

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Diseño de topología de sistemas de distribución desde cero utilizando parámetros iniciales y ChatGPT con validación mediante OpenDSS

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero Eléctrico

Presentado por:

Juan Sebastián Jiménez Arosemena

Jordy Alexander Salavarría Tutivén

GUAYAQUIL - ECUADOR

Año: 2024

DEDICATORIA

Este proyecto se lo dedico a mis padres, cuyo apoyo, constante e incondicional me ha acompañado en cada paso de mi vida, especialmente en mi carrera universitaria.

A mi abuela, quien desde el primer día soñaba con verme como ingeniero, y que siempre me ha empujado a ser mejor persona.

A mi familia, quienes siempre han estado pendientes de mi progreso, apoyándome y aconsejándome en cada etapa del camino.

A mis amigos dentro y fuera de la universidad, gracias a quienes este camino estuvo lleno de nuevos aprendizajes, experiencias y alegrías.

Juan Sebastián Jiménez Arosemena

DEDICATORIA

Dedico este trabajo a mis padres, Pedro Salavarría y Verónica Tutivén, cuyo amor incondicional y apoyo constante han sido la luz que ha iluminado mi carrera universitaria. Su sacrificio y dedicación de aguantar mi carácter son la razón por el cual puedo celebrar este triunfo.

Agradecer a mis hermanas Mayken y Alisson, a través de su ejemplo y compañía, he encontrado la motivación para alcanzar una de muchas metas que tengo por delante.

A mis amigos que, mediante risas compartidas, conversaciones y momentos especiales han hecho que enfrente los desafíos que se me presentan con fuerza.

Gracias a todas esas personas especiales en mi vida, este logro es parte de ustedes.

Jordy Alexander Salavarría Tutivén

AGRADECIMIENTOS

Quiero empezar agradeciendo a Dios por guiar mi vida hasta este punto. Todo lo bueno y lo malo, alegrías y tristezas, me llevaron a ser quien soy hoy. Siempre confiando en el plan que tiene para cada uno de nosotros.

También quiero agradecer especialmente a mis tutores, el Ph.D Luis Ugarte e Ing. Jocelyne Salgado, al igual que a nuestro profesor el Ph.D Ángel Recalde. Este proyecto no hubiera sido posible sin su constante e incondicional apoyo en cada etapa del proceso.

Y, por último, a la ESPOL y todos los docentes que formaron parte de mi educación. Gracias por toda su labor y por los conocimientos impartidos. Que estos nos sirvan para ser ingenieros comprometidos por el bien de la sociedad.

Juan Sebastián Jiménez Arosemena

AGRADECIMIENTOS

Mi más sincero agradecimiento a Dios, quien con su guía y fuerza divina ha fortalecido mi espíritu a lo largo de mi vida. Este logro es testimonio de su amor y su infinita bondad.

Agradecer a mis queridos tutores el Ph.D Luis Ugarte e Ing. Jocelyne Salgado por brindarme no solo su orientación profesional, sino también su valiosa amistad y confianza a lo largo de este proyecto. Su apoyo ha sido el motor que me impulsa a nunca rendirme. Su contribución ha sido fundamental para este camino hacia el éxito. Y a nuestro profesor el Ph.D Ángel Recalde por las mejoras y correcciones mencionadas en cada clase.

Finalmente, deseo expresar mi más sincero agradecimiento a la ESPOL y a cada uno de los docentes que contribuyeron a mi formación académica. Su dedicación y pasión por la enseñanza han dejado una huella imborrable en mi trayectoria educativa.

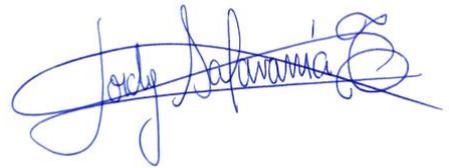
Jordy Alexander Salavarría Tutivén

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Juan Sebastián Jiménez Arosemena y Jordy Alexander Salavarría Tutivén y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Juan Sebastián
Jiménez Arosemena



Jordy Alexander
Salavarría Tutivén

EVALUADORES

Ph.D Recalde Lino Ángel Andrés

PROFESOR DE LA MATERIA

Ph.D Ugarte Vega Luis Fernando

PROFESOR TUTOR

RESUMEN

Este proyecto explora la modernización de los sistemas de distribución eléctrica mediante la integración de la inteligencia artificial (IA), específicamente a través de herramientas como ChatGPT y OpenDSS. El objetivo es abordar los desafíos de falta de datos precisos en los sistemas de distribución eléctrica, utilizando la IA para generar datos sintéticos que mejoren la precisión de los análisis. En el desarrollo del proyecto, se implementaron datos sintéticos en diferentes escenarios de operación. Estos datos se integraron con OpenDSS, facilitando análisis más detallados y precisos del flujo de carga en redes eléctricas. Este enfoque permitió superar las limitaciones de los datos reales. Los resultados obtenidos demuestran la eficacia del uso de la IA en campos tradicionales como la ingeniería. Se analizaron varios casos de estudio, desde sistemas sencillos hasta redes de gran tamaño y complejidad, utilizando conversaciones con ChatGPT 4 para modelar y optimizar estos sistemas. La validación mediante OpenDSS confirmó la aplicabilidad de los modelos generados por IA, destacando mejoras significativas en la precisión y eficiencia. Por último, este trabajo destaca el potencial de la IA, especialmente de ChatGPT, en la mejora de los sistemas de distribución eléctrica. La combinación de datos sintéticos generados por IA y su validación a través de OpenDSS ofrece un camino prometedor hacia sistemas eléctricos más eficientes y fiables, a pesar de los desafíos técnicos y regulatorios que implica su implementación.

Palabras Clave: ChatGPT, OpenDSS, Sistemas de Distribución, Inteligencia artificial.

ABSTRACT

This project explores the modernization of electrical distribution systems through the integration of artificial intelligence (AI), specifically through tools like ChatGPT and OpenDSS. The goal is to address the challenges of lacking accurate data in electrical distribution systems, using AI to generate synthetic data that improves the accuracy of analyses. In the development of the project, synthetic data were implemented in different operating scenarios. These data were integrated with OpenDSS, facilitating more detailed and accurate analyses of load flow in electrical networks. This approach allowed overcoming the limitations of real data. The results demonstrate the effectiveness of using AI in traditional fields such as engineering. Various case studies were analyzed, from simple systems to large and complex networks, using conversations with ChatGPT 4 to model and optimize these systems. The validation through OpenDSS confirmed the applicability of the AI-generated models, highlighting significant improvements in precision and efficiency. Lastly, this work highlights the potential of AI, especially ChatGPT, in improving electrical distribution systems. The combination of synthetic data generated by AI and its validation through OpenDSS offers a promising path towards more efficient and reliable electrical systems, despite the technical and regulatory challenges involved in their implementation.

Keywords: ChatGPT, OpenDSS, Distribution Systems, Artificial intelligence.

ÍNDICE GENERAL

EVALUADORES	7
RESUMEN	8
<i>ABSTRACT</i>	9
ÍNDICE GENERAL.....	10
ABREVIATURAS	13
ÍNDICE DE TABLAS	15
CAPÍTULO 1	16
1. Introducción	16
1.1 Descripción del problema	16
1.2 Justificación del problema.....	17
1.3 Objetivos.....	18
1.3.1 Objetivo General.....	18
1.3.2 Objetivos Específicos	18
1.4 Marco teórico.....	18
1.4.1 Sistemas eléctricos de distribución.....	18
1.4.2 Mediciones en sistemas de distribución	20
1.4.3 Uso de datos sintéticos	21
1.4.4 La inteligencia artificial en la ingeniería	23
1.4.5 Análisis de datos con inteligencia artificial	23
1.4.6 Descripción del modelo ChatGPT.....	25
1.4.7 Capacidades y Limitaciones de ChatGPT	25
1.4.8 Aplicaciones de ChatGPT	25
1.4.9 Descripción general de OpenDSS	26
1.4.10 Validación en OpenDSS.....	26
1.4.11 Principios de simulación en sistemas de distribución eléctrica	27

CAPÍTULO 2	28
2. OPENDSS EN SISTEMAS DE DISTRIBUCIÓN	28
2.1 Qué es OpenDSS.....	29
2.2 Funciones clave de OpenDSS.....	29
2.3 Componentes para la simulación de sistema de distribución eléctrica.	33
2.4 Interfaz de usuario de OpenDSS.....	33
2.5 Ejecución del código	35
2.6 Sintaxis en OpenDSS	36
CAPÍTULO 3	50
3. CARACTERIZANDO A CHATGPT	50
3.1 Qué es ChatGPT	51
3.2 Evolución de ChatGPT.....	51
3.3 Beneficios y desafíos de ChatGPT.....	53
3.4 Interfaz de ChatGPT	53
3.5 Comparación entre ChatGPT versión 3.5 y 4	55
3.6 Limitaciones y potencial de ChatGPT en generación de texto.....	58
3.7 ChatGPT en Sistemas de Potencia	62
3.8 Comparación Sistema Eléctrico de Potencia (SEP): Tradicional vs Innovación	63
3.9 Aplicaciones generales de ChatGPT en el campo eléctrico.....	65
CAPÍTULO 4	74
4. METODOLOGÍA	74
4.1 Análisis de las Capacidades de ChatGPT	74
4.2 Ingreso de Prompts.....	76
4.3 Generación del código por ChatGPT	77
4.4 Validación mediante OpenDSS	78

4.5	Retroalimentación del código generado.....	79
4.6	Actualización de los “prompts”.....	80
4.7	Validación posterior en OpenDSS y aprobación de código.....	80
CAPÍTULO 5		82
5. RESULTADOS Y ANÁLISIS.....		82
5.1	Caso de estudio 1: Circuitos pequeños.....	82
5.2	Caso de estudio 2: Circuitos grandes.....	96
5.3	Caso de estudio 3: Generación distribuida	102
5.4	Caso de estudio 4: Expansión de sistemas	108
5.5	Caso de estudio 5: Solución de problemas.....	116
5.6	Caso de estudio 6: Diseño de topología en base a cargas establecidas.....	123
CAPÍTULO 6		127
6. Conclusiones Y Recomendaciones.....		127
6.1	Conclusiones.....	127
6.2	Recomendaciones.....	128
REFERENCIAS		130

ABREVIATURAS

OpenDSS	Open Distribution System Simulator
IA	Inteligencia Artificial
EPRI	Electric Power Research Institute
PLN	Procesamiento del Lenguaje Natural
RED	Recursos Energéticos Distribuidos
RDA	Redes de Distribución Activa

ÍNDICE DE FIGURAS

Figura 1.1 Estructura de un sistema de distribución eléctrica.....	19
Figura 1.2 Esquema de flujo de energía en la red	21
Figura 1.3 Herramientas claves en el análisis de datos con inteligencia artificial.....	23
Figura 1.4 Aplicaciones de ChatGPT [23].....	26
Figura 2.1 Funciones de OpenDSS en redes de distribución	30
Figura 2.2 Interfaz principal de OpenDSS con el usuario	34
Figura 2.3 Pasos de ejecución del código	35
Figura 2.4 Sistema de ejemplo [29]	36
Figura 3.1 Línea de tiempo de las diferentes versiones de ChatGPT	52
Figura 3.2 Interfaz de ChatGPT 4	54
Figura 3.3 Características del funcionamiento de ChatGPT	60
Figura 3.4 Ejemplos específicos para indicar una tarea a ChatGPT	61
Figura 3.5 Comparación entre sistemas de distribución tradicional y modernos	63
Figura 3.6 Marco de programación interactivo apoyado en ChatGPT	67
Figura 3.7 Diagrama de flujo de estrategia para la resolución de problemas	68
Figura 3.8 Integrando ChatGPT en la educación y toma de decisiones en los sistemas de potencia.....	70
Figura 3.9 Marco de trabajo humano para la programación de sistemas de potencia con modelos de lenguaje grande [38].....	72
Figura 4.1 Análisis cualitativo de las capacidades de ChatGPT.....	75
Figura 4.2. Flujograma de metodología propuesta.....	76
Figura 5.1 Diagrama del sistema del Caso de estudio 1.....	82
Figura 5.2 Resultados de la simulación del Caso de estudio 1	95
Figura 5.3 Diagrama del sistema del Caso de estudio 2.....	96
Figura 5.4 Resultados de la simulación del Caso de estudio 2	102
Figura 5.5 Diagrama del sistema del Caso de estudio 3.....	103
Figura 5.6 Resultados de la simulación del Caso de estudio 3	107
Figura 5.7 Diagrama del sistema del Caso de estudio 4.....	108
Figura 5.8 Resultados de la simulación del Caso de estudio 4	115
Figura 5.9 Resultados de la simulación para el Caso de estudio 5.....	122
Figura 5.10 Resultados de la simulación para el Caso de estudio 6.....	126

ÍNDICE DE TABLAS

Tabla 2.1 Principales componentes de la interfaz de OpenDSS con el usuario	34
Tabla 2.2. Explicación de pasos de ejecución de código.....	36
Tabla 2.3. Descripción de parámetros del sistema	37
Tabla 2.4. Detalles del comando "Help"	38
Tabla 2.5. Valores de componentes de un conductor tipo 336 MCM ACSR [30][31]	41
Tabla 2.6.Descripción de parámetros de carga.....	44
Tabla 2.7. Descripción de parámetros transformador	45
Tabla 2.8. Descripción de parámetros de capacitor	46
Tabla 2.9. Descripción de parámetros de generador	47
Tabla 2.10. Descripción de parámetros de monitor.....	48
Tabla 3.1. Detalles de las diferentes versiones de ChatGPT	52
Tabla 3.2 Descripción de la interfaz de ChatGPT 4	55
Tabla 3.3. Cuadro comparativo entre las versiones 3.5 y 4 de ChatGPT	57
Tabla 3.4 Aplicaciones de ChatGPT en el campo de la electricidad	65

CAPÍTULO 1

1. INTRODUCCIÓN

En este trabajo exploraremos la modernización de los sistemas de distribución eléctrica, un campo que enfrenta desafíos debido a la falta de datos precisos para realizar análisis de operación y expansión. La inteligencia artificial (IA), especialmente a través de herramientas como ChatGPT, se presenta como una solución prometedora, generando datos sintéticos que pueden complementar y mejorar la precisión de los análisis existentes.

Una herramienta clave en este proceso es Open Distribution System Simulator (OpenDSS), un simulador de sistemas de distribución eléctrica que permite un análisis de flujo de carga en redes eléctricas. Sin embargo, su eficacia depende en gran medida de la precisión de los datos de entrada. Es aquí donde ChatGPT desempeña un papel crucial, ayudando a refinar y validar estos datos, lo que resulta en simulaciones más precisas y confiables.

Esta tesis destaca la interacción de la inteligencia artificial con la ingeniería eléctrica, abriendo un camino hacia sistemas de distribución eléctrica más eficientes y fiables. A través de la integración de tecnologías avanzadas como ChatGPT y OpenDSS, se propone un enfoque innovador para abordar los retos actuales en el ámbito de la distribución eléctrica.

1.1 Descripción del problema

La gestión de redes de distribución eléctrica se enfrenta a un desafío mayúsculo debido a la carencia de datos detallados y precisos, un problema de gran envergadura debido a la complejidad y diversidad inherente a estas redes, las cuales incluyen una amplia gama de consumidores, desde residenciales hasta industriales y comerciales. La gran cantidad de nodos y barras en estas redes incrementa la dificultad en la recopilación y análisis de datos fiables, aspectos cruciales para una operación y planificación eficientes. Esta falta de información impacta negativamente en la gestión de variables críticas como los

niveles de tensión y la cargabilidad de transformadores, fundamentales para mantener una operación eficiente y prevenir sobrecargas. Sin datos adecuados, es complicado garantizar que los niveles de tensión se mantengan dentro de límites seguros y que los transformadores operen dentro de sus capacidades óptimas, lo que es esencial para evitar interrupciones y asegurar un suministro eléctrico constante y confiable.

Además, la operación de la red eléctrica está sujeta a restricciones significativas, incluyendo capacidades máximas de equipos, normativas de seguridad y requerimientos ambientales. La escasez de datos fiables hace que sea difícil navegar estas restricciones y tomar decisiones informadas, especialmente en situaciones de emergencia o picos de demanda, donde se requieren acciones rápidas, basadas en información precisa, para mantener la estabilidad del sistema. Esta falta de datos también limita la capacidad de las empresas para prever y mitigar riesgos, lo que es crucial en el diseño de sistemas eléctricos robustos y redundantes, diseñados para resistir fallos y garantizar un suministro ininterrumpido de energía. La planificación y diseño de tales sistemas dependen en gran medida de un análisis exhaustivo basado en datos completos y fiables, y la falta de estos datos obstaculiza el desarrollo de soluciones para fortalecer la resiliencia de la red.

1.2 Justificación del problema

La generación de datos sintéticos utilizando avanzadas técnicas de inteligencia artificial, como redes neuronales y algoritmos genéticos, se presenta como una solución innovadora y efectiva frente a la escasez de datos en las redes de distribución eléctrica. Estas tecnologías de IA tienen la capacidad de simular escenarios realistas y generar datos que reflejen diversas condiciones de operación, ayudando a superar las limitaciones de escasez de datos. Al validar estos datos sintéticos con precisión, se asegura su relevancia y aplicabilidad, permitiendo su integración efectiva con los sistemas de distribución existentes. Esta integración mejora significativamente la toma de decisiones en aspectos críticos como la gestión de la demanda y la fiabilidad del suministro eléctrico, permitiendo a los operadores anticipar y responder de manera más eficiente a las variaciones y necesidades del sistema.

Para las empresas de distribución eléctrica, la implementación de estas soluciones basadas en IA ofrece beneficios tangibles. Por ejemplo, mejora la precisión en la predicción de la demanda, optimiza la asignación de recursos y facilita la planificación a largo plazo de las infraestructuras de la red. Sin embargo, la adopción de estas tecnologías también conlleva desafíos significativos. Desde un punto de vista técnico, la integración de datos sintéticos en sistemas existentes requiere un conocimiento profundo de ambos, los algoritmos de IA y la operación de la red eléctrica. Además, existen desafíos relacionados con la seguridad y privacidad de los datos, así como la necesidad de establecer estándares y protocolos para la validación de dichos datos sintéticos.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar una topología eficiente para sistemas de distribución eléctrica desde cero, utilizando la inteligencia artificial de ChatGPT y parámetros iniciales, para una posterior verificación de su rendimiento mediante simulaciones en el software OpenDSS.

1.3.2 Objetivos Específicos

1. Realizar simulaciones en OpenDSS utilizando las topologías propuestas por ChatGPT, evaluando métricas específicas como niveles de tensión y flujo de potencia.
2. Analizar los resultados de las simulaciones, identificando áreas de mejora en las topologías propuestas y ajustando el modelo de ChatGPT según sea necesario, con el objetivo de mejorar la eficiencia y confiabilidad del sistema.
3. Evaluar la capacidad de herramientas de inteligencia artificial como ChatGPT en la creación de datos sintéticos de redes de distribución desde 0 y a partir de datos iniciales.

1.4 Marco teórico

1.4.1 Sistemas eléctricos de distribución

Los sistemas de distribución eléctrica son componentes críticos de la infraestructura energética moderna, desempeñando un papel vital en la entrega de electricidad

proveniente de las fuentes generadoras, la cual pasa por redes de transmisión y debe ser entregada a los consumidores finales. Estos sistemas son esenciales para garantizar que hogares, industrias, y negocios reciban un suministro de energía confiable y de calidad [1]. La red de distribución comienza en las subestaciones de distribución, donde la electricidad se recibe a un voltaje más alto de las líneas de transmisión y luego se reduce a un voltaje más bajo, adecuado para el uso doméstico y comercial.

La configuración de un sistema de distribución típico incluye varias componentes clave: las subestaciones de distribución, las líneas de distribución, los transformadores, y los equipos de medición y protección. Las subestaciones actúan como puntos nodales, donde la electricidad se reduce a voltajes más bajos y se distribuye a través de redes de líneas, que pueden ser aéreas o subterráneas [2]. Los transformadores, ubicados en la subestación o distribuidos a lo largo de la red, ajustan aún más los niveles de voltaje para satisfacer las necesidades específicas de diferentes tipos de clientes. Además, los sistemas de distribución modernos están equipados con una variedad de dispositivos de medición y protección, como contadores inteligentes y relés de protección, que no solo ayudan a monitorear y controlar la red, sino que también mejoran la seguridad y la eficiencia del sistema [3]. Un esquema de los sistemas de distribución se puede apreciar en la Figura 1.1.

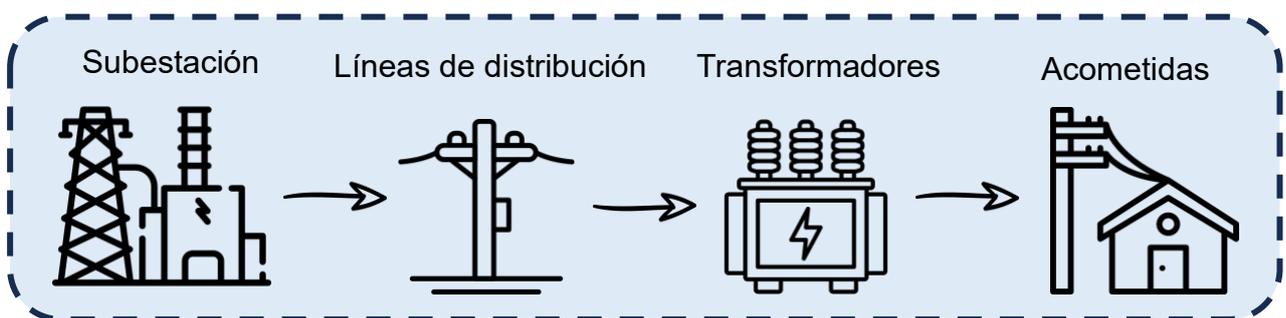


Figura 1.1 Estructura de un sistema de distribución eléctrica

Los componentes de un sistema de distribución eléctrico, como transformadores, líneas de distribución, interruptores y dispositivos de protección, son fundamentales para el funcionamiento eficiente y seguro de la red eléctrica. La protección y el mantenimiento adecuados de estos componentes son cruciales para garantizar una distribución de

energía confiable y para prolongar su vida útil [4]. Sin embargo, la obtención de datos precisos y detallados sobre el estado y el rendimiento de estos componentes puede ser un desafío, especialmente en sistemas más antiguos o menos avanzados tecnológicamente. Esta limitación en la adquisición de datos puede llevar a problemas en la identificación temprana de fallos o deterioros, lo que a su vez puede acortar la vida útil de los equipos y aumentar los costos de mantenimiento y reemplazo [5]. La falta de información detallada y en tiempo real sobre la operación y el estado de los componentes del sistema también dificulta la optimización de su funcionamiento y la prevención de interrupciones, lo que es esencial para un sistema de distribución eléctrica robusto y confiable [6].

1.4.2 Mediciones en sistemas de distribución

La medición en los sistemas de distribución eléctrica es un aspecto clave para la monitorización y el control de la red. En muchos sistemas de distribución tradicionales, la medición se concentra en las subestaciones. Estas mediciones proporcionan información sobre el flujo de electricidad en puntos de alto nivel de la red, pero no ofrecen una visión detallada del comportamiento de la red en los puntos más cercanos a los usuarios finales, como viviendas, comercios e industrias [7].

La centralización de las mediciones en las subestaciones implica que se obtiene una perspectiva general del flujo de energía en la red, sin un desglose detallado de cómo se distribuye y se consume la energía a nivel local, tal como muestra la Figura 1.2. Esto puede resultar en una visibilidad reducida de las condiciones de carga específicas en los transformadores de baja tensión y en los puntos de entrega a los consumidores [8].

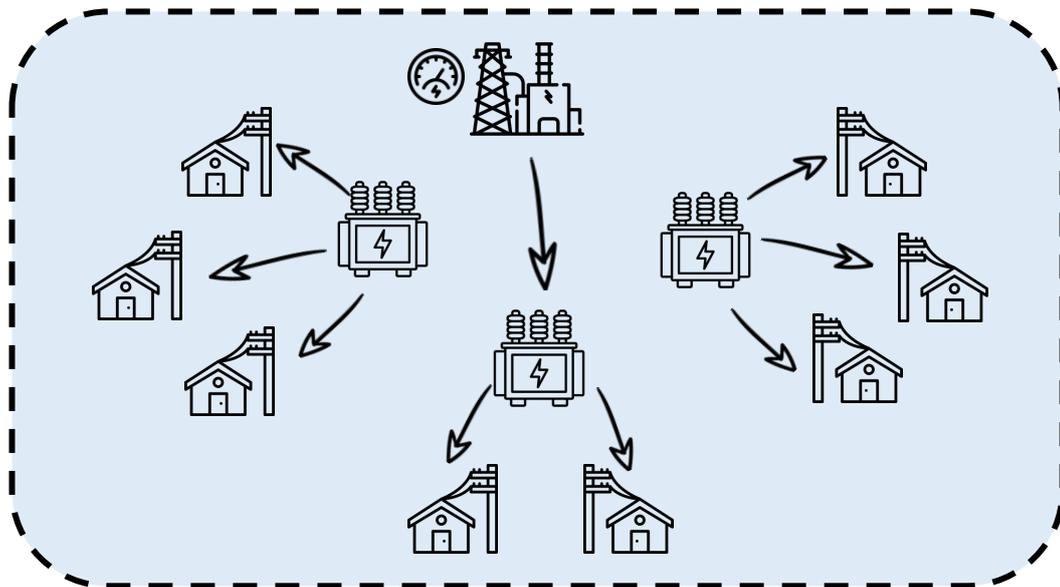


Figura 1.2 Esquema de flujo de energía en la red

Esta limitación en la medición puede influir en la capacidad para comprender y reaccionar adecuadamente a las variaciones locales en la demanda y la oferta de energía. Sin una medición detallada en los niveles más bajos de la red, se reduce la capacidad para identificar con precisión problemas como las fluctuaciones de demanda, las interrupciones del suministro o los problemas de calidad de la energía [9].

Además, la falta de mediciones detalladas en puntos cercanos a los transformadores de baja tensión y a los consumidores finales plantea desafíos teóricos en la integración de fuentes de energía distribuida y en la implementación de estrategias de gestión de demanda. La ausencia de datos aguas abajo puede limitar la capacidad de optimizar la operación de la red para acomodar estas fuentes de generación distribuida y mantener la estabilidad de la red [10].

1.4.3 Uso de datos sintéticos

Los datos sintéticos en sistemas de distribución eléctricos se refieren a información generada artificialmente que imita las propiedades y comportamientos de datos reales del sistema. Estos datos son creados a través de modelos y simulaciones que replican las operaciones de una red de distribución eléctrica, incluyendo la generación de energía, la transmisión, la distribución, y el consumo [11].

El uso de datos sintéticos es particularmente valioso en situaciones donde los datos reales son insuficientes, inaccesibles o demasiado costosos de obtener. Esto es especialmente relevante en el campo de la distribución eléctrica, donde la recopilación de datos a gran escala puede ser compleja y muy costosa [12]. Los datos sintéticos permiten realizar análisis y pruebas sin la necesidad de depender exclusivamente de datos reales, lo que puede ser útil en diversas áreas como:

1. Pruebas y Validación de Modelos: Los datos sintéticos son útiles para validar modelos matemáticos y algoritmos utilizados en la gestión y operación de la red. Al comparar los resultados de los modelos con los datos sintéticos, se puede evaluar la precisión y fiabilidad de estos modelos.
2. Análisis de Escenarios: Mediante la generación de una variedad de escenarios posibles en un entorno controlado, los datos sintéticos permiten a los operadores de red y planificadores evaluar el impacto de diferentes situaciones, como cambios en la demanda, la integración de energías renovables, o fallos en componentes específicos del sistema.
3. Formación y Desarrollo de Algoritmos de Aprendizaje Automático: En la era del big data y la inteligencia artificial, los datos sintéticos proporcionan un recurso valioso para entrenar algoritmos de aprendizaje automático. Estos algoritmos, una vez entrenados con datos sintéticos, pueden ser aplicados para la detección de anomalías, la predicción de la demanda, y la optimización de la operación de la red [13].
4. Planificación de Expansión y Mejoras: Los datos sintéticos ayudan en la planificación estratégica de la expansión de la red y en la implementación de mejoras. Al simular diferentes condiciones y demandas futuras, los planificadores pueden identificar áreas que necesitan refuerzos o actualizaciones.

1.4.4 La inteligencia artificial en la ingeniería

La introducción de la inteligencia artificial (IA) en la ingeniería ha sido un avance crucial en la evolución tecnológica. Desde sus inicios en los años 50, la IA ha progresado significativamente, ofreciendo soluciones innovadoras para desafíos complejos en ingeniería que antes requerían la intervención humana. Este avance se ha reflejado en el desarrollo de herramientas poderosas con aplicaciones prácticas en diversas áreas de la ingeniería. Destacan el uso de sistemas basados en el conocimiento, que emplean información experta para tomar decisiones informadas imitando el razonamiento humano, y la lógica difusa, que maneja la incertidumbre en datos adaptándose a situaciones del mundo real. Además, técnicas como el aprendizaje inductivo, las redes neuronales y los algoritmos genéticos han demostrado su valía en el procesamiento de información compleja y la resolución de problemas no lineales [14].

1.4.5 Análisis de datos con inteligencia artificial

El texto destaca que la aplicación de la Inteligencia Artificial (IA) se ha vuelto fundamental en el análisis de datos en diversos sectores y áreas académicas. Esta tecnología tiene la capacidad de procesar y examinar extensos conjuntos de datos de forma más ágil y efectiva que los enfoques convencionales, generando claridades valiosas y simplificando la toma de decisiones fundamentada en datos.

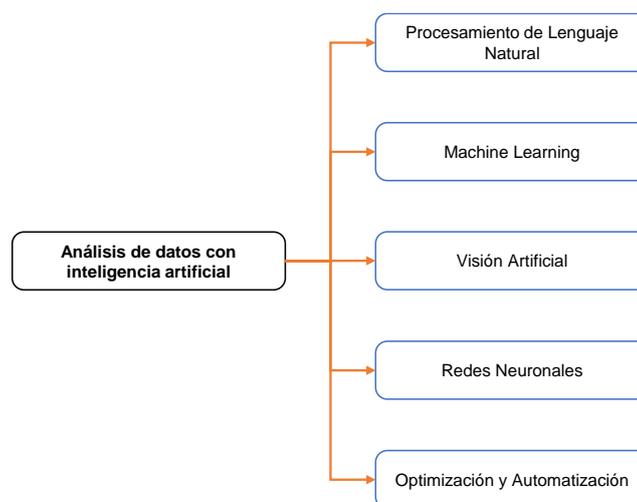


Figura 1.3 Herramientas claves en el análisis de datos con inteligencia artificial

En la Figura 1.4 se muestran los análisis de datos con la IA por lo que a continuación se detalla cada método.

Procesamiento de Lenguaje Natural

Para analizar datos, es clave poder entender y procesar el lenguaje humano. En este contexto, el procesamiento del lenguaje natural juega un papel importante al hacer que las máquinas se comuniquen de manera más fácil con los humanos, interpretando y generando texto de forma contextual y significativa [15].

