



UNIVERSIDAD TÉCNICA DE COTOPAXI
DIRECCIÓN DE POSGRADO

MAESTRÍA EN ELECTRICIDAD

MODALIDAD: PROYECTO DE INVESTIGACIÓN

Título:

DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA QUE PERMITA DETERMINAR LOS TIEMPOS DE OPERACIÓN, CURVAS DE OPERACIÓN Y COORDINACIÓN DE LOS RELÉS DE SOBRECORRIENTE EN LAS LÍNEAS DE TRANSMISIÓN

Trabajo de titulación previo a la obtención del título de magister en Electricidad
mención sistemas eléctricos de potencia

AUTOR:

Peralvo Peralvo Henry Alexander

TUTOR:

Ing. Proaño Maldonado Xavier Alfonso MSc.

LATACUNGA –ECUADOR

2023

AVAL DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación **“DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA QUE PERMITA DETERMINAR LOS TIEMPOS DE OPERACIÓN, CURVAS DE OPERACIÓN Y COORDINACIÓN DE LOS RELÉS DE SOBRECORRIENTE EN LAS LÍNEAS DE TRANSMISIÓN”** presentado por Peralvo Peralvo Henry Alexander para optar por el título Magíster en Electricidad Mención Sistemas Eléctricos de Potencia.

CERTIFICO

Que dicho trabajo de investigación ha sido revisado en todas sus partes y se considera que reúne los requisitos y méritos suficientes para ser sometido a la presentación para la valoración por parte del Tribunal de Lectores que se designe y su exposición y defensa pública.

Latacunga, 06 de enero 2023



MSc.Ing. Xavier Alfonso Maldonado Proaño.

CC.: 0502656424

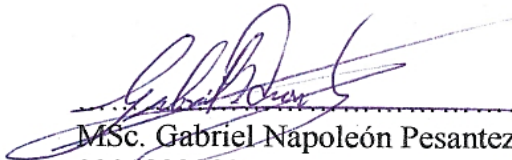
AVAL DEL TRIBUNAL

El trabajo de Titulación: “Desarrollo de una herramienta informática que permita determinar los tiempos de operación, curvas de operación y coordinación de los relés de sobrecorriente en las líneas de transmisión”, ha sido revisado, aprobado y autorizado su impresión y empastado, previo a la obtención del título de Magíster Electricidad Mención Sistemas Eléctricos de Potencia ; el presente trabajo reúne los requisitos de fondo y forma para que el estudiante pueda presentarse a la exposición y defensa.

Latacunga, marzo, 06, 2023



.....
MSc. Luis Rolando Cruz Panchi
0502595176
Presidente del tribunal



.....
MSc. Gabriel Napoleón Pesantez Palacios
0301893889
Lector 2



.....
Msc. Carlos Francisco Pacheco Mena
0503072902
Lector 3

DEDICATORIA

El presente proyecto de investigación es dedicado primeramente a Dios a mis Santos devotos por permitirme salir adelante y nunca dejarme vencer por los obstáculos que se me presentaron en la vida.

A mi ejemplo de vida como son mis padres Narciza Peralvo y Enrique Peralvo, mi hermana Daniela Peralvo, mi novia Daysi por ser las personas más importantes en mi vida y por siempre apoyarme en todas las decisiones que he tomado, y estar conmigo en los momentos más difíciles de mi vida profesional.

A toda mi familia mis tíos, primos, abuelitos, todos los de la casa por el apoyo brindado en mi vida diaria y profesional.

Henry Peralvo

AGRADECIMIENTO

En primer lugar, estrechar mis profundos agradecimientos a mis padres que gracias a su trabajo, dedicación y esfuerzo hicieron de mí un gran profesional, e hicieron que nunca me faltara nada en mis estudios y que siempre estuvieron ahí aconsejándome, apoyándome para poder salir adelante.

Agradecer a Dios, mi familia, mi novia, mis amigos por nunca dejarme vencer y mantener siempre en mí la fe en salir adelante en mi vida profesional, y que siempre me han apoyado en este difícil camino.

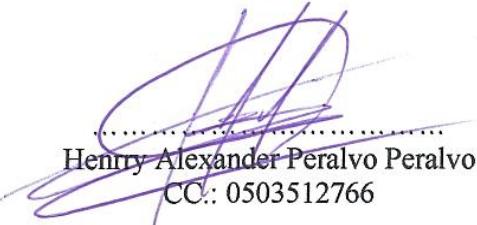
Finalmente agradecer a la Universidad Técnica de Cotopaxi por la preparación profesional, y brindarnos todos los conocimientos en cada uno de los módulos para podernos superar cada día más como profesionales, de manera especial un agradecimiento a mi tutor Xavier Alfonso Proaño Maldonado. MSc.

Henry Peralvo

RESPONSABILIDAD DE AUTORÍA

Quien suscribe, declara que asume la autoría de los contenidos y los resultados obtenidos en el presente trabajo de titulación.

Latacunga, marzo, 06, 2023



Henry Alexander Peralvo Peralvo
CC.: 0503512766

RENUNCIA DE DERECHOS

Quien suscribe, cede los derechos de autoría intelectual total y/o parcial del presente trabajo de titulación a la Universidad Técnica de Cotopaxi.

Latacunga, marzo, 06, 2023


.....
Henry Alexander Peralvo Peralvo
CC.: 0503512766

AVAL DEL PRESIDENTE DEL TRIBUNAL

Quien suscribe, declara que el presente Trabajo de Titulación: “Desarrollo de una herramienta informática que permita determinar los tiempos de operación, curvas de operación y coordinación de los relés de sobrecorriente en las líneas de transmisión” contiene las correcciones a las observaciones realizadas por los lectores en sesión científica del tribunal.

