0.0271	GMRunits=ft
DIAM=0.858	Radunits=in	Normamps=663		
New WireData.AAAC_556.5_19STR	Rac=0.0371	Runits=kft	GMR=0.0292	GMRunits=ft
DIAM=0.927	Radunits=in	Normamps=729		

B.2. Conductores AAC

new Wiredata.AAC_6_7STR	GMR=0.00555	DIAM=0.184	Rac=0.805
Normamps=103 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_4_7STR	GMR=0.00700	DIAM=0.232	Rac=0.506
Normamps=138 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_2_7STR	GMR=0.00883	DIAM=0.292	Rac=0.318
Normamps=185 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_1_7STR	GMR=0.0099	DIAM=0.328	Rac=0.252
Normamps=214 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_1/0_7STR	GMR=0.0111	DIAM=0.368	Rac=0.2000
Normamps=247 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_2/0_7STR	GMR=0.0125	DIAM=0.414	Rac=0.1590
Normamps=286 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_3/0_7STR	GMR=0.0140	DIAM=0.464	Rac=0.1260
Normamps=331 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_4/0_7STR	GMR=0.0158	DIAM=0.522	Rac=0.0999
Normamps=383 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_250_7STR	GMR=0.0171	DIAM=0.567	Rac=0.0846
Normamps=425 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_250_19STR	GMR=0.0181	DIAM=0.574	Rac=0.0846
Normamps=426 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_266.8_7STR	GMR=0.0177	DIAM=0.586	Rac=0.0793
Normamps=443 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_266.8_19STR	GMR=0.0187	DIAM=0.592	Rac=0.0793
Normamps=444 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_300_19STR	GMR=0.0198	DIAM=0.628	Rac=0.0706
Normamps=478 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_336.4_19STR	GMR=0.0210	DIAM=0.665	Rac=0.0630
Normamps=513 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_350_19STR	GMR=0.0214	DIAM=0.679	Rac=0.0605
Normamps=526 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_397.5_19STR	GMR=0.0228	DIAM=0.723	Rac=0.0534
Normamps=570 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_450_19STR	GMR=0.0228	DIAM=0.769	Rac=0.0472
Normamps=616 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_477_19STR	GMR=0.0250	DIAM=0.792	Rac=0.0445
Normamps=639 Runits=kft	Radunits=in	GMRunits=ft	

new Wiredata.AAC_477_37STR	GMR=0.0254	DIAM=0.795	Rac=0.0445
Normamps=639 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_500_19STR	GMR=0.0256	DIAM=0.811	Rac=0.0425
Normamps=658 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_500_37STR	GMR=0.0260	DIAM=0.814	Rac=0.0425
Normamps=658 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_556.5_19STR	GMR=0.0270	DIAM=0.856	Rac=0.0382
Normamps=703 Runits=kft	Radunits=in	GMRunits=ft	
new Wiredata.AAC_556.5_37STR	GMR=0.0275	DIAM=0.858	Rac=0.0382
Normamps=704 Runits=kft	Radunits=in	GMRunits=ft	