Machine Learning

Este método se basa en el aprendizaje automático, que es básicamente una forma en que las máquinas pueden volverse mejores por sí mismas a medida que se enfrentan a más información. Esta mejora constante les ayuda a adaptarse a cambios en los patrones y a realizar predicciones con mayor exactitud [16] [20].

Visión Artificial

La visión artificial es una parte destacada de la inteligencia artificial que tiene como objetivo dotar a las máquinas de la capacidad para entender y procesar información visual de su entorno. Ha experimentado progresos significativos en diversas áreas de aplicación, que van desde el reconocimiento facial y la inspección industrial hasta la automatización de procesos en la fabricación [17].

Redes Neuronales

En el ámbito de la inteligencia artificial de última generación, las redes neuronales imitan la forma en que funciona el cerebro humano para llevar a cabo tareas complicadas de procesamiento de información. Estas redes, inspiradas en la biología, son esenciales para reconocer patrones y tomar decisiones basadas en datos [18].

Optimización y Automatización

La mejora y automatización son muy importantes para hacer que los procesos analíticos funcionen mejor. La mejora se enfoca en hacer que los algoritmos y modelos sean lo

mejor posible, mientras que la automatización hace que las tareas repetitivas se realicen más rápido y eficientemente, agilizando las operaciones analíticas [19].

1.4.6 Descripción del modelo ChatGPT

ChatGPT marca un avance destacado en la frontera de los modelos de procesamiento del lenguaje natural respaldados por inteligencia artificial (IA). Se posiciona como una parte fundamental en la revolución de la gestión de sistemas de potencia. Su capacidad innata para comprender y producir respuestas coherentes emerge como una herramienta estratégica en la creación de interfaces de usuario avanzadas diseñadas para supervisar y controlar sistemas eléctricos de notable complejidad [21].

1.4.7 Capacidades y Limitaciones de ChatGPT

Las habilidades de ChatGPT son notables en la mejora significativa de la eficiencia operativa y en facilitar la toma de decisiones informadas en el ámbito de los sistemas de potencia. Esta tecnología no solo favorece una comunicación más fluida entre usuarios y sistemas, sino que también demuestra habilidades para ofrecer interfaces de usuario conversacionales, permitiendo a los operadores y gestores obtener información detallada sobre el estado actual del sistema en tiempo real [22].

Sin embargo, es crucial señalar que, al igual que cualquier sistema, ChatGPT no está exento de limitaciones. Su rendimiento está intrínsecamente relacionado con la calidad y claridad de los datos de entrada, y existe la posibilidad de generar respuestas imprecisas o inadecuadas en situaciones donde la información proporcionada sea ambigua o incompleta [21]. La implementación exitosa de ChatGPT también requiere una adaptación cuidadosa a las especificidades particulares del sistema eléctrico en cuestión, lo que puede presentar desafíos adicionales.

1.4.8 Aplicaciones de ChatGPT

El texto hace énfasis a la Figura 1.5 con sus avanzadas capacidades de procesamiento del lenguaje natural (PLN), ChatGPT se ha convertido en una herramienta versátil con una amplia gama de aplicaciones más allá del ámbito de la investigación académica. Se

está utilizando en diferentes sectores para innovar procesos, mejorar la productividad y facilitar la resolución de problemas [23].

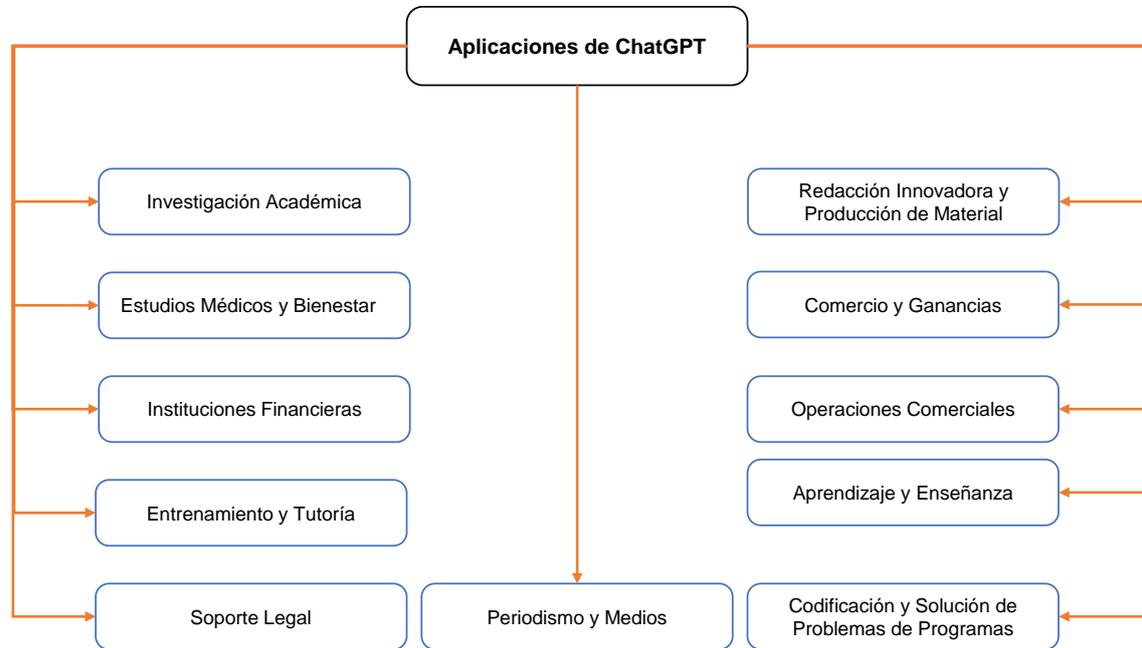


Figura 1.4 Aplicaciones de ChatGPT [23]

1.4.9 Descripción general de OpenDSS

OpenDSS se presenta como un software especializado en el análisis de sistemas de distribución eléctrica, sobresaliendo por su capacidad para llevar a cabo simulaciones detalladas. Este programa ofrece un entorno sólido que permite la modelización minuciosa de componentes específicos y condiciones particulares de la red, lo cual es esencial para una evaluación exhaustiva de estrategias de reconfiguración. Su función crucial radica en mejorar la precisión y validez de propuestas innovadoras en el ámbito de la gestión de redes de distribución eléctrica.

1.4.10 Validación en OpenDSS

Validar en OpenDSS es un paso crucial para asegurar que las simulaciones proporcionen resultados precisos y confiables. Este proceso implica comparar los resultados simulados con datos reales, asegurando que el modelo se ajuste adecuadamente a la realidad. Se realiza revisando y comparando cuidadosamente parámetros clave como voltajes, corrientes y pérdidas de potencia.

La confiabilidad de los resultados validados respalda la utilidad y eficacia de OpenDSS como una herramienta integral en la planificación y gestión de redes eléctricas.

1.4.11 Principios de simulación en sistemas de distribución eléctrica

La simulación es una herramienta esencial para modelar y comprender el comportamiento de estos sistemas. Los principios fundamentales incluyen la representación detallada de componentes, la consideración de condiciones específicas de la red y la capacidad para evaluar diversas estrategias de operación y reconfiguración. La simulación permite anticipar el rendimiento del sistema bajo diferentes escenarios, facilitando la toma de decisiones informadas en la gestión y optimización de redes de distribución eléctrica. Como se describe [24] la propuesta de combinar algoritmos genéticos implementados en MATLAB con simulaciones detalladas en OpenDSS representa una aproximación innovadora en la gestión de redes de distribución eléctrica. También se presenta [25] para la mejora y refuerzo de una microrred haciendo uso de Python junto a OpenDSS, comparando el rendimiento con otros métodos para garantizar una energía segura y confiable. Esta habilidad es fundamental para el desarrollo de estrategias óptimas de reconfiguración de la red de distribución eléctrica. La dinámica adaptativa de este enfoque se revela como clave para afrontar los desafíos asociados con la creciente presencia de fuentes de generación distribuida y las fluctuaciones dinámicas en la demanda de energía [26].

CAPÍTULO 2

2. OPENDSS EN SISTEMAS DE DISTRIBUCIÓN

En un mundo donde las energías renovables están en ascenso y las redes se vuelven más inteligentes, los sistemas de distribución están experimentando una revolución. Frente a estos cambios, la simulación y el análisis de sistemas de potencia se han convertido en herramientas cruciales para aquellos profesionales que buscan perfeccionar estas infraestructuras.

El software Open Distribution System Simulator (OpenDSS) se presenta como un aliado formidable y flexible en este escenario. Con su habilidad para simular las operaciones de las redes eléctricas y anticipar los resultados de distintas situaciones operacionales, se ha convertido en una herramienta indispensable.

Este capítulo se dedica a ofrecer una descripción amplia de OpenDSS, poniendo en evidencia las características que lo hacen único, y las principales funcionalidades que se han explotado para enriquecer los sistemas de distribución. Se examinará la estructura interna del programa, los componentes fundamentales y las interfaces de usuario que posibilitan una diversidad de análisis y simulaciones. Adicionalmente, se revisarán ejemplos específicos en los que OpenDSS ha jugado un papel crucial tanto en la solución de problemas de ingeniería como en la toma de decisiones cruciales para la gestión y planificación de redes.

Al concluir este capítulo, los lectores contarán con una visión clara de cómo OpenDSS se integra en los sistemas de distribución eléctrica y tendrán las bases necesarias para aplicar esta herramienta tanto en el ámbito investigativo como en el profesional. Al equilibrar teoría y práctica, aspiramos a que este capítulo no solo sirva de guía educativa, sino que también motive a la innovación en el ámbito de la distribución eléctrica.

2.1 Qué es OpenDSS

OpenDSS es una herramienta de simulación diseñado por el Instituto de Investigación de Energía Eléctrica (Electric Power Research Institute, EPRI) para sistemas de distribución de energía eléctrica [27]. Considerar OpenDSS como herramienta de simulación ayuda a solucionar las crecientes necesidades asociadas con los esfuerzos de modernización de las redes eléctricas actuales.

Esta herramienta permite evaluar tecnologías emergentes, la gestión de la demanda, integración de energías renovables y otros factores cruciales para maximizar el rendimiento y la eficiencia de las redes de distribución. Las capacidades de OpenDSS incluyen modelado detallado de componentes de red como líneas de transmisión, transformadores, dispositivos de equilibrio de carga y generadores distribuidos. El modelado detallado de elementos de red, incluidas líneas de distribución, transformadores, cargas y generadores distribuidos, es uno de los puntos fuertes de OpenDSS. Además, facilita el análisis del flujo de carga, el análisis de cortocircuitos, el análisis de estabilidad y otras simulaciones, todas ellas necesarias para comprender cómo se comporta un sistema eléctrico [28].

OpenDSS es un software gratuito que brinda a los usuarios la libertad de cambiar y personalizar el código fuente para adaptarlo a sus propios requisitos. Esto fomenta la innovación y el trabajo en equipo en la comunidad de ingenieros eléctricos. Además, la utilización de esta herramienta fomenta el progreso de la investigación y el desarrollo de redes de distribución modernas, permitiendo a científicos e ingenieros investigar soluciones efectivas y duraderas ante los problemas energéticos actuales.

2.2 Funciones clave de OpenDSS

OpenDSS es una herramienta de simulación que se utiliza para evaluar y mejorar la operación y diseño de sistemas de distribución eléctrica. Con capacidades para evaluar la estabilidad y seguridad de la red, el software permite identificar las áreas vulnerables, especialmente en eventos anormales de operación, como cortocircuitos. También se utiliza para medir y reducir las pérdidas de energía, lo cual es fundamental para mejorar

la eficiencia general del sistema eléctrico. El software facilita la incorporación de energías renovables al permitir la simulación de nuevas infraestructuras, como instalaciones solares fotovoltaicas, y su integración en la red existente. Además, OpenDSS maneja el modelado de patrones de carga variables, proporcionando una comprensión clara de cómo la demanda fluctuante afecta la red. Por otro lado, OpenDSS es vital para la planificación a largo plazo y el desarrollo de sistemas de distribución, lo que incluye la creación de modelos detallados para cada componente de la red y la preparación para futuras expansiones. Este software se posiciona como una herramienta integral que permite la realización de simulaciones exhaustivas y análisis complejos, llevando a una operación y planificación eficaz de las redes eléctricas.

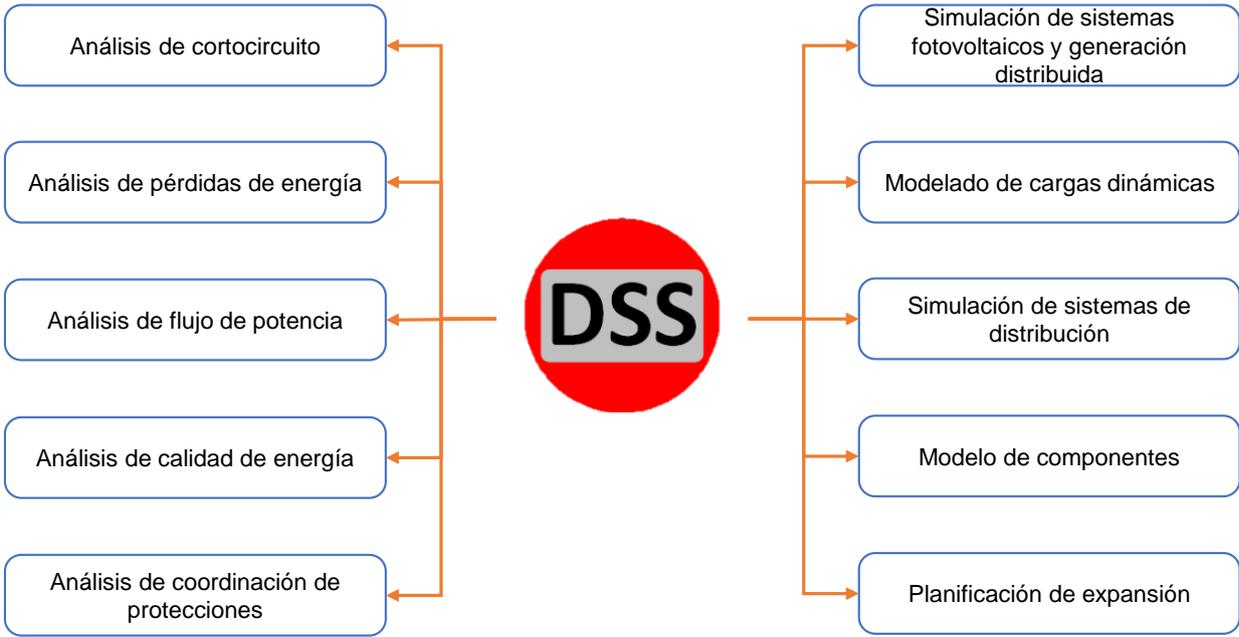


Figura 2.1 Funciones de OpenDSS en redes de distribución

En la Figura 2.1 se muestra las diferentes aplicaciones que realiza OpenDSS con sus respectivas descripciones.

Análisis de cortocircuitos

El estudio que facilita OpenDSS para el análisis de cortocircuito, permite que el operador simule y estime la capacidad de respuesta que se da en el sistema ante la presencia de una falla eléctrica. La identificación de corrientes de cortocircuitos, la coordinación eficiente de los elementos de protección y la mejora de seguridad del sistema son ejemplo de esto.

Análisis de pérdidas de energía

Este método implica determinar y medir la cantidad de electricidad que no llega a los consumidores debido a las pérdidas en su trayecto desde las plantas de producción hasta los puntos de entrega. Este paso es esencial para elevar la eficiencia del sistema de distribución de electricidad. Mediante este análisis, se pueden implementar tácticas para disminuir tales pérdidas, lo cual se logra al reestructurar la red eléctrica para asegurar un funcionamiento más efectivo.

Análisis de flujo de potencia

Los usuarios pueden analizar la distribución de potencia activa y reactiva en el sistema bajo diversas condiciones de carga. Esta información proporciona una gran contribución sobre cómo opera el sistema y ayuda a la toma de decisiones para mejorar la operación del sistema eléctrico.

Análisis de calidad de energía

Tendremos un análisis detallado en base a la simulación obtenida, esto incluye la evaluación de los armónicos presentes y otros parámetros relacionados. Los usuarios tienen la capacidad de identificar y abordar los problemas de los análisis de calidad de energía garantizando un suministro eléctrico confiable y de alta calidad.

Análisis de coordinación de protecciones

El análisis de coordinación de protecciones es útil en OpenDSS, permite a los ingenieros evaluar la secuencia y eficiencia de dispositivos de protección como interruptores, fusible, entre otros. Esto garantiza una respuesta correcta y la integridad del sistema en condiciones de falla.

Simulación de sistemas fotovoltaicos y generación distribuida

La modelación y simulación detalladas de sistema de generación distribuida, como instalaciones fotovoltaicas, son posibles en OpenDSS. Los usuarios pueden evaluar la integración en que se integran estas fuentes de energías renovables en la red y encontrar la manera de maximizar su eficiencia.

Modela de cargas dinámicas

La capacidad de modelar cargas dinámicas en OpenDSS, los usuarios pueden investigar como las variaciones de la demanda afectan dinámicamente el sistema eléctrico, es decir, se refiere al impacto en tiempo real o en periodos específicos de tiempo de los cambios en la cantidad de electricidad que los consumidores están utilizando en un sistema eléctrico.

Simulación de sistemas de distribución

Los usuarios pueden recrear escenarios específicos, evaluar el rendimiento del sistema y realizar estudios de planificación con las herramientas que ofrece OpenDSS para simular sistemas de distribución eléctrica en su totalidad.

Modelado de componentes

OpenDSS tiene la capacidad de modelar componentes del sistema detalladas, como interruptores, líneas distribución, transformadores y otros dispositivos. Esto garantiza que la representación del sistema en la simulación sea precisa considerando elementos tradicionales y especiales de la red eléctrica como por ejemplo los Recursos Energéticos Distribuidos (RED).

Planificación de expansión

La planificación de expansión con OpenDSS implica un proceso estratégico donde se evalúa la capacidad del sistema eléctrico para adaptarse al crecimiento previsto de la demanda en el futuro. Mediante el uso de OpenDSS, los ingenieros pueden realizar simulaciones detalladas de diversos escenarios de expansión, lo que les permite

identificar de manera precisa posibles limitaciones o en la infraestructura eléctrica existente.

2.3 Componentes para la simulación de sistema de distribución eléctrica.

Un sistema eléctrico está constituido por elementos tradicionales y especiales que hoy en día convierten nuestros sistemas convencionales de distribución en Redes de Distribución Activa (RDA) [29]. Dichos elementos son:

- Barras
- Líneas de distribución
- Cargas
- Transformadores
- Generadores distribuidos
- Interruptores y dispositivos de protección
- Monitoreos de carga
- Capacitores y reactores
- Reguladores de voltaje
- Controladores y escenarios de control
- Perfiles de carga (loadshapes)
- Vehículos eléctricos

Cada elemento antes mencionado, será caracterizado mediante un script en OpenDSS en las secciones posteriores.

2.4 Interfaz de usuario de OpenDSS

OpenDSS ofrece una interfaz gráfica de usuario (Figura 2.2) fácil de entender, que facilita la creación y el análisis de modelos de sistemas eléctricos detallados. Se pueden crear modelos jerárquicos precisos definiendo barras, líneas, transformadores, cargas y generadores distribuidos al iniciar una sesión. La interfaz permite configurar parámetros como perfiles de carga y características específicas de dispositivos de manera fácil de acceder.

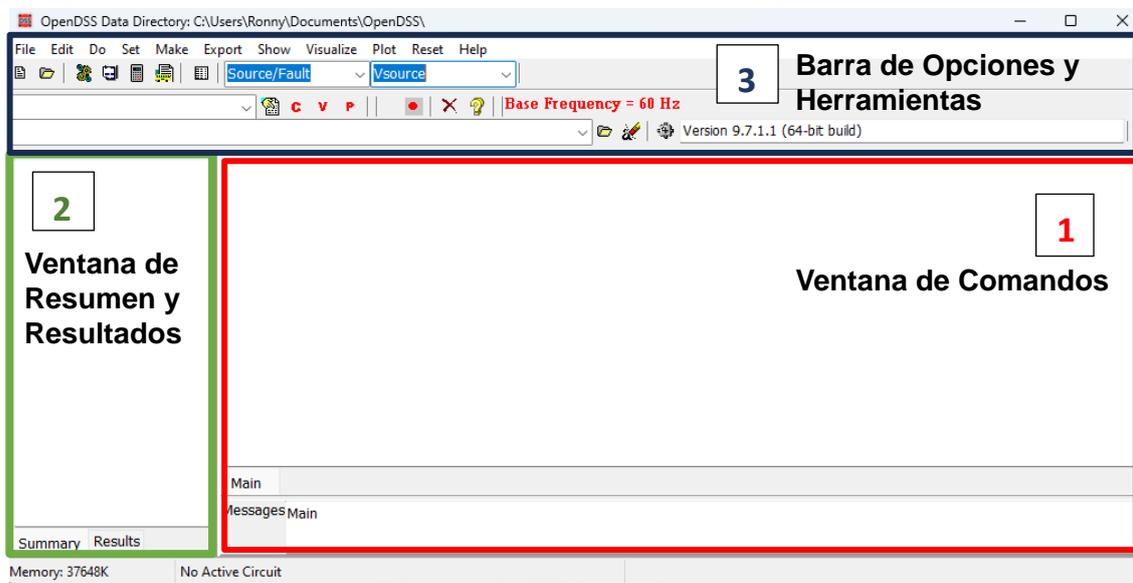


Figura 2.2 Interfaz principal de OpenDSS con el usuario

La Tabla 2.1 resume los principales componentes de la interfaz de OpenDSS, los cuales se organizan en tres categorías clave para facilitar el desarrollo y la ejecución de simulaciones en sistemas de distribución de energía eléctrica:

1	<ul style="list-style-type: none"> • Parte principal del programa donde se desarrollan las simulaciones. • Se escribe y ejecuta el código para modelar circuitos eléctricos. • Permite la flexibilidad para editar y manipular circuitos en cualquier momento. • Utiliza el lenguaje Object Pascal, enfocado en programación orientada a objetos.
2	<ul style="list-style-type: none"> • Esencial para la evaluación de la simulación eléctrica. • Muestra datos como la cantidad de buses y nodos. • Indica el método de solución y el tiempo de simulación. • Proporciona métricas importantes como potencias y rangos de voltajes.
3	<ul style="list-style-type: none"> • Ofrece herramientas para la extracción y configuración de datos • Permite ajustar parámetros específicos para la simulación. • Esencial para personalizar la simulación según las necesidades del usuario. • Contribuye a la potencia y flexibilidad de OpenDSS en la simulación de sistemas de distribución de energía.

Tabla 2.1 Principales componentes de la interfaz de OpenDSS con el usuario

2.5 Ejecución del código

La ejecución del código en OpenDSS implica el proceso de llevar a cabo simulaciones detalladas de sistemas de distribución para evaluar su desempeño y comportamiento ante diferentes escenarios. Para llevar a cabo una simulación en OpenDSS, inicialmente se debe preparar un modelo que describa la red eléctrica, definiendo los componentes como transformadores, líneas, cargas, y fuentes de generación, y especificando sus parámetros correspondientes.

La Figura 2.3 ofrece una representación visual de la secuencia de ejecución de código en OpenDSS, que es una herramienta esencial para la simulación y análisis de sistemas de distribución eléctrica. Esta figura visualiza un proceso de cuatro etapas que los usuarios deben seguir para llevar a cabo simulaciones efectivas:

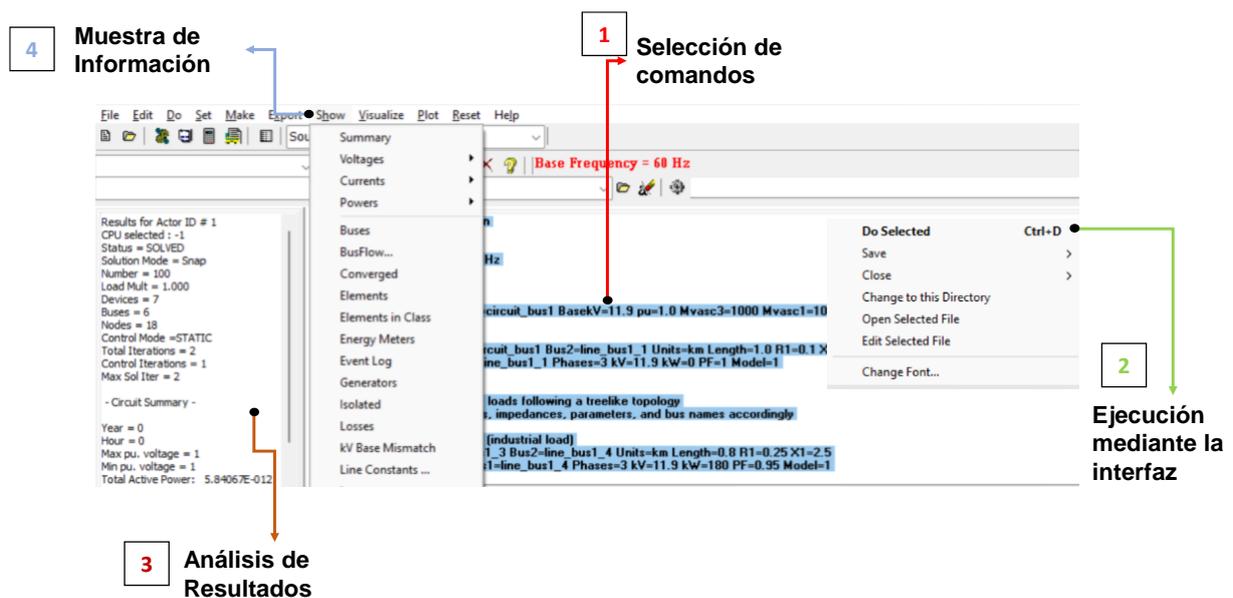


Figura 2.3 Pasos de ejecución del código

La Tabla 2.2 describe de forma detallada los pasos para ejecutar código en la plataforma OpenDSS, facilitando a los usuarios el proceso de simulación de sistemas de distribución eléctrica:

1	Seleccionar las líneas de código o los comandos que se desean ejecutar dentro del entorno de OpenDSS.
2	Hacer clic derecho para abrir el menú contextual y elegir la opción "Do Selected". Alternativamente, se puede utilizar el atajo de teclado Ctrl+D para ejecutar los comandos seleccionados. Este paso inicia la simulación o ejecuta la función seleccionada.
3	Después de ejecutar el código, OpenDSS proporciona resultados que pueden ser examinados para comprender el comportamiento de la red bajo las condiciones de prueba establecidas. Estos resultados pueden incluir detalles como perfiles de voltaje, flujos de corriente, pérdidas del sistema, y otros indicadores de rendimiento.
4	Al ejecutar este comando, OpenDSS genera reportes o visualizaciones que pueden ayudar al usuario a comprender mejor los resultados de las simulaciones o el estado de diferentes elementos del modelo de distribución eléctrica.

Tabla 2.2. Explicación de pasos de ejecución de código.

2.6 Sintaxis en OpenDSS

La sintaxis de OpenDSS será reportada mediante el uso de un sistema eléctrico modernos, aquel que considera elementos tradicionales y especiales conectados en la red eléctrica, como lo muestra la Figura 2.4.

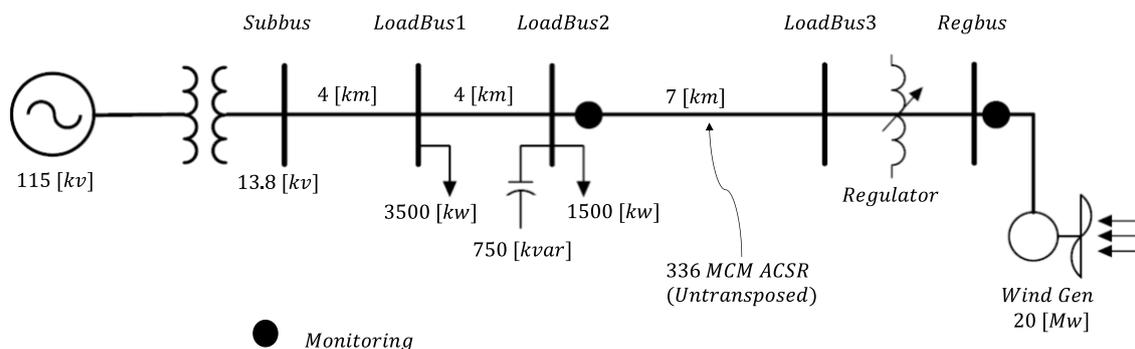


Figura 2.4 Sistema de ejemplo [29]

```
New object= circuit.NuevoNombre
~ basekv=115 1.00 0.0 60.0 3 600 660 10 4
```

Este comando de línea es completamente análogo al siguiente, donde se utilizan los comandos particulares de OpenDSS para asignar los nombres de los parámetros respectivos.

New object= circuit.NuevoNombre
 ~ basekv=115 pu=1.00 Angle=0.0 Frequency=60.0 Phases=3 Mvasc3=600
 Mvasc1=660
 ~ x1r1=10 x0r0=4

New object= circuit .NuevoNombre	Crea un nuevo objeto de circuito con el nombre "NuevoNombre"(el nombre lo asigna el usuario).
basekv=115	Establece la tensión base del circuito en 115 kV.
pu=1.00	Factor de potencia por defecto establecido en 1.00. Representa la potencia activa en relación con la potencia aparente.
Angle=0.0	Ángulo de fase en grados. En este caso, se establece en 0.0 grados.
Frequency=60.0	Frecuencia del sistema, en este caso, se establece en 60 Hz.
Phases=3	Número de fases del sistema, en este caso, se establece en 3.
Mvasc3=600	Capacidad máxima del sistema para cortocircuitos trifásicos (en MVA), se establece en 600.
Mvasc1=660	Capacidad de cortocircuito máxima para cortocircuitos monofásicos (en MVA), se establece en 660.
x1r1=10	Relación reactancia/resistencia para secuencia positiva. Este valor es 10.
x0r0=4	Relación reactancia/resistencia para secuencia cero. Este valor es 4.

Tabla 2.3. Descripción de parámetros del sistema

En OpenDSS, el comando Help como se observa en la Tabla 2.4 es una herramienta importante para obtener información detallada sobre diferentes aspectos del software. Al utilizar este comando, los usuarios pueden acceder a descripciones, ejemplos y detalles sobre la sintaxis de los comandos específicos, así como comprender mejor la funcionalidad y las opciones disponibles en OpenDSS.

Sección	Descripción
Consultar la documentación	Acceda a la información técnica detallada sobre elementos modelables como buses, líneas y cargas.
Aprender sobre comandos específicos	Obtenga instrucciones precisas sobre la sintaxis y uso de comandos específicos en OpenDSS.
Entender opciones y parámetros	Explore la variedad de opciones y parámetros para personalizar sus simulaciones a medida.
Clarificar conceptos y procedimientos	Reciba explicaciones claras sobre conceptos clave y guías paso a paso para procedimientos usuales.

Tabla 2.4. Detalles del comando "Help"

Pasos iniciales

- **Frecuencia del sistema:** Asegura que la frecuencia base del sistema esté configurada correctamente para coincidir con la red que se está modelando. Una frecuencia incorrecta puede llevar a simulaciones no válidas.

```
Set DefaultBaseFrequency = 60
```

- **Bus de origen (sourcebus):** Define el punto inicial de inyección de potencia en la red. Es crucial para establecer el flujo de potencia y las referencias de voltaje en todo el circuito.
- **Monitores de red:** Son esenciales para recopilar datos de la simulación y validar el comportamiento del modelo. Permiten la evaluación detallada del desempeño de la red bajo diversas condiciones.

Ajuste fundamental de componentes y dispositivos

- **Uso del comando "Clear".**

El comando clear en OpenDSS se utiliza para borrar o limpiar cualquier circuito previamente definido y liberar la memoria asociada. Al ejecutar el comando clear, se eliminan todos los elementos del circuito activo, incluyendo buses, líneas, cargas, transformadores, y otros componentes, dejando el entorno de trabajo en un estado limpio y listo para definir un nuevo circuito o cargar uno existente.

El uso de clear es útil cuando se desea cambiar o reiniciar el análisis de un sistema eléctrico en OpenDSS. Permite liberar la memoria ocupada por el circuito anterior y evitar posibles conflictos o interacciones no deseadas al definir un nuevo conjunto de elementos o al cargar un circuito diferente.

- ✓ Establecimiento de la ruta de trabajo. Esta acción resulta especialmente útil al trabajar con múltiples scripts que no necesariamente se encuentran en el mismo archivo. Su sintaxis es:

```
set datapatch = C:\OpenDSS\CircuitosEjemplos\
```

- **Iniciar el circuito nuevo.**

Dentro del entorno de OpenDSS, se inicia la creación de un circuito definiendo un objeto nuevo con el comando "New". A continuación, se procede a configurar el circuito y la fuente de alimentación principal. En este caso particular, la fuente consiste en una fuente de corriente alterna trifásica con un voltaje de línea de 115 kV, que funciona a una frecuencia estándar de 60 Hz y un ángulo de fase inicial establecido en cero grados.

```
New object= circuit.DSSTest  
~ basekv=115 pu=1.00 Angle=0.0 Frequency=60.0 Phases=3 Mvasc3=600  
Mvasc1=660  
~ x1r1=10 x0r0=4
```

- ✓ Para este caso designamos un nombre adecuado como es circuit.DSSTest.
- ✓ Los parámetros no esenciales restantes se emplean para lograr una modelación más precisa de la fuente de tensión. Por ejemplo, **Mvasc3** y **Mvasc1** estos son

los valores de cortocircuito trifásico y monofásico en MVA, que son estimaciones basadas en la capacidad del transformador y las impedancias de secuencia; mientras que **x1r1** y **x0r0** estas son las relaciones de reactancia a resistencia para secuencias positivas y cero, respectivamente.

- **Línea.**

En OpenDSS, el comando `linecode` se utiliza para definir las características eléctricas y geométricas de un conductor en términos de código de línea. Este comando permite a los usuarios especificar los parámetros que describen la impedancia de la línea y otros aspectos importantes del comportamiento eléctrico del conductor.

El `linecode` en OpenDSS abarca una variedad de propiedades, incluyendo resistencias, reactancias, capacidades y valores específicos relacionados con la geometría del conductor. Estos parámetros permiten a los ingenieros de sistemas eléctricos modelar de manera precisa y detallada la respuesta de las líneas en el sistema de distribución eléctrica.

Una vez que se ha definido un “`linecode`”, puede ser asignado a líneas específicas en el sistema utilizando el comando correspondiente. Esto proporciona una manera eficiente de representar diversos tipos de conductores y configuraciones de líneas en un sistema eléctrico, facilitando análisis detallados y precisos mediante simulaciones de flujo de potencia, cortocircuitos y otros estudios de sistemas eléctricos en OpenDSS.

El objeto de línea por defecto representa una línea aérea con una longitud de aproximadamente 1000 [ft] y un conductor de tipo 336 MCM ACSR donde las especificaciones se muestran en la Tabla 2.5, ubicado en una cruceta con una separación de 8 [ft] entre conductores. Los valores de los componentes simétricos están definidos en esta unidad de medida.

Componente	Valor [Ω por km]
R1	0.061
X1	0.1 – 0.3

C1	0.006 – 0.61
R0	0.183 – 0.61
X0	0.2 – 0.6
C0	0.006 – 0.012

Tabla 2.5. Valores de componentes de un conductor tipo 336 MCM ACSR [30][31]

- ✓ Los valores de R0 y X0 se basan en ser múltiplos de R1 y X1, respectivamente.
- ✓ Para R0, se ha tomado 3 veces el valor de R1 como mínimo y 10 veces como máximo.
- ✓ Para X0, se ha tomado 2 veces el valor de X1 como mínimo y 4 veces como máximo.
- ✓ Los valores están en ohmios por kilómetro (Ω/km) para la resistencia y reactancia, y en microfaradios por kilómetro ($\mu\text{F}/\text{km}$) para la capacitancia.

A continuación, se presenta un ejemplo nuevo de linecode:

```
! Definición de la geometría y propiedades del conductor
New WireData.Conductor336 GMR=0.0255 Diam=0.741 Rac=0.306
GMRunits=ft Radunits=in Racunits=ohm/mi
! Definición del código de línea usando las propiedades del conductor
New LineCode.336ACSR Phases=3 Frequency=60 Nphases=3 Nconds=4
R1=0.306 X1=0.25 R0=0.918 X0=0.75 C1=12 C0=10
! Creación de las líneas individuales
! Línea desde Subbus a LoadBus1
New Line.Line1 Phases=3 Bus1=Subbus Bus2=LoadBus1 Length=4
LineCode=336ACSR Units=km
! Línea desde LoadBus1 a LoadBus2
New Line.Line2 Phases=3 Bus1=LoadBus1 Bus2=LoadBus2 Length=4
LineCode=336ACSR Units=km
! Línea desde LoadBus2 a LoadBus3
New Line.Line3 Phases=3 Bus1=LoadBus2 Bus2=LoadBus3 Length=7
LineCode=336ACSR Units=km
```

- **Cargas.**

En OpenDSS, las cargas representan los puntos donde la energía eléctrica es consumida dentro del sistema de distribución. Las cargas son elementos fundamentales y su modelado afecta directamente los estudios de flujo de carga, calidad de energía y confiabilidad del sistema. Se asume que las cargas están balanceadas según el número de fases indicado. Si se requiere introducir cargas desbalanceadas, será necesario ingresar las cargas monofásicas de manera individual.

Las cargas en OpenDSS pueden ser modeladas y caracterizadas de múltiples maneras, pero principalmente se describen a través de tres formas:

- ✓ **Carga constante de potencia (PQ):**

- Esta es la forma más común de modelar una carga.
- La carga se define por una potencia activa constante (P) y una potencia reactiva constante (Q), independientemente de las variaciones de tensión.
- Este tipo de carga es útil para representar dispositivos que intentan mantener un consumo de energía constante.

! Carga Constante de Potencia (PQ)

```
New Load.Load1 Bus1=BusName Phases=1 Conn=Wye kW=120 kVAR=60  
kV=12.47 Model=1
```

- ✓ **Carga constante de corriente (I):**

- En este modelo, la carga consume una corriente constante.
- Esto significa que la potencia total consumida por la carga aumentará si la tensión aumenta, y disminuirá si la tensión disminuye.
- Este comportamiento es característico de ciertos tipos de equipos industriales o iluminación LED y fluorescente, donde la corriente no cambia mucho con la tensión.

! Carga Constante de Corriente (I)

New Load.Load2 Bus1=BusName Phases=1 Conn=Wye kW=75 kV=12.47

Model=2

✓ **Carga constante de impedancia (Z):**

- Aquí, la carga se modela como una impedancia constante.
- La potencia consumida por la carga varía con el cuadrado de la tensión.
- Este modelo es útil para representar la carga de dispositivos que actúan principalmente como resistencias o reactancias, como calentadores eléctricos o motores que operan a cargas muy ligeras.

! Carga Constante de Impedancia (Z)

New Load.Load3 Bus1=BusName Phases=1 Conn=Wye kW=50 kV=12.47

Model=3

A continuación, se presenta un ejemplo nuevo de cargas:

! Definición de las cargas en OpenDSS

! Carga en LoadBus1

New Load.LoadBus1 Bus1=LoadBus1 Phases=3 Conn=Wye kW=3500

kVAR=750 kV=13.8 Model=1 status=1

! Carga en LoadBus2

New Load.LoadBus2 Bus2=LoadBus2 Phases=3 Conn=Wye kW=1500

kVAR=750 kV=13.8 Model=1 status=1

New Load.LoadBus1 y New Load.LoadBus2	Se crean dos nuevos objetos de carga, LoadBus1 y LoadBus2.
Bus1=LoadBus1 y Bus2=LoadBus2	Se especifican los nombres de los buses a los que se conectan las cargas correspondientes.
Phases=3	Indica que las cargas son trifásicas.
Conn=Wye	Las cargas están conectadas en Y (estrella).

kW=3500 y kW=1500	Definen la potencia activa en kilovatios para cada carga.
kVAR=750	Establece la potencia reactiva en kilovar para ambas cargas.
kV=13.8	Indica la tensión de fase a neutro (para una carga en configuración Y, esta sería la tensión de fase; si las cargas estuvieran conectadas en delta, se debería proporcionar la tensión de línea a línea).
Loadmodel=1	Indica que se está utilizando un modelo de carga de potencia constante (PQ).
status=1	Indica que la carga está activa y participará en la simulación.

Tabla 2.6.Descripción de parámetros de carga

- **Transformador.**

Un transformador en OpenDSS se define mediante un conjunto de propiedades que incluyen las relaciones de voltaje, la conexión de los devanados, la impedancia entre devanados y las capacidades de potencia. Los transformadores pueden ser modelados con múltiples devanados (más de dos), lo cual permite representar transformadores de distribución o de potencia complejos.

```
New Transformer.SubXFMR Phases=3 Windings=2
~ Buses=[Subbus LoadBus1]
~ Conns=['Delta' 'Wye']
~ kVs=[115 13.8]
~ kVAs=[30000 30000]
~ XHL=5
```

New Transformer.Transf1	Inicia la declaración de un nuevo transformador llamado "Transf1".
Phases=3	Número de fases eléctricas del transformador.
Windings=2	Cantidad de devanados que posee el transformador.
Buses=[Subbus LoadBus1]	Buses a los que están conectados los devanados del transformador.
Conns=['Delta' 'Wye']	Configuraciones de conexión de los devanados (Delta para alta tensión, Wye para baja tensión).
kVs=[115 13.8]	Tensiones nominales de los devanados de alta y baja tensión (en kV).
kVAs=[30000 30000]	Capacidad nominal de los devanados en kVA.
XHL=5	Reactancia de dispersión en porcentaje entre los devanados de alta y baja tensión.

Tabla 2.7. Descripción de parámetros transformador

- **Capacitores.**

Los capacitores en OpenDSS se utilizan para mejorar la calidad de la energía en los sistemas de distribución eléctrica. Su función principal es proporcionar compensación de potencia reactiva y mejorar el factor de potencia en la red. Esto se traduce en una reducción de las pérdidas en la línea y una mejora en el perfil de voltaje a lo largo del sistema. Pueden ser utilizados para diversas funciones en la simulación de redes eléctricas, aquí se presentan algunas características:

- ✓ Variedad de capacidades.
- ✓ Diversos modos de operación.
- ✓ Implementación de controles inteligentes.
- ✓ Soporte para capacitores distribuidos.
- ✓ Modelado de pérdidas.
- ✓ Interacción con otros dispositivos.
- ✓ Respuesta a eventos del sistema.
- ✓ Análisis de impacto a la red.
- ✓ Compatibilidad con análisis de estado estable y dinámico.

- ✓ Estudios de armónicos.

A continuación, se presenta un ejemplo nuevo de capacitor:

New Capacitor.Cap1 Bus1=LoadBus2 Phases=3 kVAR=750 kV=13.8
Conn=Wye

New Capacitor.Cap1	Crea un nuevo objeto capacitor llamado "Cap1".
Bus1=LoadBus2	Especifica que el capacitor está conectado al bus LoadBus2.
Phases=3	Indica que el capacitor es trifásico.
kVAR=750	Establece la capacidad del capacitor. Este valor es el necesario para compensar la potencia reactiva en ese bus.
kV=13.8	Es el nivel de voltaje en el que opera el capacitor, que corresponde al voltaje de línea a neutro del sistema si está configurado en Wye.
Conn=Wye	Señala que el capacitor está conectado en configuración estrella.

Tabla 2.8. Descripción de parámetros de capacitor

- **Generador.**

Los generadores en OpenDSS son componentes críticos para la simulación de fuentes de generación de energía dentro de una red de distribución eléctrica. Pueden representar una amplia variedad de recursos de generación, como plantas de energía convencionales, sistemas de cogeneración, y fuentes de energía renovable, incluyendo parques eólicos y solares. A continuación, se presenta algunas características el software OpenDSS.

- ✓ Modelamiento de generadores.
- ✓ Control de generación.
- ✓ Operación sincronizada y asíncrona.
- ✓ Parámetros principales: kW, kVAR, kV, pf, phases.
- ✓ Soporte para almacenamiento de energía.

- ✓ Interacción con protecciones.
- ✓ Análisis de impacto en la red.

New Generator.WindGen1	Crea un nuevo objeto generador llamado "WindGen1".
Bus1=Regbus	El bus al que está conectado el generador.
Phases=3	Número de fases eléctricas del generador.
kV=13.8	Nivel de voltaje en el bus de generación.
kW=20000	Potencia activa que el generador suministra al sistema
pf=0.9	Factor de potencia del generador.
Model=1	Modelo de generación utilizado (potencia constante).

Tabla 2.9. Descripción de parámetros de generador

- **Monitores.**

Los monitores que se muestran en OpenDSS son dispositivos de simulación fundamentales que permiten a los usuarios rastrear y registrar varios tipos de datos eléctricos dentro del sistema de distribución. Estos dispositivos son esenciales para analizar el comportamiento del sistema durante las simulaciones, y pueden ayudar a identificar problemas y a optimizar el rendimiento. Mencionaremos algunas características presentes:

- ✓ Función de monitoreo.
- ✓ Tipos de datos registrados.
- ✓ Configuración flexible.
- ✓ Resolución temporal.
- ✓ Análisis de eventos.
- ✓ Modos de monitoreo.
- ✓ Soporte para análisis de armónicos.
- ✓ Interfaz de análisis de datos.

New Monitor.LineMonitor1	Crea un nuevo monitor llamado "LineMonitor1".
Element=Line.Line2	El elemento del sistema (línea, transformador, etc.) que está siendo monitoreado, que es el nombre que le daríamos a la línea que conecta LoadBus2 y LoadBus3
Terminal=1	Terminal del elemento al cual el monitor está conectado.
Mode=1	Modo de operación del monitor (potencia y energía).

Tabla 2.10. Descripción de parámetros de monitor

- **Vehículos eléctricos.**

Los vehículos eléctricos en OpenDSS generalmente se tratan como cargas dinámicas más que como generadores distribuidos. Su consumo de energía varía significativamente, influenciado por varios factores como: los hábitos de carga de los usuarios, las horas del día de mayor demanda eléctrica y el tipo específico de vehículo. Para reflejar esta variabilidad en el consumo, OpenDSS emplea "loadshapes" o perfiles de carga. Estos perfiles son cruciales para simular cómo cambia la demanda eléctrica de los vehículos eléctricos a lo largo del tiempo, adaptándose a diferentes circunstancias como el ciclo diario, los patrones semanales o los cambios estacionales. Un ejemplo claro de esto sería una estación de carga para vehículos eléctricos, cuyo uso energético es relativamente bajo durante las horas diurnas, pero que experimenta un incremento notable por la noche, coincidiendo con el momento en que los vehículos suelen conectarse para recargar sus baterías [32].

- **Loadshape.**

Es una herramienta en OpenDSS que se utiliza para definir cómo varía la carga eléctrica con el tiempo en un sistema de distribución. El comando loadshape incluye una serie de multiplicadores aplicados a las cargas base. Estos multiplicadores de 0

a 1 se ajustan para reflejar la variación en la demanda de electricidad durante diferentes periodos del día o del año.

```
New loadshape.shortday 4 6  
~ mult=(0.5 1.0 0.8 0.4)
```

El ejemplo proporcionado está dividido en 4 periodos de 6 horas cada uno. Los multiplicadores "mult" reflejan una demanda menor en la madrugada (0.5), un pico durante el mediodía (1.0), una disminución en la tarde (0.8), y luego una baja demanda nuevamente en la noche (0.4). Este tipo de perfil puede ser útil para análisis rápidos o situaciones donde se requiere una modelización menos detallada de la variación de la carga.

- **Comando Solve.**

El comando **Solve** en OpenDSS sirve para realizar el cálculo del flujo de carga en una red eléctrica modelada en el programa. Al activar este comando, el software procesa y proporciona los valores de corriente y voltaje en toda la red, teniendo en cuenta las configuraciones del modelo, las cargas, la generación y otros parámetros operativos. Este proceso es esencial para entender cómo se comporta el sistema y para fundamentar las decisiones relacionadas con su diseño y operación.

El propósito de este capítulo es detallar cómo se debe elaborar un script correcto para usar en OpenDSS. Esto es fundamental para que, al abordar el tercer capítulo de esta tesis, tengamos una comprensión clara de los resultados esperados de ChatGPT. Dado que ChatGPT proporcionará los scripts o códigos que serán validados posteriormente en OpenDSS, a continuación, exploraremos cómo funciona ChatGPT y sus diversas características, con el fin de aprovecharlo en los sistemas de potencia.

CAPÍTULO 3

3. CARACTERIZANDO A CHATGPT

Este capítulo se adentra en el cruce de caminos entre la inteligencia artificial (IA) y la ingeniería eléctrica, centrándose en ChatGPT, una innovación de OpenAI. Se resalta cómo este modelo de lenguaje, tanto sofisticado como adaptable, es competente en un amplio abanico de temas. Más allá de dar respuestas detalladas y bien informadas en áreas generales, ChatGPT puede abordar tópicos de ingeniería eléctrica y programación.

Esta sección ilustra la habilidad de ChatGPT para aplicar su conocimiento y comprensión en ingeniería, convirtiéndose en un recurso valioso para el análisis y la resolución de problemas en el sector eléctrico. Esta herramienta no solo responde preguntas con exactitud, sino que también genera datos y soluciones adaptados a situaciones reales, siendo así de gran utilidad para ingenieros, investigadores, y otros expertos en el área de energía eléctrica.

Por otro lado, ChatGPT también presenta una habilidad para interactuar con otros sistemas y herramientas computacionales, mejorando su efectividad. Se investiga cómo su interacción con programas de modelado, simulación y análisis de datos fomenta un flujo de trabajo más eficaz.

Dado lo antes mencionado, este capítulo presenta una visión completa de cómo ChatGPT puede revolucionar el enfoque de los profesionales eléctricos en sus actividades cotidianas, abarcando desde la planificación y diseño de proyectos hasta el diagnóstico y solución de problemas técnicos.

3.1 Qué es ChatGPT

ChatGPT es una herramienta avanzada de OpenAI creado para replicar el diálogo humano, utiliza un entendimiento detallado y contextualizado del lenguaje, lo cual facilita la producción de respuestas no solo pertinentes, sino también notablemente similares a las humanas en cuanto a su tono y forma. Se ha capacitado mediante una extensa serie de interacciones textuales, que cubren un diverso espectro de temas y formas de comunicación. Este modelo se destaca por su capacidad para anticipar y reaccionar de forma coherente ante una variedad de consultas, adaptándose a diferentes ambientes y estilos de conversación [33].

ChatGPT ha progresado notablemente, con cada versión, mejorando en exactitud, rapidez y adaptabilidad. Este progreso continuo ha enriquecido la interacción humano-máquina, impulsando avances en la IA. Con mejoras en el entendimiento y la naturalidad de las respuestas, ChatGPT se ha vuelto esencial en múltiples aplicaciones. En la siguiente sección, exploraremos cómo ChatGPT ha progresado desde sus inicios hasta convertirse en un modelo aplicable en el campo.

3.2 Evolución de ChatGPT

La evolución de ChatGPT se muestra en la Figura 3.1. Desde su primera versión hasta ChatGPT-4, la IA ha mostrado un desarrollo significativo en términos de capacidad de procesamiento y sofisticación del modelo. Cada iteración ha mejorado en términos de tamaño del modelo, comprensión contextual, y habilidad para generar respuestas coherentes en múltiples áreas. Los modelos posteriores, como GPT-3 y GPT-4, han incorporado características avanzadas como el aprendizaje multitarea, donde las diferencias que tienen dichas versiones en contraste con las anteriores son bastantes notorias, lo que les permite adaptarse a nuevas tareas con datos mínimos de entrenamiento. A medida que los modelos evolucionaron, también lo hicieron sus aplicaciones en una variedad de dominios, demostrando una mejora en su versatilidad y eficiencia [23].



Figura 3.1 Línea de tiempo de las diferentes versiones de ChatGPT

GPT-1	Se centró en aprender de un extenso conjunto de textos para prever palabras sucesivas, lo cual permitió que el modelo adquiriera una comprensión básica de los patrones lingüísticos.
GPT-2	Con un salto cualitativo en la cantidad de parámetros, GPT-2 perfeccionó su predecesor al mejorar la generación de texto, ofreciendo una mayor cohesión y adaptabilidad en una amplia gama de contextos lingüísticos.
GPT-3	Este modelo amplió dramáticamente la escala y eficiencia, capaz de realizar múltiples tareas de procesamiento de lenguaje y adaptarse a nuevos desafíos con mínimos datos adicionales, facilitando su aplicación en diversos dominios tecnológicos y creativos.
GPT-3.5	Destaca por su aprendizaje rápido y su habilidad para abordar tareas nuevas sin entrenamiento previo, simplificando el proceso de desarrollo al minimizar la dependencia de grandes volúmenes de datos etiquetados.
GPT-4	La versión más reciente de la serie GPT avanza hacia una mayor complejidad, tratando no solo texto sino también imágenes, y alcanzando niveles de desempeño comparables a los humanos en contextos especializados, aunque aún con margen de mejora en cuanto a precisión y confiabilidad.

Tabla 3.1. Detalles de las diferentes versiones de ChatGPT

Los modelos GPT mostrados en la Tabla 3.1 han marcado un desarrollo progresivo en el campo de la IA, mejorando la forma en que las máquinas entienden y generan lenguaje humano. Desde sus inicios con GPT-1 hasta el sofisticado GPT-4, han mostrado avances significativos en la interpretación y producción lingüística. Cada nueva versión ha ampliado el alcance de la inteligencia artificial, ofreciendo herramientas más potentes que enriquecen la manera en que interactuamos con los dispositivos tecnológicos.

3.3 Beneficios y desafíos de ChatGPT

ChatGPT resalta entre las soluciones tecnológicas por los beneficios significativos que aporta a la eficiencia operativa y la reducción de costes en servicios de atención al cliente, derivados de la automatización de interacciones. Su entrenamiento previo le permite ofrecer respuestas coherentes y naturales, superando a otros sistemas de procesamiento de lenguaje natural en cuanto a claridad y naturalidad. Adicionalmente, su diseño de aprendizaje automático le confiere la capacidad de automejorarse continuamente al aprender de interacciones previas. Sin embargo, los desafíos de ChatGPT no son menores, especialmente en lo que respecta a la seguridad. Entre los riesgos se incluyen ataques adversarios con intenciones maliciosas, el potencial uso para propagar información errónea y el peligro de suplantación de identidad. Además, ChatGPT enfrenta limitaciones técnicas, como la dependencia del input recibido y la falta de acceso a información actualizada de la web, lo cual limita su capacidad para abordar preguntas muy complejas o proveer datos en tiempo real. Ante estos desafíos, es imperativo que los usuarios estén conscientes de estas limitaciones para utilizar ChatGPT de forma efectiva y segura [34].

3.4 Interfaz de ChatGPT

La interfaz de ChatGPT mostrada en la Figura 3.2 busca ser intuitiva y fácil de usar para que los usuarios puedan enviar mensajes y recibir respuestas generadas por el modelo, facilitando la comunicación en lenguaje natural. ChatGPT puede aceptar preguntas, proporcionar información, admitir archivos o simplemente conservar conversaciones informales.

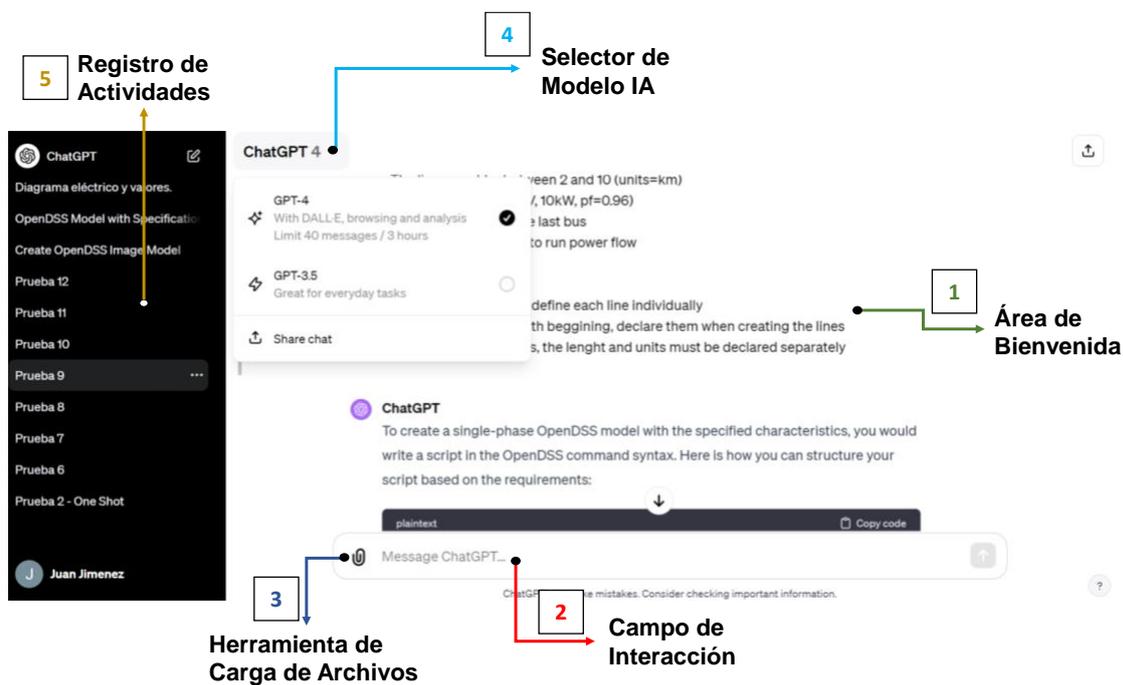


Figura 3.2 Interfaz de ChatGPT 4

En la Tabla 3.2 muestra la interfaz de ChatGPT puede dividirse en cinco principales componentes para que los usuarios tengan en consideración al momento de solicitar o realizar lo que requiera.

1	<ul style="list-style-type: none"> • Punto de inicio para las interacciones del usuario con la IA. • Puede contener mensajes de introducción, instrucciones de uso o sugerencias de temas de conversación. • Diseñada para ser visualmente clara y fácil de entender para los nuevos usuarios.
2	<ul style="list-style-type: none"> • El núcleo de la interfaz donde los usuarios y ChatGPT se comunican. • Muestra tanto las entradas del usuario como las respuestas de la IA en tiempo real. • Incluye un área de visualización de texto donde se pueden seguir las conversaciones en curso.
3	<ul style="list-style-type: none"> • Aprovechar una interfaz intuitiva y fácil de usar para adjuntar archivos, imágenes. • Mejorar la precisión y el contexto de las conversaciones con ChatGPT.

4	<ul style="list-style-type: none"> • Permite a los usuarios elegir entre diferentes versiones del modelo de lenguaje, optimizadas para diversas tareas. • Cada opción del modelo podría estar acompañada de una breve descripción o un conjunto de características esperadas. • Posibilita a los usuarios seleccionar la capacidad de procesamiento adecuada a sus necesidades específicas.
5	<ul style="list-style-type: none"> • Proporciona un acceso rápido a conversaciones anteriores o a ejemplos de modelos. • Permite a los usuarios guardar, recuperar o continuar trabajos previos. • Facilita la organización y la recuperación de datos de las interacciones pasadas para referencia futura o continuación del trabajo.

Tabla 3.2 Descripción de la interfaz de ChatGPT 4

Hay que recordar que ChatGPT-4 mejora significativamente la interacción del usuario, ofreciendo nuevas características. A diferencia de ChatGPT-3.5, que se limita más al manejo de texto, la versión 4 enriquece la experiencia al añadir la habilidad de procesar imágenes y ajustarse de manera más eficiente durante las conversaciones en vivo. Además, esta versión avanzada ofrece una generación de texto más coherente y contextual, lo que conduce a diálogos más detallados y acertados.

3.5 Comparación entre ChatGPT versión 3.5 y 4

Se presenta en la Tabla 3.3 un cuadro comparativo entre dos versiones de ChatGPT, identificadas como ChatGPT-3.5 y ChatGPT-4. Se resalta una progresión en la sofisticación tecnológica de la inteligencia artificial desde la versión 3.5 a la versión 4. Cada columna contrasta diversas características clave entre las dos versiones, mostrando avances significativos en varias áreas funcionales.

Características	ChatGPT-3.5	ChatGPT-4
Compresión de texto	Avanzada, pero puede perder el contexto en interacciones más largas o complejas.	Aún más avanzada, mejor en mantener el contexto a través de diálogos extensos.
Generación de texto	Coherente y fluida, adecuada para una amplia variedad de tareas.	Más refinada, con mejoras en coherencia y relevancia contextual.
Actualizaciones y soportes	Actualizaciones periódicas.	Soporte continuo con actualizaciones regulares.
Funcionalidades avanzadas	Principalmente texto.	Puede incluir procesamiento avanzado, archivos y comprensión de imágenes.
Interacción con imágenes	No tiene la capacidad de procesar o entender imágenes.	Potencialmente capaz de entender y generar descripciones de imágenes o incluso integrar texto e imagen en sus respuestas.
Adaptabilidad	Aprende con cada interacción hasta cierto punto, pero no en tiempo real.	Capacidad potencial para aprender y adaptarse en tiempo real durante una conversación.
Respuestas a preguntas específicas	Buen rendimiento en preguntas y respuestas, con algunas respuestas fuera de contexto.	Buen rendimiento en preguntas y respuestas, con algunas respuestas fuera de contexto.
Costo	Versión gratuita	Versión pagada

Respuestas a consultas técnicas	Respuestas técnicas adecuadas, pero a veces superficiales.	Profundidad y precisión mejoradas en temas técnicos y especializados.
Extensión de respuestas	Produce respuestas con una longitud limitada y a veces necesita ser más conciso.	Produce respuestas con una longitud limitada y a veces necesita ser más conciso.
Uso en tiempo real	Adecuado para uso en tiempo real, pero con ciertas limitaciones en la rapidez de respuesta.	Mejoras en la capacidad de procesamiento para respuestas más rápidas en aplicaciones de tiempo real.