B.3. Conductores ACSR

New WireData.ACSR_6_6STR	Rac=0.806	Runits=kft	GMR=0.0064
GMRunits=ft DIAM=0.198	Radunits=in	Normamps=105	
New WireData.ACSR_4_6STR	Rac=0.515	Runits=kft	GMR=0.0080
GMRunits=ft DIAM=0.250	Radunits=in	Normamps=140	
New WireData.ACSR_4_7STR	Rac=0.519	Runits=kft	GMR=0.0085
GMRunits=ft DIAM=0.257	Radunits=in	Normamps=140	
New WireData.ACSR_2_6STR	Rac=0.332	Runits=kft	GMR=0.0101
GMRunits=ft DIAM=0.316	Radunits=in	Normamps=184	
New WireData.ACSR_2_7STR	Rac=0.338	Runits=kft	GMR=0.0107
GMRunits=ft DIAM=0.325	Radunits=in	Normamps=184	
New WireData.ACSR_1_6STR	Rac=0.268	Runits=kft	GMR=0.0113
GMRunits=ft DIAM=0.354	Radunits=in	Normamps=212	
New WireData.ACSR_1/0_6STR	Rac=0.217	Runits=kft	GMR=0.0127
GMRunits=ft DIAM=0.398	Radunits=in	Normamps=242	
New WireData.ACSR_2/0_6STR	Rac=0.176	Runits=kft	GMR=0.0143
GMRunits=ft DIAM=0.447	Radunits=in	Normamps=276	
New WireData.ACSR_3/0_6STR	Rac=0.144	Runits=kft	GMR=0.0161
GMRunits=ft DIAM=0.502	Radunits=in	Normamps=315	
New WireData.ACSR_4/0_6STR	Rac=0.119	Runits=kft	GMR=0.0180
GMRunits=ft DIAM=0.563	Radunits=in	Normamps=357	
New WireData.ACSR_266.8_18STR	Rac=0.0787	Runits=kft	GMR=0.0197
GMRunits=ft DIAM=0.609	Radunits=in	Normamps=449	
New WireData.ACSR_266.8_22STR	Rac=0.0786	Runits=kft	GMR=0.0207
GMRunits=ft DIAM=0.624	Radunits=in	Normamps=450	
New WireData.ACSR_266.8_24STR	Rac=0.0782	Runits=kft	GMR=0.0212
GMRunits=ft DIAM=0.633	Radunits=in	Normamps=455	

New WireData.ACSR_266.8_26STR	Rac=0.0779	Runits=kft	GMR=0.0217
GMRunits=ft	DIAM=0.642	Radunits=in	Normamps=475
New WireData.ACSR_266.8_30STR	Rac=0.0773	Runits=kft	GMR=0.0227
GMRunits=ft	DIAM=0.660	Radunits=in	Normamps=460
New WireData.ACSR_300_26STR	Rac=0.0693	Runits=kft	GMR=0.0230
GMRunits=ft	DIAM=0.68	Radunits=in	Normamps=492
New WireData.ACSR_336.4_18STR	Rac=0.0625	Runits=kft	GMR=0.0221
GMRunits=ft	DIAM=0.684	Radunits=in	Normamps=519
New WireData.ACSR_336.4_20STR	Rac=0.0626	Runits=kft	GMR=0.0226
GMRunits=ft	DIAM=0.692	Radunits=in	Normamps=520
New WireData.ACSR_336.4_22STR	Rac=0.0624	Runits=kft	GMR=0.0232
GMRunits=ft	DIAM=0.701	Radunits=in	Normamps=520
New WireData.ACSR_336.4_24STR	Rac=0.0621	Runits=kft	GMR=0.0238
GMRunits=ft	DIAM=0.710	Radunits=in	Normamps=525
New WireData.ACSR_336.4_26STR	Rac=0.0618	Runits=kft	GMR=0.0244
GMRunits=ft	DIAM=0.720	Radunits=in	Normamps=529
New WireData.ACSR_336.4_30STR	Rac=0.0613	Runits=kft	GMR=0.0255
GMRunits=ft	DIAM=0.741	Radunits=in	Normamps=535
New WireData.ACSR_397.5_18STR	Rac=0.0529	Runits=kft	GMR=0.0240
GMRunits=ft	DIAM=0.743	Radunits=in	Normamps=576
New WireData.ACSR_397.5_20STR	Rac=0.0531	Runits=kft	GMR=0.0245
GMRunits=ft	DIAM=0.752	Radunits=in	Normamps=575
New WireData.ACSR_397.5_22STR	Rac=0.0528	Runits=kft	GMR=0.0252
GMRunits=ft	DIAM=0.762	Radunits=in	Normamps=580
New WireData.ACSR_397.5_24STR	Rac=0.0526	Runits=kft	GMR=0.0259
GMRunits=ft	DIAM=0.772	Radunits=in	Normamps=584
New WireData.ACSR_397.5_26STR	Rac=0.0523	Runits=kft	GMR=0.0265
GMRunits=ft	DIAM=0.783	Radunits=in	Normamps=587
New WireData.ACSR_397.5_30STR	Rac=0.0519	Runits=kft	GMR=0.0277
GMRunits=ft	DIAM=0.806	Radunits=in	Normamps=594
New WireData.ACSR_477_18STR	Rac=0.0442	Runits=kft	GMR=0.0263
GMRunits=ft	DIAM=0.814	Radunits=in	Normamps=646
New WireData.ACSR_477_24STR	Rac=0.0439	Runits=kft	GMR=0.0283
GMRunits=ft	DIAM=0.846	Radunits=in	Normamps=655
New WireData.ACSR_477_26STR	Rac=0.0436	Runits=kft	GMR=0.0290
GMRunits=ft	DIAM=0.858	Radunits=in	Normamps=659
New WireData.ACSR_477_30STR	Rac=0.0433	Runits=kft	GMR=0.0304