Tabla 3.3. Cuadro comparativo entre las versiones 3.5 y 4 de ChatGPT

En el aspecto de comprensión de texto, se nota una evolución hacia una mejor retención del contexto en conversaciones extensas y complejas, lo cual sugiere una mejora en el entendimiento y la coherencia de las respuestas proporcionadas por la versión más avanzada.

En términos de generación de texto, hay una mejora en la fluidez y adecuación del contenido generado, posiblemente reflejando una capacidad más refinada para comprender y responder a una gama más amplia de tareas.

La interacción con imágenes marca una diferencia notable, donde la versión anterior no tenía la capacidad de procesar imágenes, mientras que la versión actualizada puede no solo entender sino también generar descripciones y contextos relacionados con contenido visual.

En cuanto a la adaptabilidad, se observa que la nueva versión puede aprender y adaptarse en tiempo real durante una conversación, lo que implica una interacción más dinámica y personalizada.

La extensión de las respuestas y el uso en tiempo real se mantienen similares en ambas versiones, sugiriendo que, aunque hay mejoras en la calidad y precisión de las respuestas, la concisión y la eficiencia temporal son aspectos que conservan su importancia en la interacción con el usuario.

La respuesta a preguntas específicas y las respuestas a consultas técnicas muestran una profundización en la versión 4, indicando una mayor precisión y especialización, lo que puede ser particularmente beneficioso para usuarios que buscan información técnica o especializada.

Se menciona un cambio en el modelo de costos, pasando de una versión gratuita a una de pago, implicando que los avances en la funcionalidad vienen con un cambio en el modelo de negocio.

La versión ChatGPT-4 es superior a la versión 3.5, con mejoras en la comprensión y generación de texto, capacidad para interactuar con imágenes, aprendizaje en tiempo real, y respuestas más especializadas y técnicas.

3.6 Limitaciones y potencial de ChatGPT en generación de texto

Basado en esta investigación preliminar, es posible concluir que, para áreas específicas como sistemas de energía, donde los datos de entrenamiento en internet no son tan abundantes y amplios como los datos de otras áreas (por ejemplo, programación de propósito general), la generación de texto desde cero no funciona satisfactoriamente. Para obtener respuestas apropiadas, es importante proporcionar uno o más ejemplos primero (aprendizaje de uno o pocos ejemplos). Incluso GPT-4, el modelo más poderoso, no puede generar resultados adecuados sin ejemplos previos.

Por otro lado, cuando los errores se presentan adecuadamente a ChatGPT con información más detallada, la mayoría de los errores simples pueden corregirse rápidamente. Sin embargo, para problemas que involucran lógica más compleja, el

modelo tiene dificultades para solucionar los problemas incluso con una orientación exhaustiva [35].

ChatGPT se destaca como una herramienta multifacética y eficaz, con la capacidad de influir de manera positiva en una variedad de sectores. Aunque enfrenta ciertas limitaciones, como la necesidad de información detallada para respuestas más precisas, su habilidad para analizar y sintetizar información lo hace un recurso invaluable para la solución de problemas y el avance del conocimiento, especialmente en áreas donde son esenciales la innovación y la capacidad de adaptarse a nuevos desafíos.

Posteriormente, se detalla cómo podemos proporcionar información a ChatGPT para aumentar su eficiencia. Es esencial suministrar datos y ejemplos específicos y relevantes, lo que permite al modelo comprender mejor el contexto y los requerimientos de la consulta. La precisión en la descripción del problema o la pregunta, junto con una retroalimentación constructiva en caso de errores, resulta clave para optimizar la respuesta de ChatGPT. Al seguir estos lineamientos, se puede guiar eficazmente a ChatGPT hacia resultados más acertados y útiles.

Al adentrarnos en el uso de ChatGPT para la ejecución de tareas específicas, es esencial entender las directrices que guían su funcionamiento. Estas directrices son fundamentales para aprovechar al máximo las capacidades de la IA, permitiendo su aplicación efectiva en una amplia gama de áreas profesionales. A continuación, exploraremos cómo formular indicaciones claras y efectivas para ChatGPT, y proporcionaremos ejemplos prácticos de cómo estas indicaciones pueden traducirse en acciones concretas, optimizando así la interacción con esta poderosa herramienta.

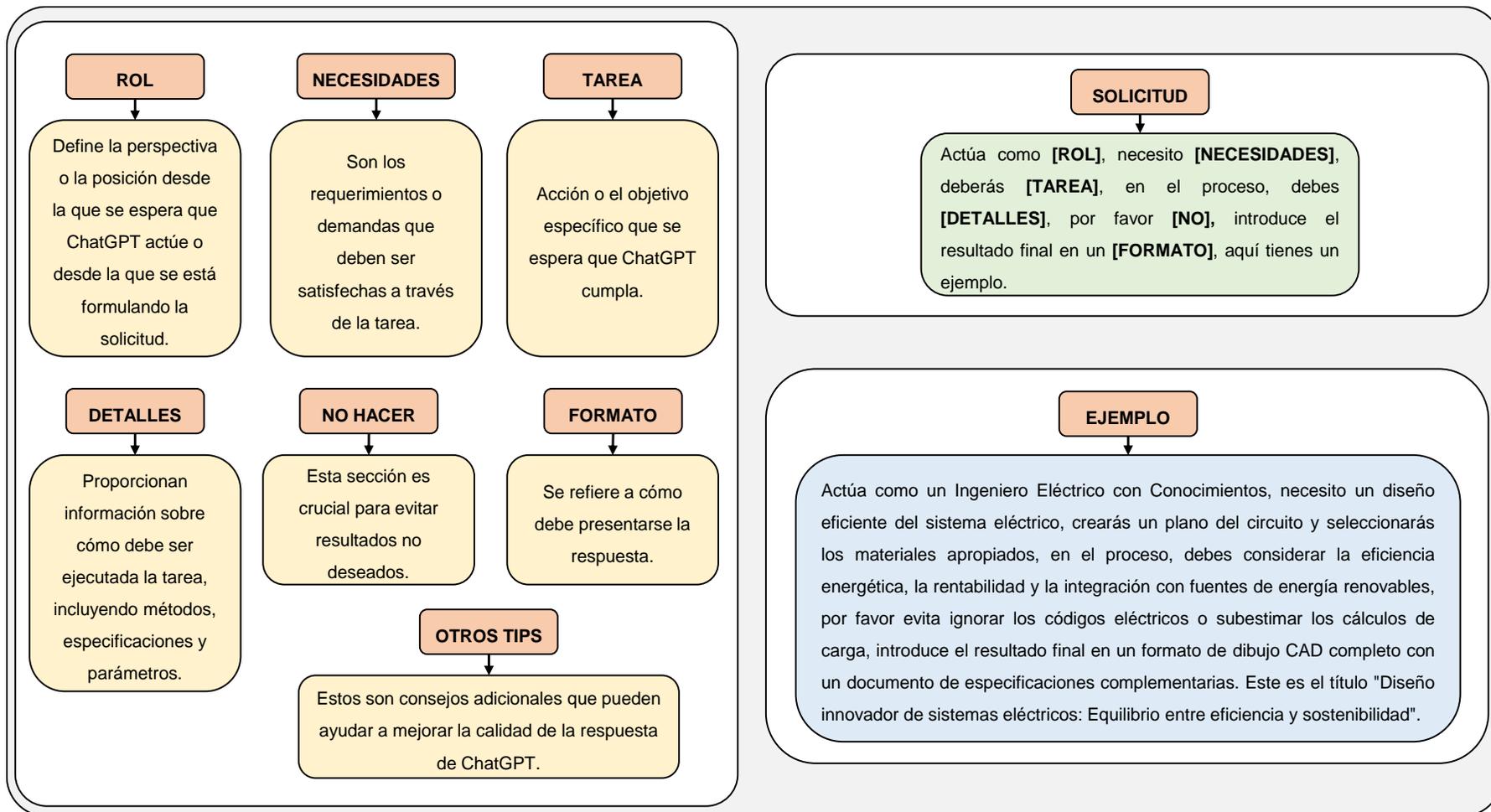


Figura 3.3 Características del funcionamiento de ChatGPT

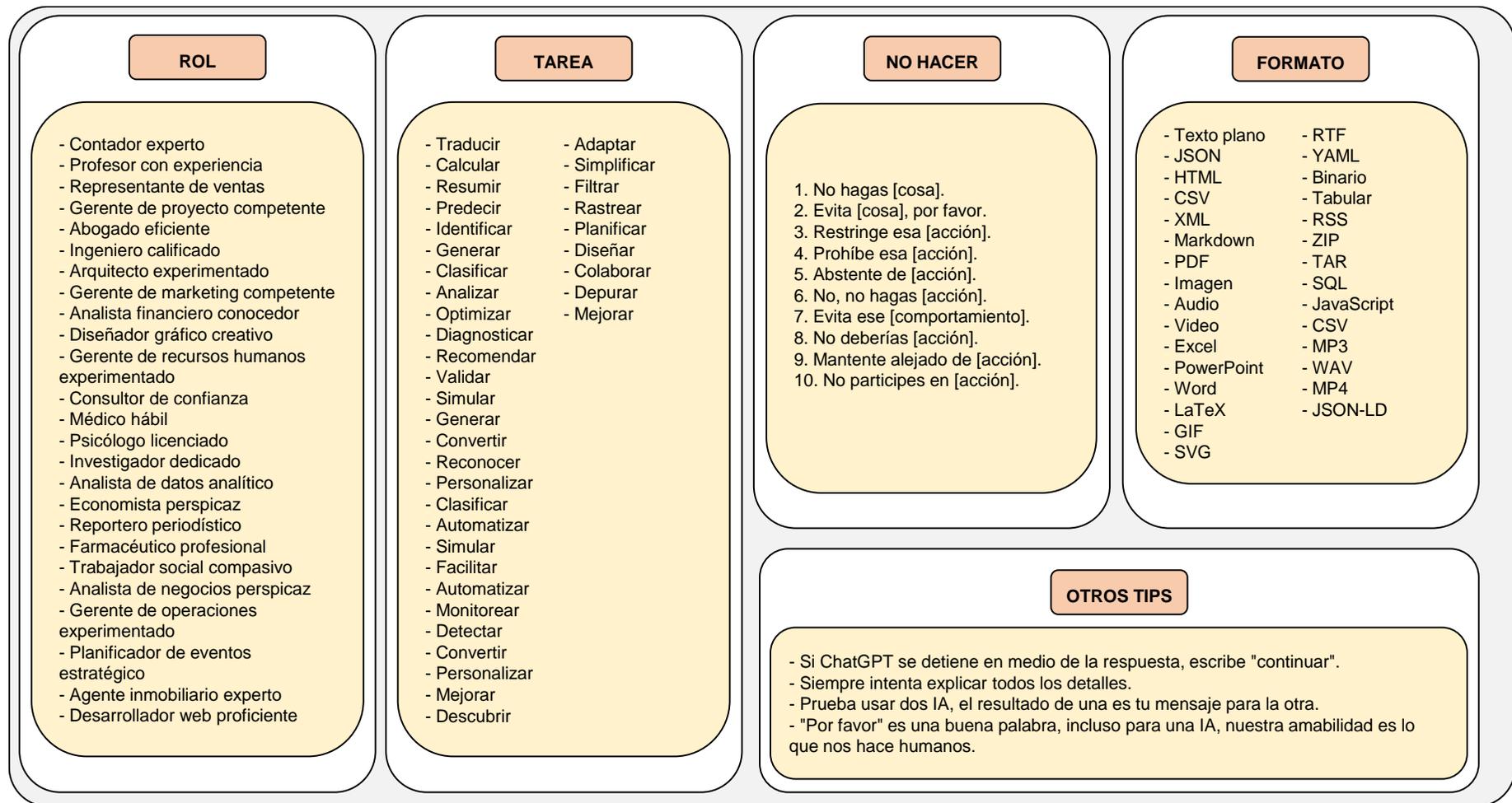


Figura 3.4 Ejemplos específicos para indicar una tarea a ChatGPT

En la Figura 3.3 se muestra una guía para mejorar la comunicación con ChatGPT, estableciendo un método claro para formular peticiones. Explica la forma en que se deben detallar las expectativas de las tareas a realizar por ChatGPT, las precauciones que se deben tomar durante su ejecución y el modo en el que se deben recibir las respuestas. Además, proporciona ejemplos prácticos y consejos útiles para afinar el diálogo y obtener la información deseada de manera efectiva. El propósito es asistir a los usuarios en la creación de solicitudes detalladas y enfocadas para obtener respuestas pertinentes y de calidad de la inteligencia artificial. Por otra parte, y haciendo un complemento, en la Figura 3.4 se muestra una estructura sugerida para comunicarse efectivamente con un asistente digital como ChatGPT. Presenta una variedad de roles y las tareas asociadas que se podrían asignar, identifica lo que se debe evitar para no comprometer la integridad de las instrucciones y detalla los diferentes formatos posibles para la entrega de los resultados. Proporciona también recomendaciones para optimizar la interacción y garantizar una comunicación efectiva con la inteligencia artificial. El propósito es proporcionar a los usuarios una dirección clara sobre la manera de utilizar un asistente de IA para obtener respuestas detalladas y adecuadas, teniendo en cuenta las restricciones específicas y las preferencias de formato.

3.7 ChatGPT en Sistemas de Potencia

La inteligencia artificial introducida en los sistemas de potencia, encabezada por ChatGPT, se presenta como un modelo avanzado de procesamiento de lenguaje natural, el cual se destaca en tareas que requieren comprensión y generación de texto, lo cual la hace especialmente útil en varias áreas del sector eléctrico, incluido el diseño conceptual de topologías en sistemas de distribución.

La capacidad de ChatGPT brinda apoyo significativo en la creación y afinación de scripts de programación, particularmente en aquellos relacionados con la modelización y análisis de sistemas de energía. Por ejemplo, puede ayudar a redactar algoritmos que pronostiquen patrones de carga o mejorar la eficiencia de las redes. Además, es útil en la interpretación de resultados, siempre que la pregunta sea específica y clara, y en proporcionar explicaciones sobre conceptos complejos en sistemas de potencia. Para aplicaciones específicas, especialmente en programación o diseño de sistemas, ChatGPT necesita ser "entrenado" o previamente facilitado de ejemplos. Esto significa

que, para obtener la mejor ayuda de ChatGPT, es importante proporcionarle contextos claros, ejemplos de código o descripciones detalladas de lo que se intenta obtener. ChatGPT puede entonces ayudar a elaborar sobre esa base, generar ideas o explicar conceptos, pero siempre dentro del marco de la información y los ejemplos proporcionados por el usuario.

3.8 Comparación Sistema Eléctrico de Potencia (SEP): Tradicional vs Innovación

La Figura 3.5 captura dos modelos distintos en la gestión de los SEP. Por un lado (1), observamos un enfoque clásico, donde la operación y supervisión se realizan de manera tradicional. Por otro lado (2), vemos un entorno altamente tecnológico, donde la inteligencia artificial juega un papel crucial en la automatización y análisis de datos. Juntas, estas escenas reflejan la evolución de las prácticas de gestión en el sector eléctrico, desde los métodos manuales hasta las soluciones avanzadas basadas en IA, resaltando el impacto de la tecnología en la eficiencia y complejidad de los procesos actuales en el ámbito energético.



Figura 3.5 Comparación entre sistemas de distribución tradicional y modernos

Figura 3.5 - Imagen 1:

La imagen muestra una sala de control de sistemas eléctricos de potencia que opera con tecnologías de monitoreo y control tradicional. Un operador se encuentra sentado frente a un escritorio equipado con monitores y paneles de control. En el fondo, una gran pizarra presenta un diagrama esquemático de un sistema eléctrico, una herramienta estándar para visualizar las conexiones y componentes de la red. Donde el operador parece estar supervisando y gestionando el sistema de potencia utilizando un sistema SCADA indicado por la interfaz en las pantallas. La configuración refleja una operación sin inteligencia artificial, enfatizando la importancia del monitoreo humano y la intervención directa. A pesar de la ausencia de IA, la sala está bien equipada con la tecnología necesaria para mantener el sistema funcionando de manera eficiente y responder a las necesidades operativas.

Figura 3.5 - Imagen 2:

La imagen muestra una sala de control de sistemas eléctricos de potencia que integra la inteligencia artificial en sus operaciones. Observamos varios operadores trabajando en puestos de trabajo modernas con múltiples monitores, que enseñan gráficos complejos y datos en tiempo real. La gran pantalla en el centro de la sala resalta la "Gestión de Datos y Análisis de IA", lo que implica que la IA está analizando y supervisando activamente la red eléctrica, se puede decir que está optimizando el rendimiento y prediciendo necesidades de mantenimiento. Las visualizaciones detalladas y las interfaces interactivas indican un alto nivel de automatización y sofisticación, lo que permite a los operadores manejar el sistema con mayor eficiencia y tomar decisiones informadas basadas en recomendaciones de la inteligencia artificial.

Hay que recordar que, si bien ChatGPT se presenta como una solución de inteligencia artificial innovadora, no está libre de desafíos. Su capacidad para generar respuestas acertadas en dominios especializados es limitada si carece de datos o ejemplos concretos previos. Ante escenarios complejos o información técnica avanzada, ChatGPT puede necesitar instrucciones precisas para formular respuestas apropiadas. La falta de representatividad en su aprendizaje puede afectar la calidad de sus respuestas. Por lo

tanto, es crucial reconocer y trabajar en torno a estas limitaciones para desbloquear su verdadero potencial.

3.9 Aplicaciones generales de ChatGPT en el campo eléctrico.

Al explorar las aplicaciones generales de ChatGPT en sistemas de potencia, nos enfocamos en cómo esta herramienta de inteligencia artificial se está utilizando para optimizar y revolucionar este sector. En La Tabla 3.4, exploraremos cómo ChatGPT se ha integrado en el campo de la ingeniería eléctrica. Además, discutiremos las aplicaciones generales de esta tecnología, abordando desde la automatización de tareas de programación hasta su influencia en la toma de decisiones y la educación, así como la evaluación de su desempeño en contextos específicos de ingeniería eléctrica. Este análisis profundo nos proporcionará una clara comprensión de cómo ChatGPT se está convirtiendo en una herramienta indispensable en este campo.

Tema general	Ref	Acotación
Automatización de tareas programas	[36]	ChatGPT se utiliza para automatizar tareas de programación complejas, escribir y optimizar códigos para algoritmos numéricos y aplicaciones en aprendizaje automático.
Soporte en tomas de decisiones y educación	[37]	El papel de ChatGPT como herramienta de apoyo en la toma de decisiones y la educación en el campo de los sistemas de potencia es cada vez más significativo.
Evaluación en ingeniera eléctrica	[38]	Se han desarrollado estructuras para evaluar el rendimiento de los modelos de lenguaje grande como ChatGPT en tareas de ingeniería eléctrica.

Tabla 3.4 Aplicaciones de ChatGPT en el campo de la electricidad

- **Automatización de tareas programas**

En las Figuras 3.2 y 3.3 se muestra cómo un asistente virtual, como ChatGPT, impulsa la innovación y la eficiencia en el trabajo de los ingenieros eléctricos. Este sistema sirve inicialmente como una fuente de ideas innovadoras, ofreciendo métodos y fomentando

el aprendizaje para ampliar habilidades técnicas. En labores habituales, el asistente facilita el proceso al crear y optimizar códigos, reduciendo así el tiempo de programación y corrección de errores.

En desafíos más complejos, como el diseño de solucionadores numéricos, el asistente aporta orientación valiosa, simplificando la formulación de problemas y la elección de los solucionadores adecuados. Frente a problemas nuevos, como acelerar la gestión operativa sin soluciones predefinidas, proporciona una estructura para desarrollar enfoques innovadores, desde la planificación tecnológica hasta la generación de datos y el desarrollo de modelos y códigos pertinentes. El gráfico también subraya un proceso iterativo de mejora y evaluación, donde el asistente colabora estrechamente con los ingenieros para perfeccionar las metodologías y mejorar los resultados. Este enfoque de retroalimentación continua promueve la mejora constante en los procesos de programación y modelado, ayudando a los ingenieros a realizar sus tareas con mayor eficacia y precisión.

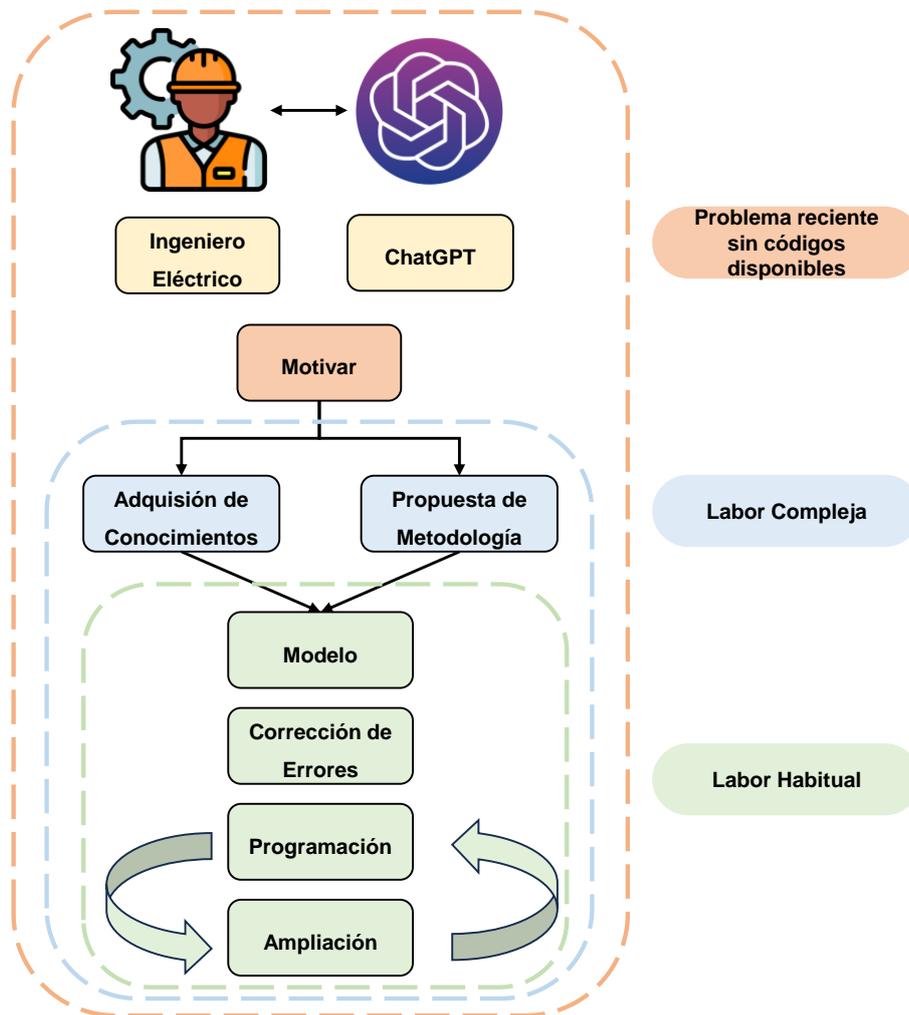


Figura 3.6 Marco de programación interactivo apoyado en ChatGPT

Las Figuras 3.6 y 3.7 fueron inspiradas por [36], esto revela una integración fascinante de ChatGPT en el ciclo de trabajo de un ingeniero eléctrico. En la Figura 3.6 comienza con la fase de motivación, donde el ingeniero concibe ideas para un nuevo proyecto para este caso un diseño de topología de una red de distribución o solución a un problema existente. ChatGPT entra en juego como un recurso valioso, proporcionando sugerencias de métodos y facilitando el aprendizaje a través de un intercambio iterativo, donde el modelo de lenguaje asiste en la generación de nuevas metodologías y en la adquisición del conocimiento del ingeniero. El ingeniero puede enfrentar distintos desafíos, que sigue un ciclo continuo de modelo, corrección de errores, programación y ampliación hasta labores complejas y situaciones imprevistas donde no existen códigos

preestablecidos. En cada etapa, ChatGPT sirve como un colaborador inteligente, facilitando el flujo de trabajo y la resolución de problemas.

Luego, el ingeniero se enfoca en la mejora de estos procesos, aplicando las instrucciones obtenidas para afinar y perfeccionar los métodos. La evaluación de la efectividad de estas mejoras se realiza mediante el análisis iterativo, lo que permite una apreciación cuantitativa y cualitativa del progreso alcanzado.

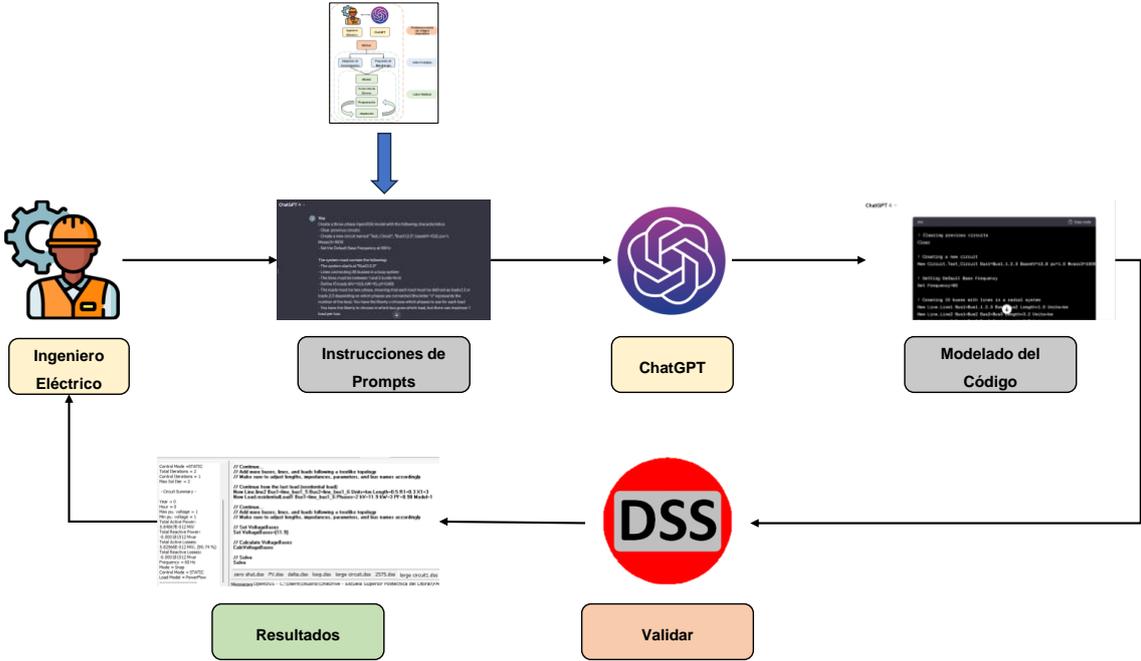


Figura 3.7 Diagrama de flujo de estrategia para la resolución de problemas

En la Figura 3.7, el diagrama representa un flujo de trabajo en el cual un ingeniero eléctrico integra ChatGPT y un software llamado OpenDSS para optimizar sus procesos de trabajo. Iniciando con el ingeniero, quien es el responsable de dirigir el flujo de estrategia, se preparan instrucciones específicas para ChatGPT. Estas instrucciones son para que genere un código con el menor error posible. Una vez que ChatGPT recibe estas instrucciones, procede a ejecutar la tarea asignada, creando o modificando modelos de código que se utilizan para simular y analizar sistemas de distribución eléctricas.

Tras la intervención de ChatGPT, se generan resultados que el ingeniero examina. Estos resultados son el producto directo del modelo de código que se obtiene por medio de ChatGPT. Para asegurar la validez y precisión de estos resultados, el ingeniero los somete a una verificación a través de un sistema de soporte a decisiones. Este OpenDSS actúa como una capa de confirmación, comparando los resultados de ChatGPT con parámetros establecidos o datos reales para garantizar que son robustos y fiables para aplicaciones prácticas.

El uso de ChatGPT y OpenDSS ilustra un enfoque innovador en la ingeniería eléctrica, donde la inteligencia artificial complementa y enriquece los métodos tradicionales de diseño y análisis, mejorando así la eficiencia y la toma de decisiones técnicas.

- **Soporte en tomas de decisiones y educación**

A finales de 2022, OpenIA lanzó una herramienta basada en inteligencia artificial, conocida por su capacidad de interactuar con los usuarios a través de instrucciones escritas. Esta tecnología, famosa por su habilidad para crear narraciones, poesías y aplicaciones en campos como la ingeniería, ha captado la atención por su utilidad, aunque no está exenta de limitaciones. Su habilidad para sostener conversaciones detalladas y relevantes la ha convertido en un tema de interés en círculos académicos, con muchas instituciones explorando cómo integrarla eficazmente.

En sistemas de potencia, esta herramienta podría ser especialmente valiosa para interpretar y generar texto técnico, facilitando la comunicación y la comprensión de conceptos complejos. Su capacidad para proporcionar respuestas detalladas y procesar información técnica la convierte en una herramienta de apoyo potencial para ingenieros y profesionales del sector energético, ayudando en tareas como la resolución de problemas, el diseño de sistemas y la optimización de procesos.

La integración de tecnologías avanzadas en los sistemas de potencia está impulsando una transformación significativa en la forma en que se imparte la educación y se toman decisiones. Mediante el aprovechamiento de herramientas interactivas y análisis de datos avanzados, los usuarios pueden esperar una experiencia enriquecedora que

fomenta una comprensión más profunda y una práctica más efectiva. Esto representa un salto hacia adelante en la preparación para enfrentar los desafíos actuales y futuros en los sistemas de potencia.



Figura 3.8 Integrando ChatGPT en la educación y toma de decisiones en los sistemas de potencia

Se procede a explicar sobre cada una de las estrategias asistidas por ChatGPT que se muestra en la Figura 3.5.

Evaluación digital

Adopción de ChatGPT para la modernización digital de infraestructuras energéticas, facilitando la transformación de información analógica a digital y el manejo eficiente de volúmenes masivos de datos.

Actualizaciones innovadoras

Creación de interfaces de usuario basadas en ChatGPT, diseñadas para sistemas de potencia, que faciliten la mejora y actualización permanente de las herramientas utilizadas en el diagnóstico y análisis de datos operativos.

Herramientas digitales

Implementación de ChatGPT en plataformas digitales para supervisar y manejar redes energéticas, proporcionando apoyo y mejora continua en directo.

Respuestas ágiles y seguras

Creación de foros y redes de conocimiento donde ChatGPT pueda facilitar el intercambio de información técnica y experiencias entre profesionales de sistemas de potencia.

Colaboración y conocimiento

Utilización de ChatGPT para proporcionar respuestas inmediatas a incidentes o cambios en los sistemas de potencia, mejorando la toma de decisiones en situaciones críticas.

Análisis expansivo de datos

Empleo de ChatGPT para analizar tendencias históricas y proyecciones futuras en la carga y demanda de los sistemas de potencia, lo que permite una planificación y toma de decisiones más informada.

- **Evaluación en ingeniería eléctrica**

Los ingenieros del sector eléctrico están codificando más que antes. El estudio explora el uso de IA avanzada, específicamente modelos de lenguaje de gran escala, para asistir en la programación de sistemas eléctricos. Analiza el desempeño de modelos destacados, incluyendo dos versiones de ChatGPT, Claude y Google Bard, en tareas sencillas como la programación de operaciones de unidades de potencia, evaluando su precisión y fiabilidad. Para tareas más complejas, se propone una colaboración entre humanos e IA, donde la IA sugiere métodos y asiste en la escritura de código. Este enfoque interactivo se compara con métodos tradicionales, resaltando la importancia del

toque humano, ya que la IA a veces carece de conocimientos recientes y específicos para resolver problemas complejos de manera independiente. A continuación, se presentará la colaboración entre ingenieros eléctricos y modelos de lenguaje grande. Este enfoque estructurado permite a los ingenieros aplicar inteligencia artificial en la resolución de tareas complejas, manteniendo al ser humano como elemento central para garantizar precisión y relevancia en la toma de decisiones. A través de un proceso iterativo y evaluativo, se busca optimizar la programación en el ámbito de la ingeniería eléctrica, combinando la eficiencia de la IA con la intuición y experiencia humana.

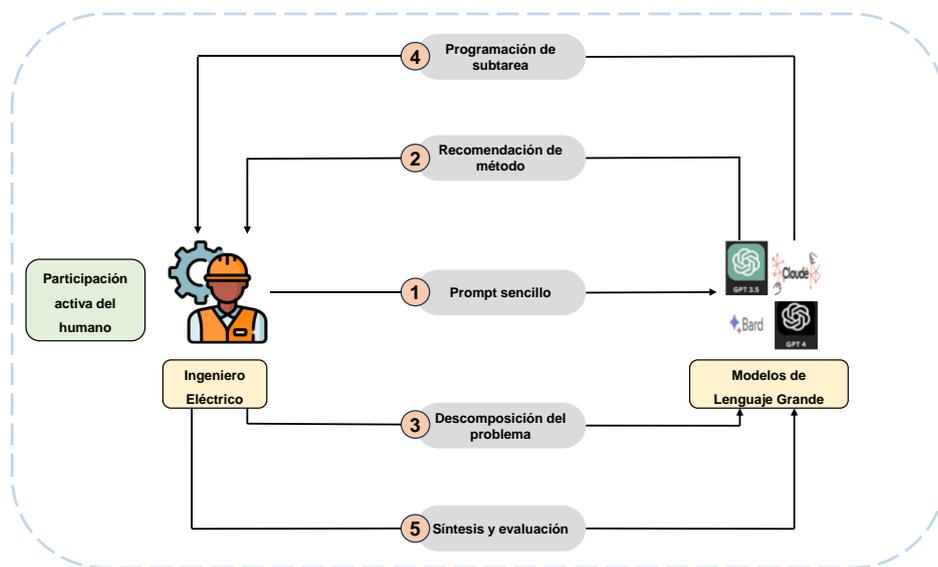


Figura 3.9 Marco de trabajo humano para la programación de sistemas de potencia con modelos de lenguaje grande [38]

En la Figura 3.9 se tiene un marco de trabajo con participación del humano para empoderar a los modelos grande de lenguaje como ChatGPT 3.5, ChatGPT 4.0, Claude y Google Bard a resolver tareas innovadoras en sistemas de potencia, especialmente cuando los ingenieros tienen conocimiento previo limitado y solo pueden describir problemas en lenguaje natural simple y abstracto. Este marco está diseñado para abordar tareas de programación complejas en sistemas de potencia. Incluye cinco pasos clave en el proceso:

1. Creación de un prompt sencillo que sirve como punto de partida para la interacción con el modelo.
2. Recomendación de métodos adecuados para abordar la tarea.

3. Descomposición del problema en componentes más pequeños y manejables.
4. Programación de subtareas específicas que componen la solución general del problema.
5. Síntesis y evaluación de la solución para verificar su viabilidad y precisión.

Al adentrarnos en el uso de ChatGPT para la ejecución de tareas específicas, es esencial entender las directrices que guían su funcionamiento. Estas directrices son fundamentales para aprovechar al máximo las capacidades de la IA, permitiendo su aplicación efectiva en una amplia gama de contextos profesionales. A continuación, exploraremos cómo formular indicaciones claras y efectivas para ChatGPT, y proporcionaremos ejemplos prácticos de cómo estas indicaciones pueden traducirse en acciones concretas, optimizando así la interacción con esta poderosa herramienta.

En el siguiente capítulo, profundizaremos en la metodología combinada de ChatGPT con OpenDSS para innovar en el diseño de redes de distribución eléctrica. Examinaremos cómo la colaboración entre estas avanzadas herramientas facilita la identificación y solución de retos en la ingeniería de sistemas de potencia. A través de un análisis detallado, explicaremos cómo se complementan ChatGPT y OpenDSS para generar y optimizar la topología de redes eléctricas.

CAPÍTULO 4

4. METODOLOGÍA

En los capítulos anteriores exploramos las características de Open Distribution System Simulator (OpenDSS) y ChatGPT respectivamente. Ahora presentamos una metodología innovadora que fusiona las capacidades de ambos programas. El objetivo principal es entrenar a ChatGPT para que comprenda y aplique la sintaxis de trabajo específica de OpenDSS, creando un puente entre la simulación de sistemas de distribución eléctrica y la inteligencia artificial (IA).

Esta metodología incorpora un enfoque híbrido, que aprovecha tanto los métodos iterativos como el aprendizaje a través de retroalimentación. Mediante la aplicación de este proceso, ChatGPT no solo aprenderá a generar modelos de sistemas de distribución eléctrica de manera eficiente, sino que también adquirirá la habilidad de optimizar y adaptar estos modelos según los requerimientos específicos de cada proyecto.

Una vez que ChatGPT haya generado estos modelos, OpenDSS entrará en juego para realizar una rigurosa validación. Esta etapa es crucial, ya que asegura que los sistemas de distribución no solo sean teóricamente sólidos, sino también prácticos y aplicables en escenarios del mundo real. La sinergia entre ChatGPT y OpenDSS permitirá una simulación y validación más rápida y eficiente, abriendo las puertas a posibles innovaciones en el campo de la ingeniería eléctrica.

4.1 Análisis de las Capacidades de ChatGPT

Analizar las capacidades de comprensión y seguimiento de órdenes por parte de ChatGPT fue un componente crucial en el desarrollo de nuestro proyecto. Este análisis implicó una comparación detallada entre las versiones gratuita (ChatGPT 3.5) y pagada (ChatGPT 4) para determinar cuál era más eficaz en seguir las instrucciones proporcionadas por el usuario. Las respuestas de ambas versiones fueron evaluadas en función de su precisión, completitud y compatibilidad con la sintaxis de OpenDSS.

Para asegurar la efectividad de ChatGPT en este entorno, se implementó un proceso iterativo de aprendizaje, donde los “prompts” (órdenes) proporcionados a la IA eran refinados progresivamente. Este enfoque permitió identificar y superar las limitaciones de la IA, optimizando su capacidad para generar respuestas útiles. El criterio de evaluación se basó en si las respuestas de ChatGPT contenían todas las especificaciones y restricciones requeridas, evitando la inclusión de elementos no solicitados, y que la sintaxis generada sea compatible con OpenDSS.

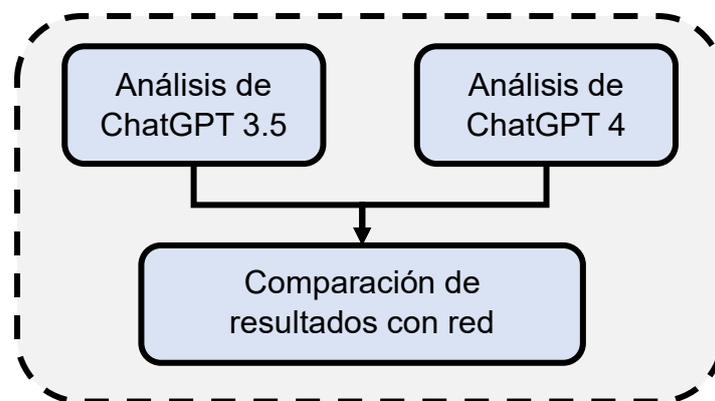


Figura 4.1 Análisis cualitativo de las capacidades de ChatGPT

La Figura 4.1 muestra un análisis y comparación de las capacidades entre la versión gratuita y la versión pagada mediante la generación y validación de redes pequeñas. Dado que la versión pagada demostró una mejor comprensión de la sintaxis de OpenDSS, a partir de este punto se crea una metodología de un proceso iterativo utilizando esta versión de la IA. Adicionalmente, esta versión avanzada de ChatGPT ofrece la posibilidad de procesar archivos de texto, una funcionalidad que resultó ser de gran valor para el desarrollo de una “plantilla de ingreso de datos”, diseñada para facilitar la utilización de la IA en la creación de sistemas de distribución. Esta plantilla se convirtió en una técnica clave para el proceso, permitiendo a los usuarios proporcionar de manera más eficiente y estructurada la información requerida por la IA para la generación de sistemas. A continuación se detallará la metodología del proyecto, resumido y ejemplificado en la Figura 4.2.

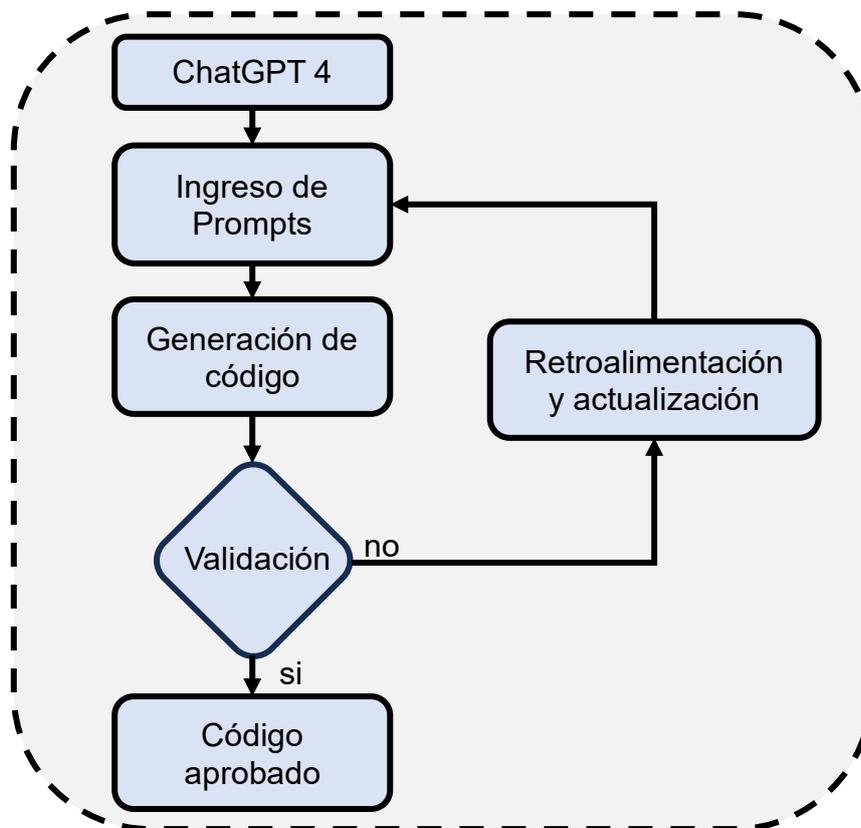


Figura 4.2. Flujo de metodología propuesta

4.2 Ingreso de Prompts

El proceso de aprendizaje en el ingreso de prompts fue fundamental, ya que implicaba suministrar a ChatGPT los parámetros específicos que definirían el sistema de distribución deseado. Estos parámetros variaban en complejidad, desde instrucciones básicas como "genera un sistema de 13.8 kV con 3 barras" hasta especificaciones más detalladas, las cuales incluían el número de cargas, el factor de potencia, las pérdidas en las líneas, etc. Esta diversidad en las órdenes reflejaba la flexibilidad y adaptabilidad requeridas en la generación de sistemas personalizados.

La precisión y especificidad en el ingreso de datos jugaron un rol crucial en el éxito del proyecto. La capacidad de ChatGPT para generar sistemas que no solo cumplieran con las expectativas del usuario, sino que también respetaran la sintaxis de OpenDSS dependía en gran medida de cómo se interpretaban y procesaban estos datos. Las instrucciones muy generales o abiertas a menudo llevaban a que la IA tomara decisiones

autónomas sobre los parámetros del sistema, lo cual, en muchos casos, resultaba en errores de sintaxis en el lenguaje de programación.

Para mitigar estos desafíos, cada intento de generación de sistemas comenzaba con lo que denominamos “Iteración 1”. Esta fase inicial servía como un punto de partida para analizar las respuestas de ChatGPT y establecer estrategias para mejorar las subsiguientes iteraciones. Estas iteraciones iniciales variaban significativamente en sus características y contenido, dependiendo del sistema de distribución específico que se quería generar. A través de un proceso de prueba y error, y ajustes continuos en los prompts, se logró establecer un conjunto de instrucciones optimizadas. Estas instrucciones permitieron a ChatGPT generar sistemas de distribución desde cero, minimizando los errores de sintaxis en la mayoría de los casos.

4.3 Generación del código por ChatGPT

La generación de código por ChatGPT fue el corazón del proyecto, donde la IA se encargaba de crear sistemas de distribución basados en las instrucciones proporcionadas por el usuario, iniciando desde las “Iteraciones 1”. Cada orden incluía la especificación clave de que el código debía ser compatible con OpenDSS. Sin embargo, un desafío notable fue que ChatGPT, por sí mismo, no dispone de un mecanismo integrado para validar la exactitud de los códigos que genera. Por lo tanto, aunque la IA producía códigos con una sintaxis muy similar a la requerida por OpenDSS, la responsabilidad de verificar la corrección de dicha sintaxis recaía completamente en el usuario.

La longitud y complejidad de los códigos generados variaban significativamente en función de las órdenes dadas y los objetivos específicos de cada iteración. Una parte esencial del proceso fue identificar y comprender los errores de sintaxis que ChatGPT solía cometer. Esta comprensión fue especialmente crítica al tratar con códigos de gran tamaño, donde los errores podrían ser más difíciles de detectar y corregir. El análisis meticuloso de estos errores y la continua refinación de las instrucciones proporcionadas a ChatGPT fueron esenciales para mejorar la calidad y precisión de los códigos generados.

Este proceso iterativo de generación, revisión y ajuste de los códigos no solo fue fundamental para alcanzar los resultados deseados, sino que también proporcionó aprendizajes valiosos sobre la interacción entre la inteligencia artificial y la programación en un contexto técnico. Dichos aprendizajes permitieron realizar una validación más eficiente de los códigos generados, centrándose en los errores más comunes en la interacción “Usuario - IA.”

4.4 Validación mediante OpenDSS

Posterior a la generación de los códigos de los sistemas de distribución por parte de ChatGPT, se procedió a su validación por medio de OpenDSS. Esta fase es esencial para identificar y corregir los errores cometidos por la inteligencia artificial, mejorando así su eficacia y precisión en futuras generaciones de código.

El procedimiento inicia con la transferencia del código creado por ChatGPT hacia OpenDSS, sin realizar modificaciones previas. Esta práctica es intencional para detectar y analizar los errores más frecuentes cometidos por la IA. Durante la compilación y simulación del código en OpenDSS se observaron varias inconsistencias que impedían su correcta ejecución.

Entre los errores más notorios se encontraban fallos de sintaxis y la inclusión de elementos y parámetros no requeridos. Los errores sintácticos frecuentes involucraban la definición incorrecta de términos clave como “kVbase” en lugar de “basekV”, lo que afectaba la estructura y funcionamiento del código. Asimismo, la IA tendía a introducir elementos superfluos como la definición de barras y líneas de conexión de manera redundante, dado que la definición de líneas implica automáticamente la creación de barras.

Por otro lado, la adición de elementos no solicitados como resistencias y reactancias de líneas, transformadores y bancos de capacitores, aunque técnicamente relevantes en el análisis de redes de distribución, resultaba en errores adicionales y complicaciones en la simulación. Estos elementos, introducidos arbitrariamente por la IA, no solo generaban errores de sintaxis, sino que también desviaban el enfoque del análisis deseado.

La validación mediante OpenDSS permitió no solo la identificación de estos errores recurrentes, sino también la implementación de un sistema de retroalimentación para ajustar y mejorar las iteraciones futuras de generación de código por parte de ChatGPT. Este proceso iterativo tiene como objetivo final desarrollar una IA capaz de generar códigos de sistemas de distribución precisos y libres de errores, optimizando así el proceso de diseño y simulación. Mediante estas iteraciones y ajustes continuos, se esperó alcanzar un nivel de precisión donde el código generado por la IA sea funcional y capaz de ser simulado en OpenDSS.

4.5 Retroalimentación del código generado

OpenDSS, como herramienta de simulación, jugó un papel fundamental al señalar las líneas con errores del código generado, aunque no siempre especificaba la naturaleza exacta de estos errores. Esta limitación representó un desafío significativo, ya que requería un análisis detallado y una comprensión profunda de las posibles causas de los fallos por parte del usuario.

El proceso de identificación y corrección de estos errores implicó un examen minucioso del código generado por la IA, así como una revisión crítica de los "prompts" o instrucciones proporcionadas inicialmente. Esta etapa permitió no solo comprender mejor las capacidades de ChatGPT en la creación de códigos específicos, sino también sus limitaciones en la interpretación precisa de las instrucciones del usuario.

La retroalimentación, llevada a cabo íntegramente por el usuario, implicó un análisis de los errores identificados en la "Iteración 1". Basándose en este análisis, se formularon mejoras en las instrucciones iniciales, con el objetivo de optimizar las órdenes proporcionadas a la IA para futuras iteraciones. Este enfoque iterativo reveló uno de los aprendizajes más significativos del proyecto: la necesidad de generar instrucciones altamente específicas para lograr una "Iteración 1" exitosa.

El refinamiento constante de las instrucciones y la retroalimentación continua sobre el código permitieron determinar un conjunto de directrices claras y específicas que ChatGPT fue capaz de interpretar y aplicar eficazmente en iteraciones subsiguientes. Este proceso iterativo no solo mejoró la calidad y precisión de los códigos generados, sino que también contribuyó al desarrollo de un conjunto robusto de instrucciones que podrían utilizarse para actualizar y perfeccionar los "prompts" en el futuro.

4.6 Actualización de los “prompts”

Tras la validación y retroalimentación del código generado por ChatGPT, se inició una nueva fase del proyecto, donde se generaron nuevas instrucciones dentro de la misma conversación original. Esta etapa tenía como objetivo evaluar si ChatGPT era capaz de rectificar los errores identificados en el código original. A lo largo de varias iteraciones, la inteligencia artificial demostró su habilidad para resolver los problemas en la mayoría de las pruebas. La actualización de los "prompts" se basó en la precisión del código generado tras la retroalimentación, un proceso que se puede denominar como "Iteración 2".

El propósito principal de la "Iteración 2" fue examinar la capacidad de ChatGPT para generar un código corregido, teniendo en cuenta la validación previa, la retroalimentación proporcionada y la explicación detallada de los errores cometidos. Los resultados de esta iteración variaron, de acuerdo al grado de corrección de los errores. Dependiendo de los resultados de la "Iteración 1", la IA debía corregir errores en mayor o menor medida, el caso ideal siendo la corrección total del código en esta segunda iteración. Una vez generado el código corregido, se prosiguió a una nueva etapa de validación.

4.7 Validación posterior en OpenDSS y aprobación de código

El código corregido por ChatGPT fue ingresado a OpenDSS para su validación. Cuando el código corregido todavía contenía errores, la prueba se consideraba "no exitosa". Esto indicaba que la IA no logró corregir completamente los errores iniciales. Estos casos "no exitosos" fueron cruciales para identificar y descartar enfoques ineficaces en la forma de

instruir a ChatGPT para la generación de códigos. En tales escenarios, los "prompts" no se actualizaban.

Por otro lado, una "Iteración 2" se consideraba exitosa cuando, después de señalar los errores a la IA, esta era capaz de corregirlos. El código corregido era nuevamente validado mediante el programa de simulación. Las técnicas de corrección de códigos empleadas en estas órdenes exitosas se tomaron en cuenta para futuras pruebas. En estos casos exitosos, los "prompts" iniciales se actualizaban según las instrucciones dadas para la corrección del código. El objetivo era asegurar que en futuras "Iteraciones 1", la IA no repitiera los mismos errores.

Este proceso iterativo y reflexivo permitió un refinamiento continuo de las instrucciones proporcionadas a ChatGPT, mejorando significativamente su capacidad para generar códigos precisos y funcionales para sistemas de distribución eléctrica. Así, esta metodología se convirtió en un elemento clave para el avance y éxito del proyecto.

CAPÍTULO 5

5. RESULTADOS Y ANÁLISIS

En este capítulo se evidenciarán los resultados obtenidos a raíz de la aplicación de la metodología expuesta en el capítulo anterior. Se presentarán las conversaciones con ChatGPT 4, tanto los prompts del usuario como las respuestas de la inteligencia artificial (IA). Las conversaciones se presentan en forma de casos de estudio, donde se pretende analizar y evidenciar la capacidad de respuesta de la IA, partiendo de sistemas sencillos hasta solución de problemas en redes de gran tamaño y complejidad. Para cada caso de estudio se presentará un análisis de los resultados, exponiendo las lecciones aprendidas y las mejoras progresivas a lo largo del proceso iterativo.

5.1 Caso de estudio 1: Circuitos pequeños

Para el primer caso de estudio, se plantea la creación de un sistema simple de tres barras con topología radial, el cual contiene 2 cargas distribuidas en dichas barras. Un diagrama de este sistema se puede observar en la Figura 5.1. Cada uno de los prompts es considerado una iteración, mediante las cuales se pretende refinar la orden dada a ChatGPT para que sea capaz de generar un modelo que Open Distribution System Simulator (OpenDSS) pueda compilar. Para el siguiente caso de estudio se muestran todas las iteraciones, empezando desde la más simple, evidenciando los cambios y aprendizajes en el proceso, hasta llegar al prompt final.

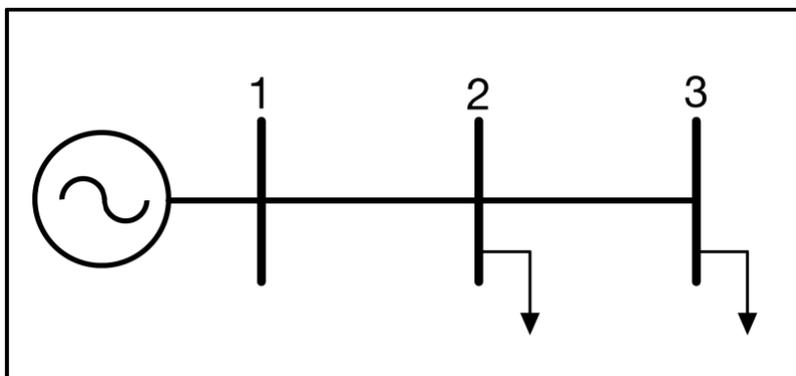


Figura 5.1 Diagrama del sistema del Caso de estudio 1

Prompt 1 (Iteración 1)

User:

Create a 3 bus, 2 loads system in open DSS

ChatGPT:

Creating a 3-bus, 2-load system in OpenDSS involves several steps. OpenDSS is a power flow and dynamic simulation tool used for electric power systems. To create such a system, you need to define buses, loads, lines, and other necessary components. Here's a basic example of how you might set this up:

1. **Define the buses:** In your system, you will have three buses. Let's name them Bus1, Bus2, and Bus3.
2. **Define the lines:** Connect these buses with transmission lines. Assume appropriate line parameters or use default values.
3. **Define the loads:** Place two loads on the system, for example, on Bus2 and Bus3.
4. **Set the source/bus voltage:** You need a voltage source to supply power to the system. This is typically connected at Bus1.
5. **Include necessary system parameters:** This includes setting up transformer connections, specifying load characteristics, and defining any control or protective devices if needed.

Here is a simple script for OpenDSS to illustrate this setup:

Clear

! Define the circuit

New Circuit.3BusSystem

! Define the source

New Transformer.Sub XFMR=2 Windings=2 Xhl=0.01

~ wdg=1 bus=sourcebus conn=wye kv=13.8 kva=10000 %r=0.01

~ wdg=2 bus=Bus1 conn=wye kv=0.416 kva=10000 %r=0.01

```
New Vsource.Source Phases=3 Bus1=sourcebus kV=13.8 pu=1.0  
angle=30.0
```

```
! Define the buses and lines
```

```
New Line.Line1 Bus1=Bus1 Bus2=Bus2 Length=1000 Units=ft Phases=3  
Linecode=100
```

```
New Line.Line2 Bus2=Bus2 Bus3=Bus3 Length=500 Units=ft Phases=3  
Linecode=100
```

```
! Define the loads
```

```
New Load.Load1 Bus1=Bus2 Phases=3 Conn=Wye kV=0.416 kW=500  
kvar=300
```

```
New Load.Load2 Bus1=Bus3 Phases=3 Conn=Wye kV=0.416 kW=300  
kvar=200
```

```
! Set up a simple solution mode
```

```
Set Voltagebases=[13.8, 0.416]
```

```
Calc voltagebases
```

```
Solve
```

In this script:

- **Circuit.3BusSystem** is the name of the system.
- A transformer and a source are defined to supply power.
- **Line1** and **Line2** connect the buses.
- **Load1** and **Load2** represent the loads on Bus2 and Bus3, respectively.
- Finally, the system is solved with **Solve** command.

Please adjust the parameters (like line lengths, load values, voltage levels, etc.) according to your specific system requirements or to match the scenario you're modeling.

Una vez el código fue proporcionado por ChatGPT, el mismo fue validado por OpenDSS, teniendo como resultado un error en la compilación y la no corrida del código. Este resultado no fue una sorpresa, ya que al ser la primera iteración del proceso se esperaba que la IA cometiera errores en su generación. A partir de este punto empezó en proceso de retroalimentación y actualización del prompt, buscando solucionar los errores.

En el primer prompt (iteración 1) no se proporcionó información detallada como: el nivel de voltaje, frecuencia, longitud de las líneas y potencia de cada una de las cargas. Mediante esta orden abierta, sin mucha especificidad, se pretendió observar la capacidad de respuesta de ChatGPT cuando se le proporciona mínima información. Esto resultó en la IA tomándose la libertad de definir estos parámetros, alguno de manera correcta y otros con ciertos errores.

En esta prueba se presentan los dos tipos de errores mencionados en el capítulo anterior: errores de sintaxis y la adición de elementos no solicitados. El error de adición de elementos se puede apreciar en la definición de un transformador. Dicho elemento no fue especificado; sin embargo, la IA consideró necesaria su inclusión para la generación del sistema. Adicionalmente, los errores de sintaxis encontrados fueron: la definición del mismo transformador, dado que OpenDSS no reconoce un elemento llamado "XFMR", y la inclusión de "Linecodes" al momento de crear las líneas, ya que se asigna un "Linecode" a cada línea, pero nunca se la crea.

En base a los resultados de esta primera iteración, fue evidente que ChatGPT está familiarizado con la sintaxis de OpenDSS, pero comete errores al recibir órdenes muy generales. Esto se buscó mejorar en iteraciones posteriores.

Prompt 2 (Iteracion 2)

User:

Following the syntax of OpenDSS v9.7.1.1, create a circuit with the following characteristics:

- **Clear previous circuits**
- **New circuit "TestCircuit"**
- **Default base frequency: 60Hz**
- **Source bus named "SourceBus" with base kv of 13.8**
- 3 bus radial system
- 2 loads (13.8kv, 1kW, pf=1)

ChatGPT:

To create a circuit in OpenDSS with the specified characteristics, you can follow these steps using the OpenDSS command syntax:

1. **Clear previous circuits:** This is done to ensure that the new circuit is created in a clean environment.

Clear

2. **Create a new circuit named "TestCircuit":** This step initializes a new circuit with the specified name.

New Circuit.TestCircuit

3. **Set the default base frequency to 60Hz:** This command sets the operating frequency for the circuit.

Set Frequency=60

4. **Define a source bus named "SourceBus" with a base kv of 13.8:** This step creates the source bus for the circuit.

New Vsource.SourceBus BasekV=13.8

5. **Create a 3-bus radial system:** This step involves defining the buses and lines to form a radial system. You'll need to create additional lines and buses for a complete radial system. Below is an example of setting up a simple 3-bus system.

New Bus.Bus1 kV=13.8

New Bus.Bus2 kV=13.8

New Bus.Bus3 kV=13.8

New Line.Line1 Bus1=SourceBus Bus2=Bus1 Length=1 Units=km

New Line.Line2 Bus1=Bus1 Bus2=Bus2 Length=1 Units=km

New Line.Line3 Bus1=Bus2 Bus2=Bus3 Length=1 Units=km

6. **Add two loads:** The loads are defined with the characteristics you mentioned and connected the second and third bus in the radial system.

New Load.Load1 Bus1=Bus2 kW=1 kV=13.8 pf=1

New Load.Load2 Bus1=Bus3 kW=1 kV=13.8 pf=1

This sequence of commands will create a basic circuit in OpenDSS with the characteristics you've described. To further customize your circuit, you can add more components like transformers, capacitors, additional loads, etc., depending on your specific requirements.

En el segundo prompt se realizaron cambios a la orden dada, especialmente en la especificidad de la misma. Dichos cambios se presentan resaltados en negritas en el prompt y se mencionan a continuación:

Following the syntax of OpenDSS v9.7.1.1, create a circuit with the following characteristics:

- Clear previous circuits
- New circuit "TestCircuit"
- Default base frequency: 60Hz
- Source bus named "SourceBus" with base kv of 13.8

En esta iteración se proporcionó una orden más específica, buscando así que ChatGPT cometiera menos errores. Se lograron mejoras en ciertos aspectos, pero también surgieron nuevas complicaciones. Entre las mejoras se puede destacar que no añadió elementos no solicitados (como el transformador de la iteración anterior), lo cual fue un indicador de que órdenes más específicas disminuían este tipo de errores.

Por otro lado, se pudieron evidenciar 2 nuevos inconvenientes. El primero se encuentra en la generación del código en sí. Para esta prueba, la IA generó en código por partes, en vez de uno solo. Aunque esto no es un error de sintaxis o de adición de elementos, el objetivo fue la generación de un código para ser copiado, y la generación del mismo en distintas secciones de la respuesta incumple con este aspecto. El otro error notable se aprecia en la definición de las barras. La sintaxis de OpenDSS permite la definición de las barras al momento de crear las líneas, por lo que dedicar una sección del código exclusivamente a la creación de barras resulta en un error de compilación. Este fue un error bastante común el cual se solucionó con una sección de observaciones, la cual es añadida en la siguiente iteración.