GMRunits=ft DIAM=0.883 Radunits=in Normamps=666

New WireData.ACSR_556.5_18STR Rac=0.0379 Runits=kft GMR=0.0275
GMRunits=ft DIAM=0.879 Radunits=in Normamps=711

New WireData.ACSR_556.5__24STR Rac=0.0376 Runits=kft GMR=0.0275
GMRunits=ft DIAM=0.914 Radunits=in Normamps=721

New WireData.ACSR_556.5__26STR Rac=0.0375 Runits=kft GMR=0.0275
GMRunits=ft DIAM=0.927 Radunits=in Normamps=726

New WireData.ACSR_556.5__30STR Rac=0.0372 Runits=kft GMR=0.0275
GMRunits=ft DIAM=0.953 Radunits=in Normamps=734

B.4. Conductores de cobre

New WireData.CU_14_SLD Rdc=2.626 Runits=kft GMR=0.00208
GMRunits=ft DIAM=0.0641 Radunits=in Normamps=20

New WireData.CU_8_SLD Rdc=0.653 Runits=kft GMR=0.00416
GMRunits=ft DIAM=0.1285 Radunits=in Normamps=95

New WireData.CU_6_SLD Rdc=0.411 Runits=kft GMR=0.00526
GMRunits=ft DIAM=0.1620 Radunits=in Normamps=125

New WireData.CU_4_SLD Rdc=0.258 Runits=kft GMR=0.00663
GMRunits=ft DIAM=0.2043 Radunits=in Normamps=170

New WireData.CU_2_SLD Rdc=0.163 Runits=kft GMR=0.00836
GMRunits=ft DIAM=0.2576 Radunits=in Normamps=225

New WireData.CU_2_7STR Rdc=0.1660 Runits=kft GMR=0.00883
GMRunits=ft DIAM=0.2920 Radunits=in Normamps=230

New WireData.CU_1/0_7STR Rdc=0.1042 Runits=kft GMR=0.01113
GMRunits=ft DIAM=0.3680 Radunits=in Normamps=310

New WireData.CU_1/0_19STR Rdc=0.1040 Runits=kft GMR=0.01178
GMRunits=ft DIAM=0.3730 Radunits=in Normamps=310

New WireData.CU_2/0_7STR Rdc=0.08267 Runits=kft GMR=0.01252
GMRunits=ft DIAM=0.4140 Radunits=in Normamps=355

New WireData.CU_3/0_7STR Rdc=0.06556 Runits=kft GMR=0.01404
GMRunits=ft DIAM=0.4640 Radunits=in Normamps=410

New WireData.CU_4/0_19STR Rdc=0.05199 Runits=kft GMR=0.01668
GMRunits=ft DIAM=0.5280 Radunits=in Normamps=480

New WireData.CU_250_19STR Rdc=0.04400 Runits=kft GMR=0.01813
GMRunits=ft DIAM=0.5740 Radunits=in Normamps=530

New WireData.CU_350_19STR Rdc=0.03143 Runits=kft GMR=0.02140
GMRunits=ft DIAM=0.6790 Radunits=in Normamps=650

New WireData.CU_500_37STR Rdc=0.02200 Runits=kft GMR=0.02605
GMRunits=ft DIAM=0.8140 Radunits=in Normamps=810