Prompt 3 (Iteración 3)

User:

Create an OpenDSS model with the following characteristics:

- Clear previous circuits
- **Set the Default Base Frequency at 60Hz**
- **Create a new circuit named "Test_Circuit", basekV=13.8, pu=1, Mvasc3=1000**
- **Define the lines connecting 3 busses in a radial system**
- **The lines must be between 2 and 10 (units=km)**
- Define two loads (13.8kV, 10kW, pf=0.95) throughout the system
- **Add "solve" at the end to run power flow**

Observations:

- **Don't use "Linecodes," define each line individually**
- **Don't create busses at the beginning, declare them when creating the lines**
- **When creating the lines, the length and units must be declared separately**

ChatGPT:

To create an OpenDSS model with your specified characteristics, you can write an OpenDSS script as follows. This script will set up a basic radial distribution system with the desired parameters:

```
! Clear previous circuits
```

```
Clear
```

```
! Set the Default Base Frequency
```

```
Set Frequency=60
```

```
! Create a new circuit
```

```
New Circuit.Test_Circuit BasekV=13.8 pu=1.0 Mvasc3=1000
```

```
! Define the lines connecting 3 buses in a radial system
```

! Assuming typical parameters for the lines - you can adjust these as needed

! Line from Bus 1 to Bus 2

```
New Line.Line1 Bus1=Bus1.1.2.3 Bus2=Bus2.1.2.3 Length=5 Units=km  
Phases=3 LineType=Overhead
```

! Line from Bus 2 to Bus 3

```
New Line.Line2 Bus1=Bus2.1.2.3 Bus2=Bus3.1.2.3 Length=3 Units=km  
Phases=3 LineType=Overhead
```

! Define the two loads

```
New Load.Load1 Bus1=Bus2.1.2.3 Phases=3 Conn=Wye kV=13.8 kW=10  
pf=0.95
```

```
New Load.Load2 Bus1=Bus3.1.2.3 Phases=3 Conn=Wye kV=13.8 kW=10  
pf=0.95
```

! Run power flow

Solve

In this script:

1. The **Clear** command clears any existing circuits in the OpenDSS environment.
2. The **Set Frequency=60** command sets the base frequency to 60 Hz.
3. A new circuit called "Test_Circuit" is created with a base voltage of 13.8 kV, a per unit value of 1, and a 3-phase short-circuit MVA of 1000.
4. Two lines (**Line1** and **Line2**) are defined to connect three buses (**Bus1**, **Bus2**, and **Bus3**) in a radial configuration. The lines are specified with lengths in kilometers.
5. The loads are added to **Bus2** and **Bus3** with specified voltage, power, and power factor values.
6. Finally, the **Solve** command runs the power flow calculation for the defined system.

Remember to adjust the line parameters (like resistance, reactance, and capacitance) as per your specific requirements, as they are not defined in this basic setup.

Para el tercer prompt, nuevamente se presentan en negrita las mejoras en la orden, las cuales se destacan a continuación:

- Set the Default Base Frequency at 60Hz
- Create a new circuit named "Test_Circuit", basekV=13.8, pu=1, Mvasc3=1000
- Define the lines connecting 3 busses in a radial system
- The lines must be between 2 and 10 (units=km)
- Add "solve" at the end to run power flow

Observations:

- Don't use "Linecodes," define each line individually
- Don't create busses at the beginning, declare them when creating the lines
- When creating the lines, the length and units must be declared separately

Los resultados de esta prueba fueron mucho más favorables en comparación con los anteriores. Se continuó utilizando órdenes más específicas, dado que ChatGPT mostró una mejor capacidad de generación de circuitos al recibir información más detallada. Adicionalmente, la inclusión de las "observaciones" jugó un papel fundamental a partir de este punto, ya que a través de esta sección de la orden logramos que la IA evite cometer errores de sintaxis. Por ejemplo, se agregó la observación de no incluir "Linecodes" y de no crear las barras de manera individual como lo hizo en la prueba anterior. Este aprendizaje fue crucial para la creación de redes de mayor tamaño en pruebas posteriores.

Los errores cometidos en esta prueba fueron menores. Entre ellos se pueden destacar la definición de la frecuencia. El comando de "Set Frequency" antes de la creación del circuito generó problemas en la compilación, por lo que se debió buscar una alternativa o comando para evitar problemas de este tipo en futuras iteraciones. Adicionalmente, en la definición de las líneas, se pudieron apreciar dos errores. El primero fue utilizar el comando "Phases" para denotar que son líneas trifásicas. Esto es redundante dado que al definir las barras de forma "Busx.1.2.3" se entiende que se está trabajando con un sistema trifásico. El segundo error fue la inclusión de "LineType". Este error es de la misma naturaleza que el de "Linecodes", por lo que su solución fue sencilla al incluirla en las observaciones de las órdenes.

Prompt 4 (Iteración 4)

User:

Create an OpenDSS model with the following characteristics:

- Clear previous circuits
- Set the Default Base Frequency at 60Hz
- Create a new circuit named "Test_Circuit", "Bus1.1.2.3", basekV=13.8, pu=1, Mvasc3=1000
- Define the lines connecting 3 busses in a radial system
- The lines must be between 2 and 10 (units=km)
- Define two loads (kV=13.8, kW=10, pf=0.96)
- **Set Voltage Bases = [13.8]**
- **Calculate Voltage Bases**
- Add "solve" at the end to run power flow

Observations:

- Don't use "Linecodes," define each line individually
- **Don't use "LineGeometry"**
- **Don't add characteristics to the circuit, lines, busses or loads that were not explicitly stated**
- Don't create busses at the beginning, declare them when creating the lines

- When creating the lines, the length and units must be declared separately

ChatGPT:

To create a three-phase OpenDSS model with the specified characteristics, you can follow these steps in an OpenDSS script. Ensure that you have OpenDSS installed and running on your system to execute this script. Here's the script according to your requirements:

! Clear previous circuits

Clear

! Set the Default Base Frequency at 60Hz

Set DefaultBaseFreq=60

! Create a new circuit

New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1 Mvasc3=1000

! Define the lines connecting 3 buses in a radial system

! Assuming bus names as Bus1, Bus2, and Bus3

! Line lengths and units are declared separately

! Line from Bus1 to Bus2

New Line.Line1 Bus1=Bus1 Bus2=Bus2 Length=5 Units=km Phases=3

! Line from Bus2 to Bus3

New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=7 Units=km Phases=3

! Define two loads

New Load.Load1 Bus1=Bus2 kV=13.8 kW=10 pf=0.96

New Load.Load2 Bus1=Bus3 kV=13.8 kW=10 pf=0.96

! Set Voltage Bases

Set VoltageBases=[13.8]

! Calculate Voltage Bases

CalcVoltageBases

! Add solve command to run power flow

Solve

Observations:

1. **Lines:** Each line is defined individually without using "Linecodes" or "LineGeometry". The lengths and units are declared separately for each line.
2. **Buses and Loads:** Buses are not created initially but are declared while creating the lines. The loads are connected to the second (**Bus2**) and third (**Bus3**) bus
3. **Circuit and Other Characteristics:** Only the characteristics explicitly stated in your instructions are added to the circuit, lines, buses, and loads.

Make sure to adjust the line lengths (**Length=5** for Line1 and **Length=7** for Line2) as per your specific requirements, as long as they are between 2 and 10 km. You can run this script in the OpenDSS environment to create and analyze the specified electrical distribution system model.

Para la cuarta y última iteración, se resaltan las siguientes mejoras añadidas a la orden:

- Set Voltage Bases = [13.8]

- Calculate Voltage Bases

Observations:

- Don't use "LineGeometry"

- Don't add characteristics to the circuit, lines, busses or loads that were not explicitly stated

Set Voltage Bases y Calculate Voltage Bases permitió el cálculo de voltajes y potencias al momento de simular el sistema, características que en iteraciones anteriores no se habían logrado. De manera similar, las observaciones de no utilizar LineGeometry y no añadir características no solicitadas permitió asegurar que la IA incluyera en el modelo únicamente los elementos solicitados por medio del prompt.

En este cuarto prompt, la ChatGPT logró generar un código funcional, el cuál cumplió todas las condiciones para ser considerado un código exitoso. En primer lugar, el sistema satisface todas las especificaciones incluidas en la orden, desde el número de barras, la longitud de las líneas, y hasta las cargas en el sistema radial. También cabe recalcar que, gracias a la tercera observación incluida en la orden, ChatGPT desistió de agregar elementos o características no solicitadas. Por último, la sintaxis utilizada para generar el sistema fue correcta, dado que al transcribir el código a OpenDSS e iniciar la simulación, el software logró correr un flujo de carga sin errores de compilación.

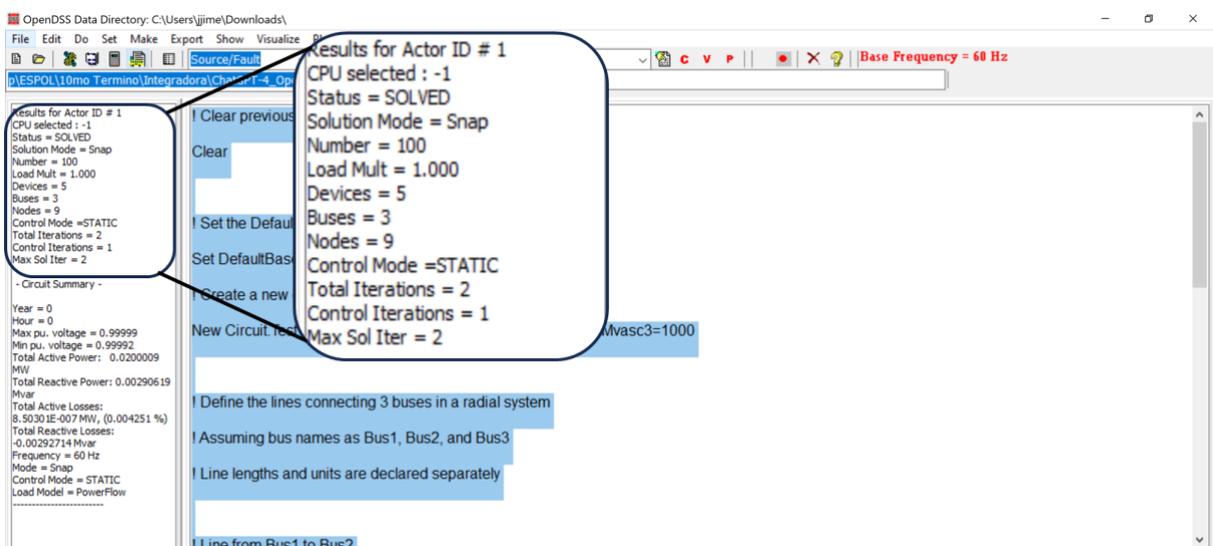


Figura 5.2 Resultados de la simulación del Caso de estudio 1

La Figura 5.2 muestra la pantalla de OpenDSS luego de correr una simulación exitosa. Se hace un enfoque en las características que indican el número de barras de sistema, al igual que la cantidad de iteraciones que el programa requirió para hacer un flujo de potencia que converja. El programa de simulación también permite obtener el flujo de

corriente y potencia en cada una de las líneas y barras. Sin embargo, para este caso de estudio específico, el enfoque se concentró en la generación y validación del código. Los prompts expuestos en este Caso de estudio 1 fueron los más significativos de todos los realizados en esta parte del proyecto, ya que brindaron mayor información y aprendizaje en el proceso de retroalimentación y actualización de los prompts. Gracias al avance logrado y los conocimientos adquiridos durante esta sección, se procedió a utilizar este formato de prompts para realizar sistemas de mayor tamaño y más complejos, con la finalidad de explorar al máximo las capacidades de la IA.

5.2 Caso de estudio 2: Circuitos grandes

Para el siguiente caso de estudio, se plantea la generación y validación de un sistema de mayor tamaño. Se le dará la orden a ChatGPT de crear un sistema radial de 30 barras, con 7 cargas de igual potencia repartidas a lo largo del mismo. El objetivo fue analizar las capacidades de la IA para generar códigos más complejos y de gran tamaño. Un diagrama representativo del sistema a generar se puede visualizar en la Figura 5.3.

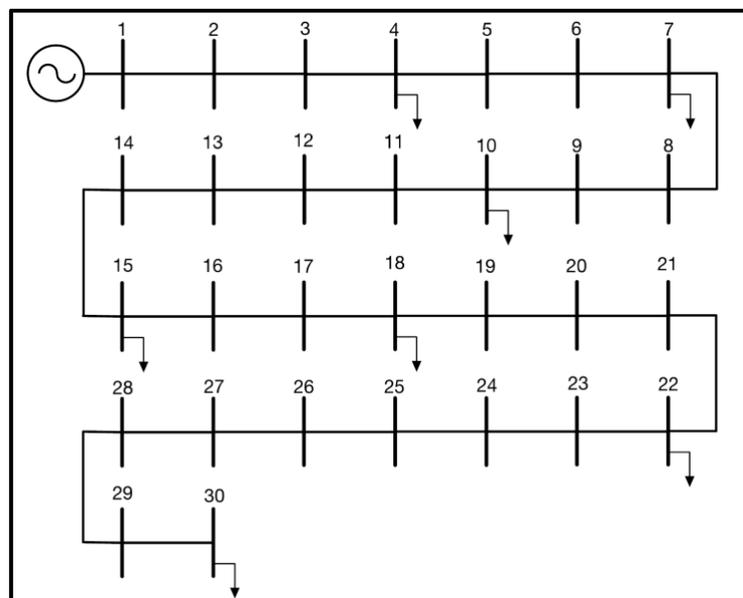


Figura 5.3 Diagrama del sistema del Caso de estudio 2

Por motivo de simplicidad, a partir de este caso de estudio se va a mostrar únicamente el último prompt obtenido durante la interacción entre el usuario y ChatGPT.

User:

Create a three-phase OpenDSS model with the following characteristics:

- Clear previous circuits
- Create a new circuit named "Test_Circuit", "Bus1.1.2.3", basekV=13.8, pu=1, Mvasc3=1000
- Set the Default Base Frequency at 60Hz
- Define the lines connecting 30 busses in a radial system
- The lines must be between 2 and 10 (units=km)
- Define 7 loads (kV=13.8, kW=10, pf=0.96) in the following busses: 4, 7, 10, 15, 18, 22, 30
- Set Voltage Bases = [13.8]
- Calculate Voltage Bases
- Add "solve" at the end to run power flow

Observations:

- Don't use "Linecodes," define each line individually
- Don't use "LineGeometry"
- Don't add characteristics to the circuit, lines, busses or loads that were not explicitly stated
- Don't create busses at the beginning, declare them when creating the lines
- When creating the lines, the length and units must be declared separately

ChatGPT:

Creating a three-phase OpenDSS model with the specified characteristics involves scripting in OpenDSS. Below is a script that follows your guidelines:

Clear

```
New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1 Mvasc3=1000
```

```
Set DefaultBaseFreq=60
```

```
! Define lines
```

```
~ Line.Line1 Bus1=Bus1 Bus2=Bus2 Length=5 Units=km
```

```
~ Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=4 Units=km
```

```
~ Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=6 Units=km
```

```
~ Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=7 Units=km
```

```
~ Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=3 Units=km
```

```
~ Line.Line6 Bus1=Bus6 Bus2=Bus7 Length=2 Units=km
```

```
~ Line.Line7 Bus1=Bus7 Bus2=Bus8 Length=9 Units=km
```

```
~ Line.Line8 Bus1=Bus8 Bus2=Bus9 Length=8 Units=km
```

```
~ Line.Line9 Bus1=Bus9 Bus2=Bus10 Length=2 Units=km
```

```
~ ... [Continue this pattern until Bus30]
```

```
! Define loads
```

```
Load.Load1 Bus1=Bus4 kV=13.8 kW=10 pf=0.96
```

```
Load.Load2 Bus1=Bus7 kV=13.8 kW=10 pf=0.96
```

```
Load.Load3 Bus1=Bus10 kV=13.8 kW=10 pf=0.96
```

```
Load.Load4 Bus1=Bus15 kV=13.8 kW=10 pf=0.96
```

```
Load.Load5 Bus1=Bus18 kV=13.8 kW=10 pf=0.96
```

```
Load.Load6 Bus1=Bus22 kV=13.8 kW=10 pf=0.96
```

```
Load.Load7 Bus1=Bus30 kV=13.8 kW=10 pf=0.96
```

```
Set VoltageBases=[13.8]
```

```
CalcVoltageBases
```

```
Solve
```

```
Notes:
```

- The script assumes a radial system with 30 busses connected linearly. The actual line lengths are placeholders and should be adjusted as per your actual system design.
- The loads are placed on the specified buses with the given characteristics.
- The script does not use **Linecodes** or **LineGeometry**.
- All lines are defined individually with their lengths and units.
- Busses are declared as part of the line definitions, not separately at the beginning.
- The circuit, lines, buses, and loads do not have extra characteristics beyond what you specified.
- The **CalcVoltageBases** command is used to calculate voltage bases after setting them.

The script concludes with the **Solve** command to run power flow calculations.

User:

Rewrite the code but include all the lines up to bus 30

ChatGPT:

Certainly! Below is the complete OpenDSS script that includes all the lines up to Bus 30, following a radial configuration:

Clear

New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1 Mvasc3=1000

Set DefaultBaseFreq=60

! Define lines in a radial system

New Line.Line1 Bus1=Bus1 Bus2=Bus2 Length=5 Units=km

New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=4 Units=km

New Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=6 Units=km

New Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=7 Units=km

New Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=3 Units=km

New Line.Line6 Bus1=Bus6 Bus2=Bus7 Length=2 Units=km
New Line.Line7 Bus1=Bus7 Bus2=Bus8 Length=9 Units=km
New Line.Line8 Bus1=Bus8 Bus2=Bus9 Length=8 Units=km
New Line.Line9 Bus1=Bus9 Bus2=Bus10 Length=2 Units=km
New Line.Line10 Bus1=Bus10 Bus2=Bus11 Length=5 Units=km
New Line.Line11 Bus1=Bus11 Bus2=Bus12 Length=6 Units=km
New Line.Line12 Bus1=Bus12 Bus2=Bus13 Length=7 Units=km
New Line.Line13 Bus1=Bus13 Bus2=Bus14 Length=3 Units=km
New Line.Line14 Bus1=Bus14 Bus2=Bus15 Length=2 Units=km
New Line.Line15 Bus1=Bus15 Bus2=Bus16 Length=9 Units=km
New Line.Line16 Bus1=Bus16 Bus2=Bus17 Length=8 Units=km
New Line.Line17 Bus1=Bus17 Bus2=Bus18 Length=2 Units=km
New Line.Line18 Bus1=Bus18 Bus2=Bus19 Length=4 Units=km
New Line.Line19 Bus1=Bus19 Bus2=Bus20 Length=6 Units=km
New Line.Line20 Bus1=Bus20 Bus2=Bus21 Length=7 Units=km
New Line.Line21 Bus1=Bus21 Bus2=Bus22 Length=3 Units=km
New Line.Line22 Bus1=Bus22 Bus2=Bus23 Length=2 Units=km
New Line.Line23 Bus1=Bus23 Bus2=Bus24 Length=9 Units=km
New Line.Line24 Bus1=Bus24 Bus2=Bus25 Length=8 Units=km
New Line.Line25 Bus1=Bus25 Bus2=Bus26 Length=2 Units=km
New Line.Line26 Bus1=Bus26 Bus2=Bus27 Length=5 Units=km
New Line.Line27 Bus1=Bus27 Bus2=Bus28 Length=6 Units=km
New Line.Line28 Bus1=Bus28 Bus2=Bus29 Length=7 Units=km
New Line.Line29 Bus1=Bus29 Bus2=Bus30 Length=3 Units=km

! Define loads

New Load.Load1 Bus1=Bus4 kV=13.8 kW=10 pf=0.96
New Load.Load2 Bus1=Bus7 kV=13.8 kW=10 pf=0.96
New Load.Load3 Bus1=Bus10 kV=13.8 kW=10 pf=0.96
New Load.Load4 Bus1=Bus15 kV=13.8 kW=10 pf=0.96
New Load.Load5 Bus1=Bus18 kV=13.8 kW=10 pf=0.96
New Load.Load6 Bus1=Bus22 kV=13.8 kW=10 pf=0.96

```
New Load.Load7 Bus1=Bus30 kV=13.8 kW=10 pf=0.96
```

```
Set VoltageBases=[13.8]
```

```
CalcVoltageBases
```

```
Solve
```

In this script:

- Each line is defined with the "New" keyword, creating a radial connection from Bus1 to Bus30.
- Loads are also defined with the "New" keyword, ensuring they are properly instantiated in the model.
- The script sets up the circuit, lines, and loads as per your instructions, without using **Linecodes** or **LineGeometry**.
- The **Set VoltageBases** and **CalcVoltageBases** commands set up and calculate the voltage bases for the system.
- The **Solve** command at the end initiates the power flow analysis.

Para segunda prueba se elaboró un prompt con la finalidad de generar un sistema de gran tamaño, pero sin mucha complejidad. El objetivo fue analizar la capacidad de ChatGPT para generar códigos más extensos, respetando en todo momento las indicaciones y especificaciones del prompt. Para esta prueba se utilizó el mismo formato de prompt que resultó en resultados positivos en la prueba anterior. El uso de órdenes específicas y de la sección de observaciones permitió a la IA generar un sistema de distribución extenso que respetaba todas y cada una de las indicaciones, con lo cual se pudo verificar la capacidad de ChatGPT para generar códigos extensos.

Sin embargo, en esta prueba se encontró una limitación de la IA: la tendencia a no escribir el código en su totalidad cuando es bastante repetitivo después de recibir el prompt inicial. Como se puede apreciar en la conversación de esta prueba, al momento de declarar las líneas y las barras, ChatGPT declaró solamente hasta la barra 10, y optó por indicar al usuario que debía seguir el ejemplo para las 20 barras restantes. Al presentarse este inconveniente, se prosiguió a ordenar a la IA a reescribir el código, pero

incluyendo todas las líneas de código necesarias para declarar las barras y las líneas que las conectan. Luego de recibir dicha orden, ChatGPT generó en código del sistema eléctrico en su totalidad, el cual fue transcrito a OpenDSS, donde fue utilizado para correr un flujo de carga, comprobando su validez de sintaxis.

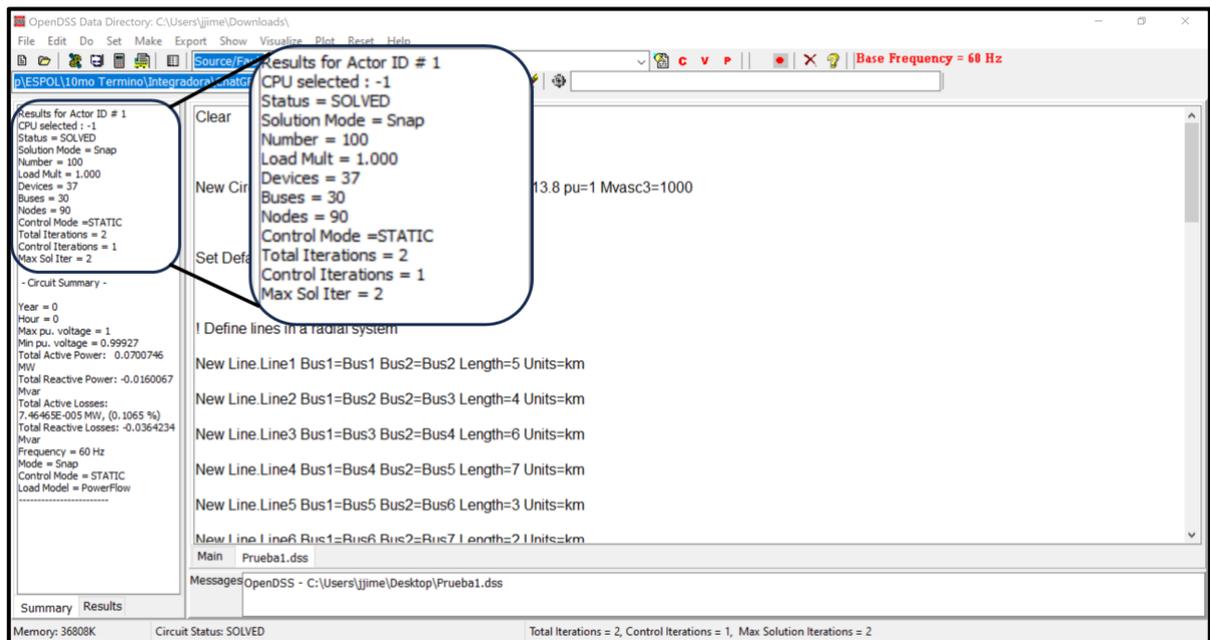


Figura 5.4 Resultados de la simulación del Caso de estudio 2

La Figura 5.4 muestra los resultados de la simulación al generar un circuito de gran tamaño. De acuerdo a los datos de la simulación, se logró compilar exitosamente un sistema trifásico de 30 barras, incluyendo el flujo de carga de las 7 cargas conectadas en distintos puntos del sistema.

5.3 Caso de estudio 3: Generación distribuida

Para el tercer caso de estudio se planteará la creación de un sistema que refleja de mejor manera los sistemas modernos. Actualmente, existen cargas de distintos tipos, entre las cuales se pueden destacar las comerciales y residenciales. El objetivo de esta prueba consiste en analizar las capacidades de ChatGPT para generar un sistema que sea capaz de diferenciar ambos tipos de cargas. Adicionalmente, debido a la tendencia actual de ser más eficientes en el consumo eléctrico, se planteó que cada barra con una carga comercial posea adicionalmente un sistema de generación fotovoltaica, el cual genere la

mitad de la potencia requerida por la carga en dicha barra. La Figura 5.5 muestra una representación del sistema a generar para este caso de estudio.

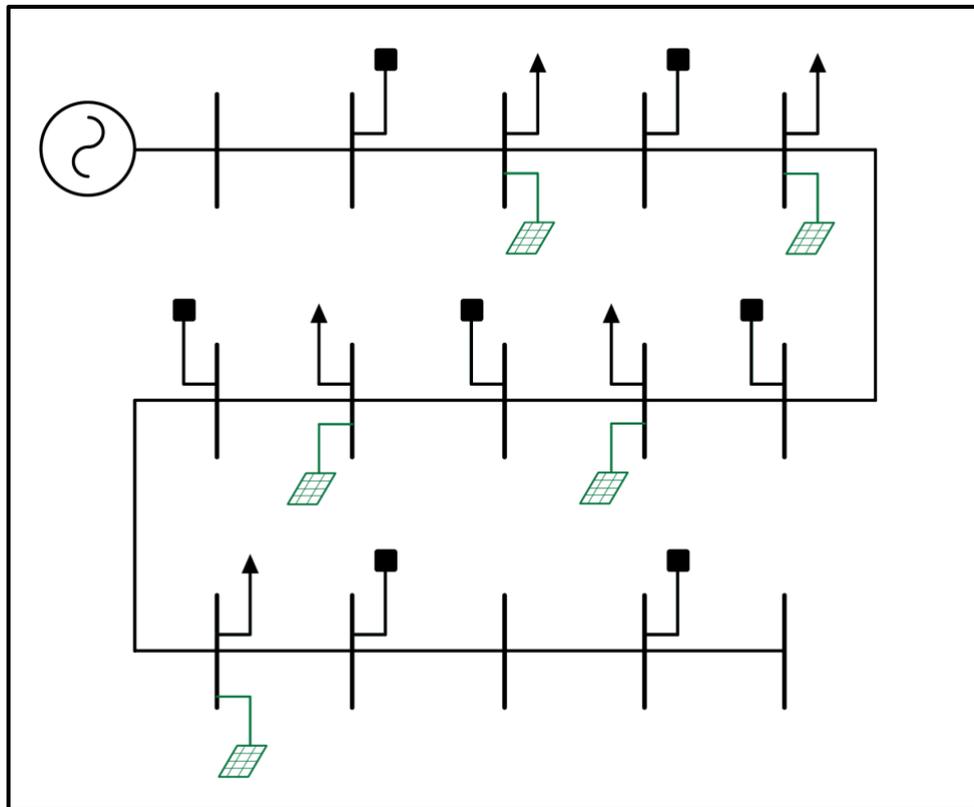


Figura 5.5 Diagrama del sistema del Caso de estudio 3

User:

Create a three-phase OpenDSS model with the following characteristics:

- Clear previous circuits
- Create a new circuit named "Test_Circuit", "Bus1.1.2.3", basekV=13.8, pu=1, Mvasc3=1000
- Set the Default Base Frequency at 60Hz

The system must contain the following:

- The system starts at "Bus1.1.2.3"
- Lines connecting 15 busses in a radial system
- The lines must be between 1 and 5 (units=km)

- Define 7 residential loads (kV=13.8, kW=between 25 and 100, pf=between 0.93 and 0.96)
- Define 5 commercial loads (kV=13.8, kW=between 200 and 500, pf=between 0.88 and 0.93)
- Create a PV System in each bus with a commercial load, and make the kVA of each PV System half of the load's kW connected to that bus

- Set Voltage Bases = [13.8]
- Calculate Voltage Bases
- Add "solve" at the end to run power flow

Observations:

- Don't use "Linecodes," define each line individually
- Don't use "LineGeometry"
- Don't add characteristics to the circuit, lines, busses or loads that were not explicitly stated
- Don't create busses at the beginning, declare them when creating the lines
- When creating the lines, the length and units must be declared separately
- The loads must be two-phase, meaning that each load must be defined as loadx.1.2 or loadx.2.3 depending on which phases are connected (the letter "x" represents the number of the bus). You have the liberty to choose which phases to use for each load
- You have the liberty to choose in which bus goes which load, but there can maximum 1 load per bus
- When naming the loads, use specific names to differentiate the residential from the commercial ones

ChatGPT:

Certainly! Here is the complete OpenDSS script for your specified model, including all lines, loads and PV Systems:

! Clearing previous circuits

Clear

! Creating a new circuit

New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1.0
Mvasc3=1000

! Setting Default Base Frequency

Set Frequency=60

! Creating 15 buses with lines in a radial system

New Line.Line1 Bus1=Bus1.1.2.3 Bus2=Bus2 Length=1.5 Units=km

New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=3.2 Units=km

New Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=2.7 Units=km

New Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=4.5 Units=km

New Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=2.1 Units=km

New Line.Line6 Bus1=Bus6 Bus2=Bus7 Length=1.2 Units=km

New Line.Line7 Bus1=Bus7 Bus2=Bus8 Length=4.8 Units=km

New Line.Line8 Bus1=Bus8 Bus2=Bus9 Length=3.3 Units=km

New Line.Line9 Bus1=Bus9 Bus2=Bus10 Length=1.8 Units=km

New Line.Line10 Bus1=Bus10 Bus2=Bus11 Length=2.5 Units=km

New Line.Line11 Bus1=Bus11 Bus2=Bus12 Length=4.2 Units=km

New Line.Line12 Bus1=Bus12 Bus2=Bus13 Length=1.9 Units=km

New Line.Line13 Bus1=Bus13 Bus2=Bus14 Length=3.5 Units=km

New Line.Line14 Bus1=Bus14 Bus2=Bus15 Length=2.8 Units=km

! Defining 7 residential loads

New Load.RLoad1 Bus1=Bus2.1.2 kW=30 kV=13.8 PF=0.95

New Load.RLoad2 Bus1=Bus4.2.3 kW=25 kV=13.8 PF=0.93

New Load.RLoad3 Bus1=Bus6.1.2 kW=60 kV=13.8 PF=0.96

New Load.RLoad4 Bus1=Bus8.2.3 kW=80 kV=13.8 PF=0.94

New Load.RLoad5 Bus1=Bus10.1.2 kW=40 kV=13.8 PF=0.95

```
New Load.RLoad6 Bus1=Bus12.2.3 kW=90 kV=13.8 PF=0.94
New Load.RLoad7 Bus1=Bus14.1.2 kW=100 kV=13.8 PF=0.93

! Defining 5 commercial loads
New Load.CLoad1 Bus1=Bus3.1.2 kW=200 kV=13.8 PF=0.91
New Load.CLoad2 Bus1=Bus5.2.3 kW=250 kV=13.8 PF=0.88
New Load.CLoad3 Bus1=Bus7.1.2 kW=350 kV=13.8 PF=0.90
New Load.CLoad4 Bus1=Bus9.2.3 kW=400 kV=13.8 PF=0.89
New Load.CLoad5 Bus1=Bus11.1.2 kW=500 kV=13.8 PF=0.92

! Adding PVSystems at each commercial load bus with kVA equal to half of
the load's kW on their corresponding bus
New PVSystem.PV1 Bus1=Bus3 Phases=3 kV=13.8 kVA=100 PF=1
New PVSystem.PV2 Bus1=Bus5 Phases=3 kV=13.8 kVA=125 PF=1
New PVSystem.PV3 Bus1=Bus7 Phases=3 kV=13.8 kVA=175 PF=1
New PVSystem.PV4 Bus1=Bus9 Phases=3 kV=13.8 kVA=200 PF=1
New PVSystem.PV5 Bus1=Bus11 Phases=3 kV=13.8 kVA=250 PF=1
Set VoltageBases=[13.8]
CalcVoltageBases

Solve
```

Analizando los resultados de este caso de estudio, se puede apreciar por medio de la conversación que ChatGPT logró generar exitosamente un código que cumpliera con todas las condiciones de la orden, respetando la sintaxis de OpenDSS. Se crearon 15 barras en un sistema radial, al igual que todas las cargas, tanto residenciales como comerciales. La IA logró satisfacer la orden de diferenciar el tipo de carga por medio del nombre, al igual que considerar a todas las cargas como bifásicas, generando un sistema desbalanceado, asemejándose a sistemas reales.

La introducción de generación distribuida también fue exitosa por medio de la definición de paneles fotovoltaicos. Fue llamativa la facilidad con la que ChatGPT logró definir

correctamente estos elementos, ya que, a diferencia de las líneas y cargas, no se le proporcionó mayor información con respecto a estos componentes del sistema. Cabe resaltar el hecho de que cumplió correctamente la orden de que todos los sistemas fotovoltaicos estén ubicados en barras con cargas comerciales, al igual de que la potencia aparente de la generación distribuida sea la mitad de la potencia activa en cada una de las barras.

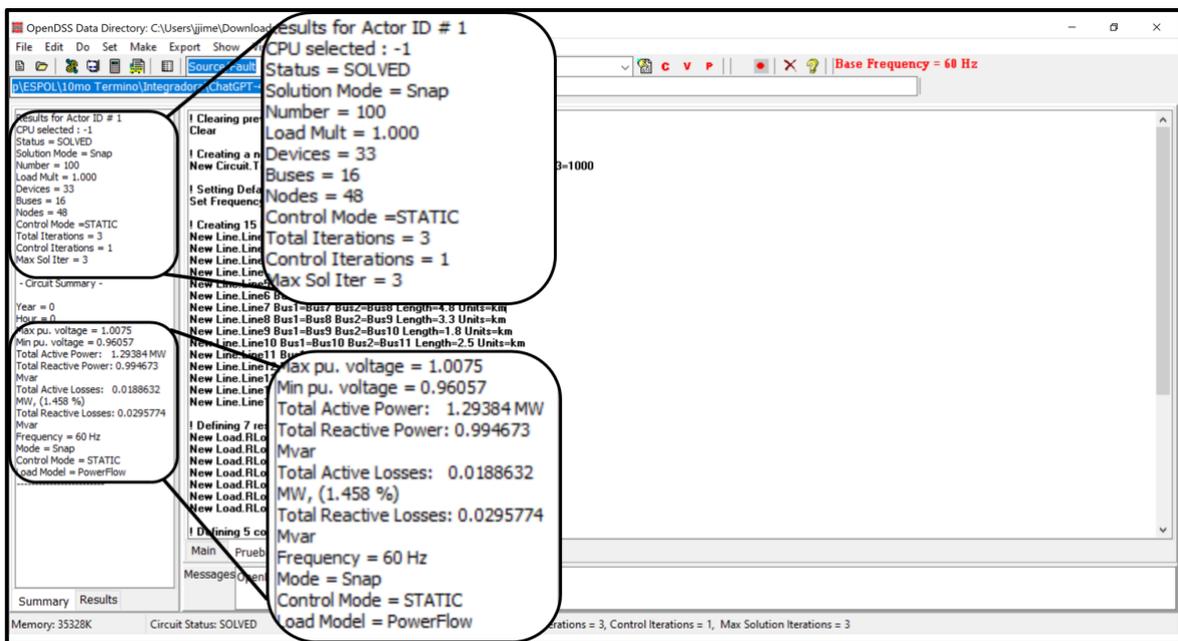


Figura 5.6 Resultados de la simulación del Caso de estudio 3

La Figura 5.6 muestra los resultados de la validación en OpenDSS. Al igual que en los casos anteriores, se muestran los resultados de la simulación con respecto a la cantidad de elementos, barras e iteraciones que le tomó al programa resolver el sistema. Adicionalmente, debido a la naturaleza del modelo y su similitud con sistemas reales, se muestran también parámetros como voltaje, potencia y pérdidas en la red, evidenciando cómo sistemas desbalanceados pueden llegar a tener problemas de calidad de energía, especialmente cuando se dan expansiones no planificadas.

5.4 Caso de estudio 4: Expansión de sistemas

Para el siguiente caso de estudio, se realizarán expansiones al código generado en el caso 3. En sistemas de distribución reales, las expansiones de dichos sistemas ocurren constantemente, por lo que una correcta planificación es indispensable para velar por integridad de la misma. El caso se enfocará en añadir nuevas barras y cargas, para luego analizar los efectos de dichos cambios en la red. La Figura 5.7 detalla en rojo la expansión del sistema que se busca generar por medio de la IA.

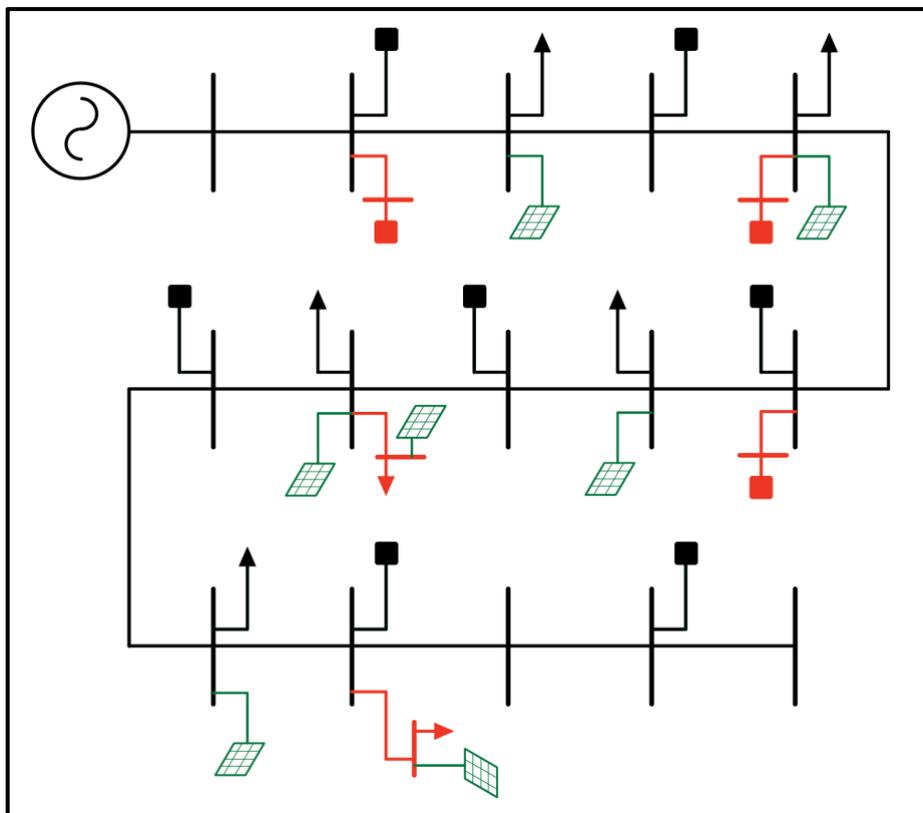


Figura 5.7 Diagrama del sistema del Caso de estudio 4

User:

The following is an OpenDSS code:

```
! Clearing previous circuits
```

```
Clear
```

! Creating a new circuit

New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1.0
Mvasc3=1000

! Setting Default Base Frequency

Set Frequency=60

! Creating 15 buses with lines in a radial system

New Line.Line1 Bus1=Bus1.1.2.3 Bus2=Bus2 Length=1.5 Units=km

New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=3.2 Units=km

New Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=2.7 Units=km

New Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=4.5 Units=km

New Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=2.1 Units=km

New Line.Line6 Bus1=Bus6 Bus2=Bus7 Length=1.2 Units=km

New Line.Line7 Bus1=Bus7 Bus2=Bus8 Length=4.8 Units=km

New Line.Line8 Bus1=Bus8 Bus2=Bus9 Length=3.3 Units=km

New Line.Line9 Bus1=Bus9 Bus2=Bus10 Length=1.8 Units=km

New Line.Line10 Bus1=Bus10 Bus2=Bus11 Length=2.5 Units=km

New Line.Line11 Bus1=Bus11 Bus2=Bus12 Length=4.2 Units=km

New Line.Line12 Bus1=Bus12 Bus2=Bus13 Length=1.9 Units=km

New Line.Line13 Bus1=Bus13 Bus2=Bus14 Length=3.5 Units=km

New Line.Line14 Bus1=Bus14 Bus2=Bus15 Length=2.8 Units=km

New Line.Line15 Bus1=Bus15 Bus2=Bus16 Length=4.6 Units=km

! Defining 7 residential loads

New Load.RLoad1 Bus1=Bus2.1.2 kW=30 kV=13.8 PF=0.95

New Load.RLoad2 Bus1=Bus4.2.3 kW=25 kV=13.8 PF=0.93

New Load.RLoad3 Bus1=Bus6.1.2 kW=60 kV=13.8 PF=0.96

New Load.RLoad4 Bus1=Bus8.2.3 kW=80 kV=13.8 PF=0.94

New Load.RLoad5 Bus1=Bus10.1.2 kW=40 kV=13.8 PF=0.95

New Load.RLoad6 Bus1=Bus12.2.3 kW=90 kV=13.8 PF=0.94

New Load.RLoad7 Bus1=Bus14.1.2 kW=100 kV=13.8 PF=0.93

! Defining 5 commercial loads

New Load.CLoad1 Bus1=Bus3.1.2 kW=200 kV=13.8 PF=0.91

New Load.CLoad2 Bus1=Bus5.2.3 kW=250 kV=13.8 PF=0.88

New Load.CLoad3 Bus1=Bus7.1.2 kW=350 kV=13.8 PF=0.90

New Load.CLoad4 Bus1=Bus9.2.3 kW=400 kV=13.8 PF=0.89

New Load.CLoad5 Bus1=Bus11.1.2 kW=500 kV=13.8 PF=0.92

! Adding PVSystems at each commercial load bus with kVA equal to half of the load's kW on their corresponding bus

New PVSystem.PV1 Bus1=Bus3 Phases=3 kV=13.8 kVA=100 PF=1

New PVSystem.PV2 Bus1=Bus5 Phases=3 kV=13.8 kVA=125 PF=1

New PVSystem.PV3 Bus1=Bus7 Phases=3 kV=13.8 kVA=175 PF=1

New PVSystem.PV4 Bus1=Bus9 Phases=3 kV=13.8 kVA=200 PF=1

New PVSystem.PV5 Bus1=Bus11 Phases=3 kV=13.8 kVA=250 PF=1

Set VoltageBases=[13.8]

CalcVoltageBases

Solve

I want you to expand this system in the following way:

- Create 5 new busses. Each bus branches from the following existing busses: 2, 5, 6, 9, 12. The line to each new bus is between 1 and 3 km
- Add 3 new residential loads to the first 3 new busses (the loads are identical to the ones connected to the bus from where the expansion happened)
- Add 2 new commercial loads to the last 2 new busses (the loads are identical to the ones connected to the bus from where the expansion happened)
- To each new bus with a commercial load, add a PV System with a kVA equal to half of the load's kW in that bus

ChatGPT:

To expand the OpenDSS system as per your requirements, I'll create 5 new buses, each branching from the specified existing buses. Additionally, I'll add residential and commercial loads to these new buses, replicating the characteristics of the loads connected to their parent buses. Finally, I'll add PV Systems to the buses with commercial loads. Here is the modified OpenDSS code reflecting these changes:

! Adding 5 new buses branching from the existing buses: 2, 5, 6, 9, 12

New Line.Line16 Bus1=Bus2 Bus2=Bus17 Length=2.1 Units=km

New Line.Line17 Bus1=Bus5 Bus2=Bus18 Length=1.8 Units=km

New Line.Line18 Bus1=Bus6 Bus2=Bus19 Length=2.5 Units=km

New Line.Line19 Bus1=Bus9 Bus2=Bus20 Length=1.3 Units=km

New Line.Line20 Bus1=Bus12 Bus2=Bus21 Length=2.9 Units=km

! Adding 3 new residential loads to the first 3 new buses (Bus17, Bus18, Bus19)

! The loads are identical to the ones connected to the respective parent buses

New Load.RLoad8 Bus1=Bus17.1.2 kW=30 kV=13.8 PF=0.95

New Load.RLoad9 Bus1=Bus18.2.3 kW=25 kV=13.8 PF=0.93

New Load.RLoad10 Bus1=Bus19.1.2 kW=60 kV=13.8 PF=0.96

! Adding 2 new commercial loads to the last 2 new buses (Bus20, Bus21)

! The loads are identical to the ones connected to the respective parent buses

New Load.CLoad6 Bus1=Bus20.2.3 kW=400 kV=13.8 PF=0.89

New Load.CLoad7 Bus1=Bus21.1.2 kW=500 kV=13.8 PF=0.92

! Adding PV Systems to each new bus with a commercial load

! The kVA of each PV System is half of the load's kW in that bus

New PVSystem.PV6 Bus1=Bus20 Phases=3 kV=13.8 kVA=200 PF=1

New PVSystem.PV7 Bus1=Bus21 Phases=3 kV=13.8 kVA=250 PF=1

```
! Rest of the existing system configuration remains unchanged
```

```
Set VoltageBases=[13.8]
```

```
CalcVoltageBases
```

```
Solve
```

This code will effectively expand the existing system with new buses, loads, and PV systems as requested.

User:

Now write the entire code, the original one with the one you just created

ChatGPT:

Certainly! Below is the complete OpenDSS code, combining the original system with the expansions you requested:

```
! Clearing previous circuits
```

```
Clear
```

```
! Creating a new circuit
```

```
New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1.0  
Mvasc3=1000
```

```
! Setting Default Base Frequency
```

```
Set Frequency=60
```

```
! Creating 15 buses with lines in a radial system
```

```
New Line.Line1 Bus1=Bus1.1.2.3 Bus2=Bus2 Length=1.5 Units=km
```

```
New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=3.2 Units=km
```

```
New Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=2.7 Units=km
```

```
New Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=4.5 Units=km
```

```
New Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=2.1 Units=km
```

New Line.Line6 Bus1=Bus6 Bus2=Bus7 Length=1.2 Units=km
New Line.Line7 Bus1=Bus7 Bus2=Bus8 Length=4.8 Units=km
New Line.Line8 Bus1=Bus8 Bus2=Bus9 Length=3.3 Units=km
New Line.Line9 Bus1=Bus9 Bus2=Bus10 Length=1.8 Units=km
New Line.Line10 Bus1=Bus10 Bus2=Bus11 Length=2.5 Units=km
New Line.Line11 Bus1=Bus11 Bus2=Bus12 Length=4.2 Units=km
New Line.Line12 Bus1=Bus12 Bus2=Bus13 Length=1.9 Units=km
New Line.Line13 Bus1=Bus13 Bus2=Bus14 Length=3.5 Units=km
New Line.Line14 Bus1=Bus14 Bus2=Bus15 Length=2.8 Units=km
New Line.Line15 Bus1=Bus15 Bus2=Bus16 Length=4.6 Units=km

! Defining 7 residential loads

New Load.RLoad1 Bus1=Bus2.1.2 kW=30 kV=13.8 PF=0.95
New Load.RLoad2 Bus1=Bus4.2.3 kW=25 kV=13.8 PF=0.93
New Load.RLoad3 Bus1=Bus6.1.2 kW=60 kV=13.8 PF=0.96
New Load.RLoad4 Bus1=Bus8.2.3 kW=80 kV=13.8 PF=0.94
New Load.RLoad5 Bus1=Bus10.1.2 kW=40 kV=13.8 PF=0.95
New Load.RLoad6 Bus1=Bus12.2.3 kW=90 kV=13.8 PF=0.94
New Load.RLoad7 Bus1=Bus14.1.2 kW=100 kV=13.8 PF=0.93

! Defining 5 commercial loads

New Load.CLoad1 Bus1=Bus3.1.2 kW=200 kV=13.8 PF=0.91
New Load.CLoad2 Bus1=Bus5.2.3 kW=250 kV=13.8 PF=0.88
New Load.CLoad3 Bus1=Bus7.1.2 kW=350 kV=13.8 PF=0.90
New Load.CLoad4 Bus1=Bus9.2.3 kW=400 kV=13.8 PF=0.89
New Load.CLoad5 Bus1=Bus11.1.2 kW=500 kV=13.8 PF=0.92

! Adding PVSystems at each commercial load bus with kVA equal to half of the load's kW on their corresponding bus

New PVSystem.PV1 Bus1=Bus3 Phases=3 kV=13.8 kVA=100 PF=1
New PVSystem.PV2 Bus1=Bus5 Phases=3 kV=13.8 kVA=125 PF=1
New PVSystem.PV3 Bus1=Bus7 Phases=3 kV=13.8 kVA=175 PF=1

```
New PVSystem.PV4 Bus1=Bus9 Phases=3 kV=13.8 kVA=200 PF=1
New PVSystem.PV5 Bus1=Bus11 Phases=3 kV=13.8 kVA=250 PF=1
```

! Adding 5 new buses branching from the existing buses: 2, 5, 6, 9, 12

```
New Line.Line16 Bus1=Bus2 Bus2=Bus17 Length=2.1 Units=km
```

```
New Line.Line17 Bus1=Bus5 Bus2=Bus18 Length=1.8 Units=km
```

```
New Line.Line18 Bus1=Bus6 Bus2=Bus19 Length=2.5 Units=km
```

```
New Line.Line19 Bus1=Bus9 Bus2=Bus20 Length=1.3 Units=km
```

```
New Line.Line20 Bus1=Bus12 Bus2=Bus21 Length=2.9 Units=km
```

! Adding 3 new residential loads to the first 3 new buses (Bus17, Bus18, Bus19)

```
New Load.RLoad8 Bus1=Bus17.1.2 kW=30 kV=13.8 PF=0.95
```

```
New Load.RLoad9 Bus1=Bus18.2.3 kW=25 kV=13.8 PF=0.93
```

```
New Load.RLoad10 Bus1=Bus19.1.2 kW=60 kV=13.8 PF=0.96
```

! Adding 2 new commercial loads to the last 2 new buses (Bus20, Bus21)

```
New Load.CLoad6 Bus1=Bus20.2.3 kW=400 kV=13.8 PF=0.89
```

```
New Load.CLoad7 Bus1=Bus21.1.2 kW=500 kV=13.8 PF=0.92
```

! Adding PV Systems to each new bus with a commercial load

```
New PVSystem.PV6 Bus1=Bus20 Phases=3 kV=13.8 kVA=200 PF=1
```

```
New PVSystem.PV7 Bus1=Bus21 Phases=3 kV=13.8 kVA=250 PF=1
```

```
Set VoltageBases=[13.8]
```

```
CalcVoltageBases
```

```
Solve
```

This code integrates the new buses, loads, and PV systems into the original configuration, creating an expanded and more comprehensive electrical distribution system.

Inicialmente, tras recibir la orden de expandir el sistema, ChatGPT genera únicamente la expansión, mas no el código original junto con las nuevas líneas de código. Sin embargo, dicha expansión la realiza de manera exitosa, cumpliendo todas las instrucciones sin mayor dificultad. Mediante una simple orden subsiguiente de escribir tanto el código original como la expansión, la IA genera el código completo, añadiendo comentarios para una mejor comprensión de las secciones del código.

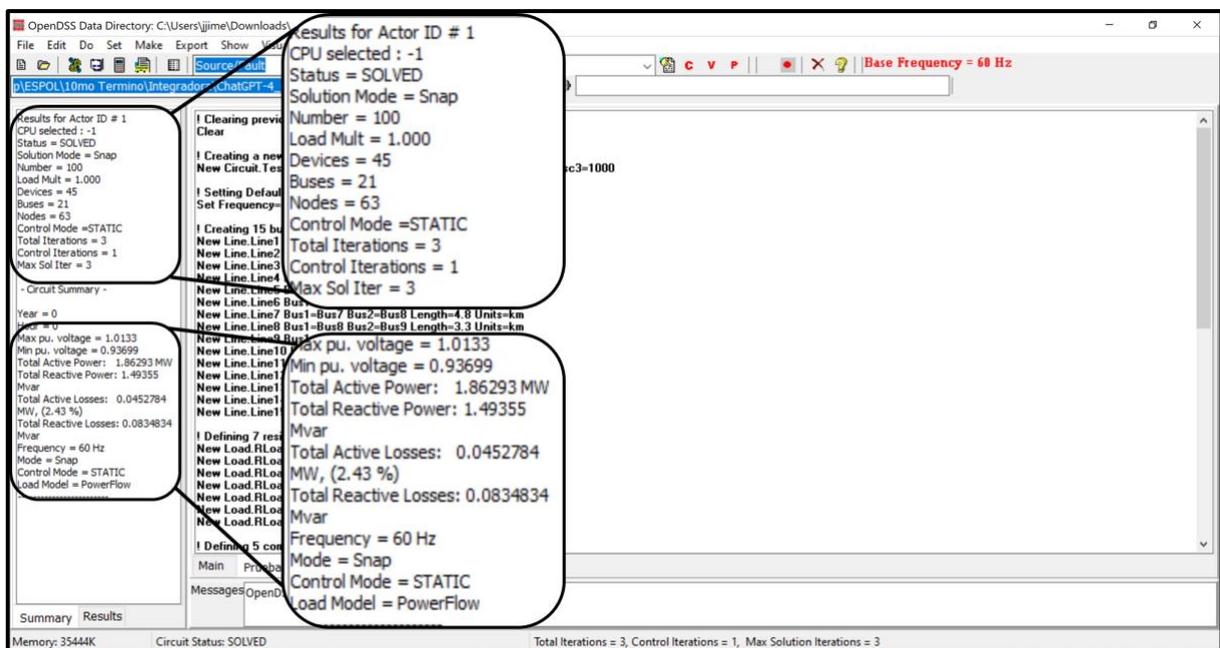


Figura 5.8 Resultados de la simulación del Caso de estudio 4

De manera similar al caso de estudio anterior, en la Figura 5.8 se muestran resultados relevantes tanto con respecto al proceso de solución del sistema como de valores importantes en el análisis de la red. Como se mencionó al final del caso previo, una expansión sin planificación puede llevar a serios problemas de calidad de energía y estabilidad de sistemas eléctricos de distribución. Como se observan en los resultados, la expansión del sistema ocasionó en una reducción aún mayor del voltaje por unidad en algunas barras, al igual que la cantidad de pérdidas en la red. Esta realidad debería llevar a los encargados de los sistemas de distribución a desarrollar expansiones planificadas y a buscar maneras de solucionar problemas en caso de presentarse.

5.5 Caso de estudio 5: Solución de problemas

Para el siguiente caso de estudio, se planteó un objetivo distinto a todos los casos anteriores, analizar la capacidad de ChatGPT para solucionar problemas en la red. Se le proporcionó a la IA el mismo código que generó en el caso anterior junto con los resultados de la simulación, con la intención de que detecte problemas en la red y proponga soluciones a las mismas. Debido a que se usó el mismo sistema del caso de estudio anterior, la Figura 5.7 muestra la red eléctrica a analizar.

User:

The following is an OpenDSS code:

```
! Clearing previous circuits
```

```
Clear
```

```
! Creating a new circuit
```

```
New Circuit.Test_Circuit Bus1=Bus1.1.2.3 BasekV=13.8 pu=1.0  
Mvasc3=1000
```

```
! Setting Default Base Frequency
```

```
Set Frequency=60
```

```
! Creating 15 buses with lines in a radial system
```

```
New Line.Line1 Bus1=Bus1.1.2.3 Bus2=Bus2 Length=1.5 Units=km
```

```
New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=3.2 Units=km
```

```
New Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=2.7 Units=km
```

```
New Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=4.5 Units=km
```

```
New Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=2.1 Units=km
```

```
New Line.Line6 Bus1=Bus6 Bus2=Bus7 Length=1.2 Units=km
```

```
New Line.Line7 Bus1=Bus7 Bus2=Bus8 Length=4.8 Units=km
```

```
New Line.Line8 Bus1=Bus8 Bus2=Bus9 Length=3.3 Units=km
```

```
New Line.Line9 Bus1=Bus9 Bus2=Bus10 Length=1.8 Units=km
```

```
New Line.Line10 Bus1=Bus10 Bus2=Bus11 Length=2.5 Units=km
```

New Line.Line11 Bus1=Bus11 Bus2=Bus12 Length=4.2 Units=km
New Line.Line12 Bus1=Bus12 Bus2=Bus13 Length=1.9 Units=km
New Line.Line13 Bus1=Bus13 Bus2=Bus14 Length=3.5 Units=km
New Line.Line14 Bus1=Bus14 Bus2=Bus15 Length=2.8 Units=km
New Line.Line15 Bus1=Bus15 Bus2=Bus16 Length=4.6 Units=km

! Defining 7 residential loads

New Load.RLoad1 Bus1=Bus2.1.2 kW=30 kV=13.8 PF=0.95
New Load.RLoad2 Bus1=Bus4.2.3 kW=25 kV=13.8 PF=0.93
New Load.RLoad3 Bus1=Bus6.1.2 kW=60 kV=13.8 PF=0.96
New Load.RLoad4 Bus1=Bus8.2.3 kW=80 kV=13.8 PF=0.94
New Load.RLoad5 Bus1=Bus10.1.2 kW=40 kV=13.8 PF=0.95
New Load.RLoad6 Bus1=Bus12.2.3 kW=90 kV=13.8 PF=0.94
New Load.RLoad7 Bus1=Bus14.1.2 kW=100 kV=13.8 PF=0.93

! Defining 5 commercial loads

New Load.CLoad1 Bus1=Bus3.1.2 kW=200 kV=13.8 PF=0.91
New Load.CLoad2 Bus1=Bus5.2.3 kW=250 kV=13.8 PF=0.88
New Load.CLoad3 Bus1=Bus7.1.2 kW=350 kV=13.8 PF=0.90
New Load.CLoad4 Bus1=Bus9.2.3 kW=400 kV=13.8 PF=0.89
New Load.CLoad5 Bus1=Bus11.1.2 kW=500 kV=13.8 PF=0.92

! Adding PVSystems at each commercial load bus with kVA equal to half of the load's kW on their corresponding bus

New PVSystem.PV1 Bus1=Bus3 Phases=3 kV=13.8 kVA=100 PF=1
New PVSystem.PV2 Bus1=Bus5 Phases=3 kV=13.8 kVA=125 PF=1
New PVSystem.PV3 Bus1=Bus7 Phases=3 kV=13.8 kVA=175 PF=1
New PVSystem.PV4 Bus1=Bus9 Phases=3 kV=13.8 kVA=200 PF=1
New PVSystem.PV5 Bus1=Bus11 Phases=3 kV=13.8 kVA=250 PF=1

! Adding 5 new buses branching from the existing buses: 2, 5, 6, 9, 12

New Line.Line16 Bus1=Bus2 Bus2=Bus17 Length=2.1 Units=km

New Line.Line17 Bus1=Bus5 Bus2=Bus18 Length=1.8 Units=km
New Line.Line18 Bus1=Bus6 Bus2=Bus19 Length=2.5 Units=km
New Line.Line19 Bus1=Bus9 Bus2=Bus20 Length=1.3 Units=km
New Line.Line20 Bus1=Bus12 Bus2=Bus21 Length=2.9 Units=km

! Adding 3 new residential loads to the first 3 new buses (Bus17, Bus18, Bus19)

New Load.RLoad8 Bus1=Bus17.1.2 kW=30 kV=13.8 PF=0.95
New Load.RLoad9 Bus1=Bus18.2.3 kW=25 kV=13.8 PF=0.93
New Load.RLoad10 Bus1=Bus19.1.2 kW=60 kV=13.8 PF=0.96

! Adding 2 new commercial loads to the last 2 new buses (Bus20, Bus21)

New Load.CLoad6 Bus1=Bus20.2.3 kW=400 kV=13.8 PF=0.89
New Load.CLoad7 Bus1=Bus21.1.2 kW=500 kV=13.8 PF=0.92

! Adding PV Systems to each new bus with a commercial load

New PVSystem.PV6 Bus1=Bus20 Phases=3 kV=13.8 kVA=200 PF=1
New PVSystem.PV7 Bus1=Bus21 Phases=3 kV=13.8 kVA=250 PF=1

Set VoltageBases=[13.8]

CalcVoltageBases

Solve

And these are the results of a power flow simulation:

LINE-LINE VOLTAGES BY BUS & NODE

Bus	Node	VLN (kV)	Angle	pu	Base kV
BUS1	1-2	13.785 /_	29.9	0.99894	13.800
	- 2-3	13.771 /_	-90.1	0.99786	13.800

- 3-1	13.765	/	150.0	0.99748	13.800
BUS2 1-2	13.767	/	29.9	0.9976	13.800
- 2-3	13.733	/	-90.2	0.99517	13.800
- 3-1	13.727	/	149.9	0.99469	13.800
BUS3 1-2	13.729	/	29.7	0.99482	13.800
- 2-3	13.656	/	-90.5	0.98954	13.800
- 3-1	13.646	/	149.9	0.98886	13.800
BUS4 1-2	13.699	/	29.7	0.99272	13.800
- 2-3	13.594	/	-90.7	0.98504	13.800
- 3-1	13.582	/	149.9	0.98417	13.800
BUS5 1-2	13.651	/	29.5	0.98923	13.800
- 2-3	13.491	/	-91.0	0.97764	13.800
- 3-1	13.475	/	149.8	0.97646	13.800
BUS6 1-2	13.629	/	29.4	0.98763	13.800
- 2-3	13.45	/	-91.1	0.97461	13.800
- 3-1	13.432	/	149.8	0.97332	13.800
BUS7 1-2	13.618	/	29.4	0.98679	13.800
- 2-3	13.427	/	-91.2	0.97296	13.800
- 3-1	13.408	/	149.8	0.9716	13.800
BUS8 1-2	13.582	/	29.3	0.98419	13.800
- 2-3	13.347	/	-91.4	0.96715	13.800
- 3-1	13.324	/	149.8	0.9655	13.800
BUS9 1-2	13.558	/	29.2	0.98244	13.800
- 2-3	13.294	/	-91.6	0.96334	13.800
- 3-1	13.269	/	149.8	0.96154	13.800
BUS10 1-2	13.545	/	29.1	0.98153	13.800
- 2-3	13.28	/	-91.6	0.96228	13.800
- 3-1	13.255	/	149.8	0.9605	13.800
BUS11 1-2	13.528	/	29.1	0.98031	13.800
- 2-3	13.26	/	-91.7	0.96086	13.800
- 3-1	13.236	/	149.7	0.95911	13.800
BUS12 1-2	13.512	/	29.1	0.97916	13.800

- 2-3 13.24 /_ -91.8 0.95942 13.800
 - 3-1 13.216 /_ 149.7 0.95767 13.800
 BUS13 1-2 13.511 /_ 29.0 0.97906 13.800
 - 2-3 13.239 /_ -91.8 0.95931 13.800
 - 3-1 13.214 /_ 149.7 0.95756 13.800
 BUS14 1-2 13.508 /_ 29.0 0.97887 13.800
 - 2-3 13.236 /_ -91.8 0.95911 13.800
 - 3-1 13.212 /_ 149.7 0.95737 13.800
 BUS15 1-2 13.508 /_ 29.0 0.97888 13.800
 - 2-3 13.236 /_ -91.8 0.95911 13.800
 - 3-1 13.212 /_ 149.7 0.95737 13.800
 BUS16 1-2 13.509 /_ 29.0 0.97888 13.800
 - 2-3 13.236 /_ -91.8 0.95912 13.800
 - 3-1 13.212 /_ 149.7 0.95737 13.800
 BUS17 1-2 13.766 /_ 29.9 0.99756 13.800
 - 2-3 13.733 /_ -90.2 0.99514 13.800
 - 3-1 13.726 /_ 149.9 0.99466 13.800
 BUS18 1-2 13.651 /_ 29.5 0.98923 13.800
 - 2-3 13.491 /_ -91.0 0.97761 13.800
 - 3-1 13.475 /_ 149.8 0.97642 13.800
 BUS19 1-2 13.628 /_ 29.4 0.98756 13.800
 - 2-3 13.449 /_ -91.1 0.97454 13.800
 - 3-1 13.431 /_ 149.8 0.97325 13.800
 BUS20 1-2 13.557 /_ 29.2 0.98242 13.800
 - 2-3 13.289 /_ -91.6 0.96298 13.800
 - 3-1 13.264 /_ 149.8 0.96114 13.800
 BUS21 1-2 13.504 /_ 29.0 0.97855 13.800
 - 2-3 13.231 /_ -91.8 0.95877 13.800
 - 3-1 13.207 /_ 149.7 0.95706 13.800

Using criteria of electrical engineering, make changes to the code to address the issues

ChatGPT:

Based on the results of the power flow simulation, it appears there are issues with voltage drop along the distribution line. As you progress further from the source (BUS1), the voltage levels decrease. This is a common issue in distribution systems, especially in radial systems like the one you've modeled. To address the voltage drop issue in your OpenDSS model, you can add capacitor banks at key buses. Capacitor banks are used in power distribution to improve the power factor, which in turn helps in reducing voltage drops across the network. The selection of these key buses typically depends on where the most significant voltage drops occur, often further from the source.

From your power flow simulation results, it appears that voltage levels start to drop more significantly from BUS5 onwards. Therefore, adding capacitor banks at strategic buses beyond BUS5 might be beneficial.

Here's an example of how you can modify your OpenDSS code to include capacitor banks at BUS5, BUS10, and BUS15:

! Adding capacitor banks

```
New Capacitor.CapBus5 Bus1=BUS5 kVAR=500 kV=13.8
```

```
New Capacitor.CapBus10 Bus1=BUS10 kVAR=500 kV=13.8
```

```
New Capacitor.CapBus15 Bus1=BUS15 kVAR=500 kV=13.8
```

In this example, I've added capacitors with a rating of 300 kVAR at 13.8 kV to BUS5, BUS10, and BUS15. You might need to adjust these values (kVAR ratings) based on the actual reactive power requirements at these buses. This would typically be determined by a more detailed analysis of the load flow results and the reactive power demand at these locations.

Remember, the effectiveness of capacitors in improving voltage profiles depends on several factors, including the amount of reactive power demand, the distance of the loads from the source, and the existing infrastructure. It's

often a good practice to perform a sensitivity analysis to determine the optimal size and location of these capacitors for your specific network.

Como se explicó previamente, en este caso se proporcionó a ChatGPT con el código del caso de estudio anterior junto con los resultados de la simulación del mismo. Se le pidió que analizara los resultados, indicara cuáles problemas observa en la red, y haga modificaciones al código para mejorar el estado de la misma. Indicó correctamente el hecho de que, debido a la topología radial del sistema, en las barras más alejadas de la barra inicial habían caídas significativas de tensión.

Como propuesta para solucionar este problema, la IA plantea el uso de bancos de capacitores en barras claves del sistema. Dichos bancos son colocados a lo largo de la red, con el objetivo de mantener un nivel estable y aceptable de tensión a lo largo de las barras. A continuación, se muestran los resultados de simular el sistema con los cambios sugeridos por ChatGPT.

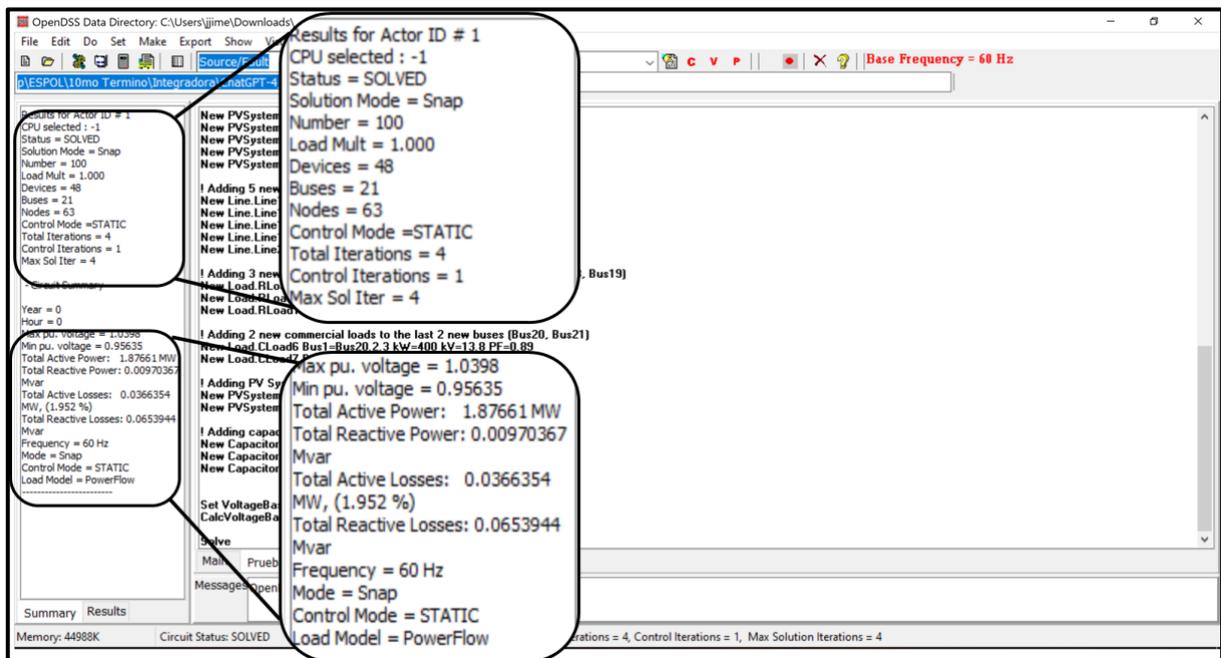


Figura 5.9 Resultados de la simulación para el Caso de estudio 5

Como se aprecia en la Figura 5.9, la adición de bancos de capacitores en las barras seleccionadas por ChatGPT resultó en una mejora en los niveles de tensión de la red. Gracias a los cambios, los valores máximos y mínimos en por unidad se encuentran en un rango de ± 0.05 por unidad, lo cual suele ser un valor aceptable. En el caso de estudio anterior, donde no se tenía los bancos de capacitores, el valor mínimo se encontraba por debajo de este valor.

5.6 Caso de estudio 6: Diseño de topología en base a cargas establecidas

Para el último caso de estudio expuesto, se le solicitará a ChatGPT que genere una topología de un sistema de distribución, proporcionándole únicamente las cargas que componen el sistema. El objetivo es evidenciar y analizar las capacidades de la IA para generar topologías funcionales y eficientes de redes de distribución con información muy general.

User:

Create an OpenDSS code following these instructions:

- Clear previous circuits
- New circuit named "Test_Circuit", "Bus1.1.2.3", "basekV"=13.8, pu=1, Mvasc3=1000
- Set the Default Base Frequency at 60Hz

The system must contain the following:

- Starting bus: Bus1.1.2.3

- The following loads:

(the loads follow this format: Load Name/kV/kW/kVAR)

- Load1/13.8/10/2/1
- Load2/13.8/10/2/1
- Load3/13.8/10/2/1
- Load4/13.8/10/2/1

- Load5/13.8/10/2/1

- Set Voltage Bases = [13.8]
- Calculate Voltage Bases
- "Solve" at the end to run power flow

Observations:

- Don't use "Linecodes," define each line individually
- Don't use "LineGeometry"
- Don't add characteristics to the circuit, lines, busses or loads that were not explicitly stated
- Don't create busses at the beginning, declare them when creating the lines
- When creating the lines, the length and units must be declared separately
- You have the liberty to choose in which bus goes which load
- No load can be in Bus1, there can be multiple loads on a single bus
- Create the lines connecting the busses, you can use any topology you choose
- Write the entirety of the code, all the lines for the lines, busses and loads

ChatGPT:

Based on your instructions, I'll create an OpenDSS code that sets up a power system with specified loads and a defined circuit structure. Here's the complete script:

```
// Clear previous circuits
```

```
Clear
```

```
// Create a new circuit
```

```
New Circuit.Test_Circuit Bus1=Bus1.1.2.3 basekV=13.8 pu=1 Mvasc3=1000
```

```

// Set the Default Base Frequency
Set Frequency=60

// Define Voltage Bases
Set VoltageBases=[13.8]
CalcVoltageBases

// Defining Loads
// Assigning loads to different buses (Bus2 to Bus6)
New Load.Load1 Bus1=Bus2 kV=13.8 kW=10 kvar=2 pf=1
New Load.Load2 Bus1=Bus3 kV=13.8 kW=10 kvar=2 pf=1
New Load.Load3 Bus1=Bus4 kV=13.8 kW=10 kvar=2 pf=1
New Load.Load4 Bus1=Bus5 kV=13.8 kW=10 kvar=2 pf=1
New Load.Load5 Bus1=Bus6 kV=13.8 kW=10 kvar=2 pf=1

// Create Lines
// Defining individual lines without using Linecodes or LineGeometry
New Line.Line1 Bus1=Bus1 Bus2=Bus2 Length=1 units=km
New Line.Line2 Bus1=Bus2 Bus2=Bus3 Length=1 units=km
New Line.Line3 Bus1=Bus3 Bus2=Bus4 Length=1 units=km
New Line.Line4 Bus1=Bus4 Bus2=Bus5 Length=1 units=km
New Line.Line5 Bus1=Bus5 Bus2=Bus6 Length=1 units=km

// Solve the power flow
Solve

```

La orden de generación de código fue bastante sencilla, pero demuestra la capacidad de ChatGPT para crear una topología basándose únicamente en información referente a las cargas de un posible sistema de distribución. En base al código generado para este caso (y otros elaborados en pruebas no incluidas en este documento), la topología de preferencia de la IA es la radial, empezando desde la Barra 1 hasta la última con carga.

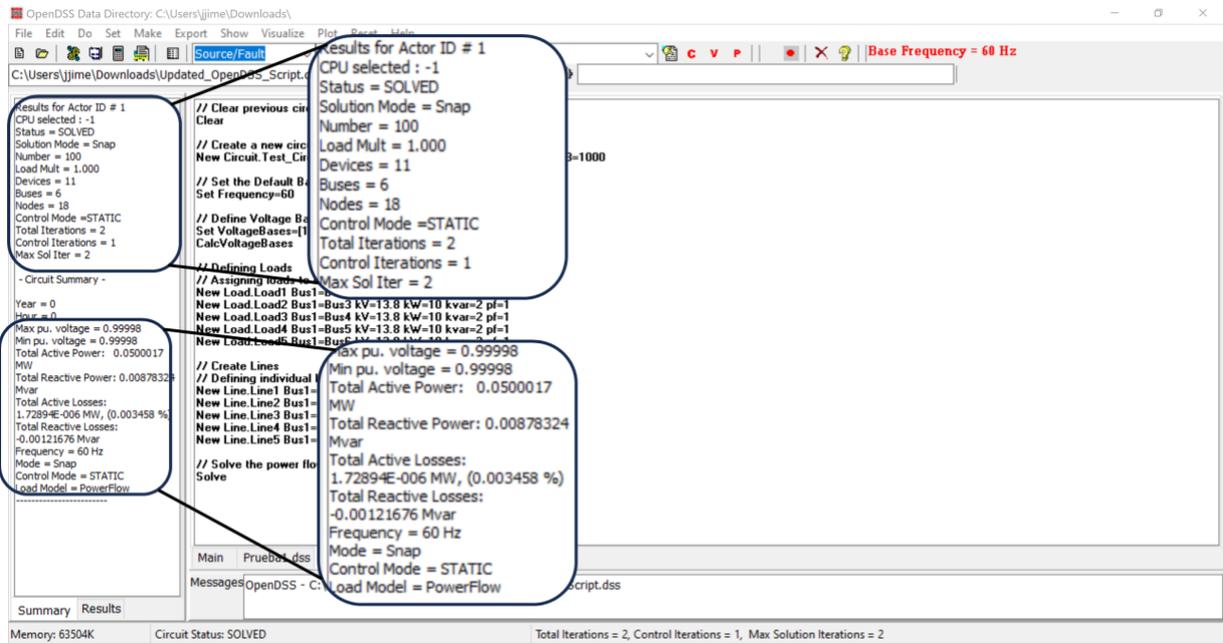


Figura 5.10 Resultados de la simulación para el Caso de estudio 6

La Figura 5.10 muestra los resultados de la simulación para este caso de estudio. Se evidencia la cantidad de barras y cargas, al igual que la potencia consumida por todo el sistema. Al contrastar las potencias de las cargas generadas en el código y las mostradas en los resultados, es posible verificar que tanto ChatGPT como OpenDSS generaron en conjunto una topología radial en base a las cargas dadas en un inicio por medio del usuario.

En base a los casos de estudios presentados en este capítulo, se logró evidenciar la capacidad de ChatGPT no solo para generar sistemas de distribución desde 0 utilizando la sintaxis de OpenDSS, sino también para analizar los resultados de dichas redes y brindar posibles soluciones para mejorar la calidad de las mismas. Cabe recalcar que esta IA no está diseñada ni especializada para realizar este tipo de trabajos, y sin embargo los logra ejecutar de manera rápida y eficiente. A pesar de las limitaciones y complicaciones que mostró en el primer caso de estudio para generar inicialmente las redes, ChatGPT demostró en los casos de estudios posteriores su capacidad de adaptabilidad y aprendizaje para satisfacer las demandas del usuario.

CAPÍTULO 6

6. CONCLUSIONES Y RECOMENDACIONES

El trabajo realizado sobre la incorporación de la inteligencia artificial (IA) en la ingeniería eléctrica, enfocándose en modelos de lenguaje avanzados como ChatGPT, ha revelado el potencial que dichos modelos poseen en campos tradicionales de la ingeniería. Para este proyecto se trabajó únicamente con ChatGPT y Open Distribution System Simulator (OpenDSS), pero abre las puertas a nuevos estudios e investigaciones con otros modelos de IA y simuladores para obtener nuevos y mejores resultados. Este estudio ha mostrado que ChatGPT no solo facilita la generación de datos sintéticos y la modelización de sistemas a partir de parámetros iniciales, sino que también es capaz de abordar desafíos actuales en la gestión de redes de distribución eléctrica. En base a la experiencia y lecciones aprendidas en este proceso, pudimos comprobar que los modelos de lenguaje grande como ChatGPT, tienen una capacidad de resolución de problemas e interpretación de información en proporción a la cantidad de datos que se les proporciona, al igual que con la especificidad de las órdenes suministradas por parte del usuario. La sinergia entre ChatGPT y herramientas especializadas como OpenDSS sugiere una posible integración de la IA en la ingeniería eléctrica, abriendo caminos para la expansión, optimización y corrección de sistemas eléctricos. Este avance demuestra el inmenso valor de aplicar tecnologías modernas, como la IA, en sectores técnicos y tradicionales como la ingeniería eléctrica.

6.1 Conclusiones

- A través de un proceso iterativo, se logró establecer un conjunto de instrucciones específicas, las cuales sirvieron como fundamento para que ChatGPT generara sistemas eléctricos respetando la sintaxis de OpenDSS. Esto implicó una etapa de aprendizaje y ajuste, donde se definieron parámetros clave para asegurar la coherencia y precisión en la generación de los modelos. El uso de ChatGPT para la generación de estos sistemas no solo aporta eficiencia en términos de tiempo y recursos, sino que también introduce una flexibilidad y adaptabilidad que es crucial en el diseño de redes eléctricas modernas.

- Los modelos elaborados por ChatGPT, y validados exitosamente por medio de OpenDSS, demostraron la capacidad de la IA para generar correcta y eficazmente sistemas de distribución eléctrica, semejantes a los encontrados en redes actuales en base a topologías y variables de control. Esta capacidad de simulación y validación no solo aporta una herramienta valiosa para el diseño de nuevas redes, sino que también permite la revisión y optimización de redes existentes, identificando puntos críticos y oportunidades de mejora en términos de eficiencia y fiabilidad.
- ChatGPT demostró ser capaz, no solo de generar datos sintéticos a partir de instrucciones respecto a las generalidades de una red, sino también la expansión y solución de problemas de calidad de energía en dichas redes y en las ya existentes proporcionadas por el usuario. Esto incluye, pero no está limitado a, la identificación y análisis de problemas como bajos niveles de tensión en las redes. La capacidad de ChatGPT para interactuar con el usuario y adaptar las soluciones a situaciones específicas se presenta como un avance significativo en el campo de la ingeniería eléctrica, ofreciendo una herramienta poderosa para la gestión y optimización de redes eléctricas.
- El proyecto ha mostrado la posibilidad de una posible integración de ChatGPT en sistemas de monitoreo y control en tiempo real de redes eléctricas. Su capacidad para procesar grandes volúmenes de datos y generar respuestas rápidas es vital para la toma de decisiones en situaciones críticas.

6.2 Recomendaciones

- Utilizar algoritmos avanzados de inteligencia artificial con capacidades superiores de procesamiento analítico como ChatGPT 4 para la generación de códigos de programación de alta complejidad y estructura intrincada. Este enfoque permitirá no solo optimizar el desarrollo de sistemas complejos, sino también mejorar la precisión y eficacia en la automatización de tareas de ingeniería avanzadas.

- Continuar explorando y profundizando en las funcionalidades y aplicaciones potenciales de ChatGPT, extendiendo su uso a la interacción con plataformas especializadas de simulación. Esta investigación deberá enfocarse en identificar y desarrollar nuevas aplicaciones prácticas de inteligencia artificial en áreas estratégicas, promoviendo así una sinergia entre herramientas lingüísticas avanzadas y soluciones de software específicas para la ingeniería.
- Realizar estudios comparativos sobre las capacidades y rendimiento de distintas plataformas de inteligencia artificial con características similares a ChatGPT. El objetivo de esta investigación debería enfocarse en la evaluación y comparación de la eficiencia y aplicabilidad en contextos de ingeniería eléctrica y otros campos tradicionales, lo cual permitirá identificar las plataformas más adecuadas para diferentes tipos de aplicaciones y escenarios operativos.
- Implementar programas de formación y actualización profesional en IA para ingenieros eléctricos. Dado el rápido avance de la tecnología, es crucial que los profesionales del sector estén constantemente actualizados sobre las últimas tendencias y herramientas en inteligencia artificial. Estos programas podrían incluir cursos, seminarios y talleres que cubran desde los fundamentos de la IA hasta aplicaciones específicas en ingeniería eléctrica, asegurando que los ingenieros estén equipados con las habilidades necesarias para integrar eficazmente estas tecnologías en su trabajo diario.

REFERENCIAS

- [1] P. Razmi, M. Ghaemi Asl, G. Canarella, and A. S. Emami, "Topology identification in distribution system via machine learning algorithms," PLOS ONE, vol. 16, no. 6. Public Library of Science (PLoS), p. e0252436, Jun. 01, 2021. doi: 10.1371/journal.pone.0252436.
- [2] A. Pokahr and L. Braubach, "The active components approach for distributed systems development," International Journal of Parallel, Emergent and Distributed Systems, vol. 28, no. 4. Informa UK Limited, pp. 321–369, Aug. 2013. doi: 10.1080/17445760.2013.785546.
- [3] P. Dehghanian, M. Fotuhi-Firuzabad, S. Bagheri-Shouraki and A. A. Razi Kazemi, "Critical Component Identification in Reliability Centered Asset Management of Power Distribution Systems Via Fuzzy AHP," in IEEE Systems Journal, vol. 6, no. 4, pp. 593-602, Dec. 2012, doi: 10.1109/JSYST.2011.2177134.
- [4] S. K. Nalini Ramakrishna, S. Koziel, D. Karlsson and P. Hilber, "Component ranking and importance indices in the distribution system," 2021 IEEE Madrid PowerTech, Madrid, Spain, 2021, pp. 1-6, doi: 10.1109/PowerTech46648.2021.9494968.
- [5] S. F. Alwash, V. K. Ramachandaramurthy and N. Mithulananthan, "Fault-Location Scheme for Power Distribution System with Distributed Generation," in IEEE Transactions on Power Delivery, vol. 30, no. 3, pp. 1187-1195, June 2015, doi: 10.1109/TPWRD.2014.2372045.
- [6] R. Singh, B. C. Pal and R. B. Vinter, "Measurement Placement in Distribution System State Estimation," in IEEE Transactions on Power Systems, vol. 24, no. 2, pp. 668-675, May 2009, doi: 10.1109/TPWRS.2009.2016457.
- [7] L. Cristaldi, A. Ferrero and S. Salicone, "A distributed system for electric power quality measurement," in IEEE Transactions on Instrumentation and Measurement, vol. 51, no. 4, pp. 776-781, Aug. 2002, doi: 10.1109/TIM.2002.803300.
- [8] H. Wang, Z. Feng, X. Chu and W. Zhang, "A robust measurement placement method in distribution system state estimation," 2014 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Hong Kong, China, 2014, pp. 1-6, doi: 10.1109/APPEEC.2014.7066160.

- [9] R. Dutta, A. Dubey, S. Chakrabarti and A. Sharma, "Comparative Performance Analysis of Optimization Based Distribution System State Estimator with Varying Measurement Uncertainties," 2018 20th National Power Systems Conference (NPSC), Tiruchirappalli, India, 2018, pp. 1-6, doi: 10.1109/NPSC.2018.8771865.
- [10] Y. Kang, X. Zhao, W. Gao, B. Shi, J. Wang and D. Jin, "New Measurement Method of Distributed Power Generation," 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 2022, pp. 714-718, doi: 10.1109/ICPICS55264.2022.9873749.
- [11] C. Mateo et al., "Building Large-Scale U.S. Synthetic Electric Distribution System Models," in IEEE Transactions on Smart Grid, vol. 11, no. 6, pp. 5301-5313, Nov. 2020, doi: 10.1109/TSG.2020.3001495.
- [12] E. Schweitzer, A. Scaglione, A. Monti and G. A. Pagani, "Automated Generation Algorithm for Synthetic Medium Voltage Radial Distribution Systems," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 7, no. 2, pp. 271-284, June 2017, doi: 10.1109/JETCAS.2017.2682934.
- [13] G. Viganò, M. Rossi, C. Michelangeli and D. Moneta, "Creation of the Italian Distribution System Scenario by Using Synthetic Artificial Networks," 2020 AEIT International Annual Conference (AEIT), Catania, Italy, 2020, pp. 1-6, doi: 10.23919/AEIT50178.2020.9241109.
- [14] D. T. Pham and P. T. N. Pham, "Artificial intelligence in engineering," International Journal of Machine Tools and Manufacture, vol. 39, no. 6. Elsevier BV, pp. 937–949, Jun. 1999. doi: 10.1016/s0890-6955(98)00076-5.
- [15] M. Abdullah, A. Madain and Y. Jararweh, "ChatGPT: Fundamentals, Applications and Social Impacts," 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), Milan, Italy, 2022, pp. 1-8, doi: 10.1109/SNAMS58071.2022.10062688.
- [16] J. Moysen, L. Giupponi and J. Manges-Bafalluy, "A machine learning enabled network planning tool," 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 2016, pp. 1-7, doi: 10.1109/PIMRC.2016.7794909.
- [17] J. Qadir, "Engineering Education in the Era of ChatGPT: Promise and Pitfalls of Generative AI for Education," 2023 IEEE Global Engineering Education Conference

(EDUCON), Kuwait, Kuwait, 2023, pp. 1-9, doi: 10.1109/EDUCON54358.2023.10125121.

[18] S. Omran and E. M. F. El Houby, "Prediction of electrical power disturbances using machine learning techniques," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 7. Springer Science and Business Media LLC, pp. 2987–3003, Sep. 04, 2019. doi: 10.1007/s12652-019-01440-w.

[19] "Research on the Application of Artificial Intelligence Technology in Electrical Automation Control," *Journal of Artificial Intelligence Practice*, vol. 6, no. 7. Clausius Scientific Press, Inc., 2023. doi: 10.23977/jaip.2023.060701.

[20] K. Mahmoud, M. Abdel-Nasser, H. Kashef, D. Puig, and M. Lehtonen, "Machine Learning Based Method for Estimating Energy Losses in Large-Scale Unbalanced Distribution Systems with Photovoltaics," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 4. Universidad Internacional de La Rioja, p. 157, 2020. doi: 10.9781/ijimai.2020.08.002.

[21] D. Kalla, and N. Smith, "Study and Analysis of Chat GPT and its Impact on Different Fields of Study," *International Journal of Innovative Science and Research Technology*, vol 8, no. 3, March 2023.

[22] J. Qadir, "Engineering Education in the Era of ChatGPT: Promise and Pitfalls of Generative AI for Education," 2023 IEEE Global Engineering Education Conference (EDUCON), Kuwait, Kuwait, 2023, pp. 1-9, doi: 10.1109/EDUCON54358.2023.10125121.

[23] X. Zhao and X. Zhang, "Artificial Intelligence Applications in Power System," *Proceedings of the 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*. Atlantis Press, 2016. doi: 10.2991/aiie-16.2016.36.

[24] Z. Zhao and J. Mutale, "Simulation of automatic network reconfiguration technology in distribution system by OpenDss," 2016 51st International Universities Power Engineering Conference (UPEC), Coimbra, Portugal, 2016, pp. 1-5, doi: 10.1109/UPEC.2016.8114085.

[25] K. Moy, C. Tae, Y. Wang, G. Henri, N. Bambos and R. Rajagopal, "An OpenAI-OpenDSS framework for reinforcement learning on distribution-level microgrids," 2021 IEEE Power & Energy Society General Meeting (PESGM), Washington, DC, USA, 2021, pp. 1-5, doi: 10.1109/PESGM46819.2021.9638106.

- [26] S. A. Subramanya, M. Parathodiyil and A. N. Nagashree, "Implementation of Peak Demand Reduction on a Distribution Feeder using Python-OpenDSS co-simulation," 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2021, pp. 01-06, doi: 10.1109/CONECCT52877.2021.9622589.
- [27] EPRI, OpenDSS Tutorial. Distribution System Simulator.
- [28] Roger Dugan, What is Unique About OpenDSS?, 2014, p. 1.
- [29] D. Montenegro and R. C. Dugan, "OpenDSS and OpenDSS-PM open source libraries for NI LabVIEW," 2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA). IEEE, May 2017. doi: 10.1109/pepqa.2017.7981639.
- [29] Claro Báez, J.P. (2016). Modelado y simulación de Smart Grid con OpenDSS y Matlab. (Trabajo fin de grado inédito). Universidad de Sevilla, Sevilla.
- [30] Southwire. "ACSR." Southwire.com. [Online]. Available: <https://www.southwire.com/wire-cable/bare-aluminum-overhead-transmission-distribution/acsr/p/10136032>. Accessed on: Dec. 10, 2023.
- [31] Vwcable. "ACSR Overhead Transmission Line 336.4 Mcm ACSR Conductor." es.vwcable.com. [Online]. Available: <https://es.vwcable.com/acsr-overhead-transmission-line-336-4-mcm-acsr-conductor/>. Accessed on: Dec. 10, 2023.
- [32] M. Toledo-Orozco, E. Bravo-Padilla, C. Álvarez-Bel, D. Morales-Jadan, and L. Gonzalez-Morales, "Methodological Evaluation to Integrate Charging Stations for Electric Vehicles in a Tram System Using OpenDSS—A Case Study in Ecuador," Sustainability, vol. 15, no. 8. MDPI AG, p. 6382, Apr. 07, 2023. doi: 10.3390/su15086382.
- [33] J.-J. Zhu, J. Jiang, M. Yang, and Z. J. Ren, "ChatGPT and Environmental Research," Environmental Science & Technology, vol. 57, no. 46. American Chemical Society (ACS), pp. 17667–17670, Mar. 21, 2023. doi: 10.1021/acs.est.3c01818.
- [34] J. Deng y Y. Lin, "The Benefits and Challenges of ChatGPT: An Overview," en Frontiers in Computing and Intelligent Systems, vol. 2, no. 2, 2022.
- [35] N. L. Rane, "Contribution of ChatGPT and Other Generative Artificial Intelligence (AI) in Renewable and Sustainable Energy," Oct. 09, 2023.
- [36] R. Li, C. Pu, F. Fan, J. Tao, y Y. Xiang, "Leveraging ChatGPT for Power System Programming Tasks," 2023.

[37] A. Haleem, M. Javaid, and R. P. Singh, "An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges," *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 2, no. 4. Elsevier BV, p. 100089, Oct. 2022. doi: 10.1016/j.tbench.2023.100089.

[38] R. Li, C. Pu, J. Tao, C. Li, F. Fan, Y. Xiang, y S. Chen, "LLM-based Frameworks for Power Engineering from Routine to Novel Tasks," 8 de mayo de 2023. [Online]. Disponible: arXiv:2305.11202