Latacunga, marzo, 06, 2023



.....
MSc. Luis Rolando Cruz Panchi
0502595176

UNIVERSIDAD TÉCNICA DE COTOPAXI

DIRECCIÓN DE POSGRADOS

MAESTRÍA EN ELECTRICIDAD

MENCIÓN SISTEMAS ELÉCTRICOS DE POTENCIA

Título:

“DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA QUE PERMITA DETERMINAR LOS TIEMPOS DE OPERACIÓN, CURVAS DE OPERACIÓN Y COORDINACIÓN DE LOS RELÉS DE SOBRECORRIENTE EN LAS LÍNEAS DE TRANSMISIÓN”

Autor:

Peralvo Peralvo Henry Alexander

Tutor:

Proaño Maldonado Xavier Alfonso. MSc.

RESUMEN

El presente trabajo de investigación fue enfocado principalmente al desarrollo de una herramienta informática que permitió determinar los tiempos de operación, curvas de operación y coordinación de los relés de sobrecorriente en las líneas de transmisión, la herramienta es una interfaz gráfica realizado en el Software Python con el intérprete Visual Studio Code comunicado directamente con el software DIGSILENT PowerFactory, se desarrolló una ventana de inicio en donde el usuario ingresara la dirección y la versión del programa de caso de estudio, posteriormente se verifico la comunicación entre estos con la lectura de componentes y datos de nuestro sistema eléctrico. Mediante botones dispuestos de forma uniforme en la segunda pantalla se ejecutó comandos de DIGSILENT PowerFactory como flujos de carga y análisis de corto circuito en las barras de nuestro caso de estudio, también se logró una lectura de las corrientes de pickup y dial de los relés de sobrecorriente. Con esta comunicación la herramienta informática genera un reporte en una hoja de cálculo en Excel de todos los datos, graficas requeridos para el análisis de nuestro caso de estudio. El cálculo del error relativo entre la interface y Powerfactory, obteniendo un valor máximo de 0.91% y un mínimo de 0%.

PALABRAS CLAVE: DIGSILENT PowerFactory; Python; Visual Studio Code; relés de sobrecorriente; tiempos de operación; curvas de operación; coordinación de protecciones.

UNIVERSIDAD TÉCNICA DE COTOPAXI

DIRECCIÓN DE POSGRADOS

MAESTRÍA EN ELECTRICIDAD

MENCIÓN SISTEMAS ELÉCTRICOS DE POTENCIA

Title:

“DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA QUE PERMITA DETERMINAR LOS TIEMPOS DE OPERACIÓN, CURVAS DE OPERACIÓN Y COORDINACIÓN DE LOS RELÉS DE SOBRECORRIENTE EN LAS LÍNEAS DE TRANSMISIÓN”

Autor:

Peralvo Peralvo Henry Alexander

Tutor:

Proaño Maldonado Xavier Alfonso. MSc.

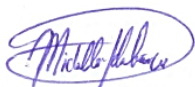
ABSTRACT

The present research work was mainly focused on the development of a computer tool that allowed determining the operating times, operating curves and coordination of overcurrent relays in transmission lines, the tool is a graphical interface made in Python software with the Visual Studio Code interpreter communicated directly with the DIgSILENT PowerFactory, a start window was developed where the user entered the address and version of the case study program, then the communication between these was verified with the reading of components and data from our electrical system. By means of buttons arranged uniformly on the second screen, DIgSILENT PowerFactory commands were executed such as load flows and short circuit analysis in the busbars of our case study, also a reading of the pickup currents and dial of the overcurrent relays was achieved. With this communication with the software tool, a report was generated in an Excel spreadsheet with all the data and graphs required for the analysis of our case study. The calculation of the relative error between the interface and Powerfactory, obtaining a maximum value of 0.91% and a minimum of 0%.

KEYWORD: DIgSILENT PowerFactory; Python; Visual Studio Code; overcurrent relays; operating times; operating curves; coordination of protections.

Yo, Cynthia Michelle Pilatuña Espinoza con cedula de identidad número: 1805243548 Licenciada en: Ciencias de la Educación Mención Ingles. Con número de registro de la SENECYT: 1010-2021-2288016; **CERTIFICO** haber revisado y aprobado la traducción al idioma de ingles del resumen del trabajo de investigación con el título: **“DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA QUE PERMITA DETERMINAR LOS TIEMPOS DE OPERACIÓN, CURVAS DE OPERACIÓN Y COORDINACIÓN DE LOS RELÉS DE SOBRECORRIENTE EN LAS LÍNEAS DE TRANSMISIÓN”** de: Henry Alexander Peralvo Peralvo, aspirante a magister en Electricidad mención Sistemas Eléctricos de Potencia.

Latacunga, 06 de enero del 2023



.....
Cynthia Michelle Pilatuña Espinoza

CI: 1805243548

ÍNDICE DE CONTENIDO

AVAL DEL TUTOR.....	II
AVAL DEL TRIBUNAL	III
DEDICATORIA	IV
AGRADECIMIENTO	V
RESPONSABILIDAD DE AUTORÍA.....	VI
RENUNCIA DE DERECHOS.....	VII
AVAL DEL PRESIDENTE DEL TRIBUNAL.....	VIII
RESUMEN.....	IX
ABSTRACT	X
ÍNDICE DE CONTENIDO	XI
INTRODUCCIÓN	1
Antecedentes:	2
Planteamiento del problema:	2
Formulación del problema	3
Objetivo General.....	3
Objetivos Específicos:.....	3
Justificación.....	4
Hipótesis.....	5
1 CAPITULO I. FUNDAMENTACIÓN TEÓRICA	6
1.1 Antecedentes de la investigación.	6
1.2 Fundamentación Teórica	7
1.2.1 Protecciones Eléctricas.....	7
1.2.2 Características de las Protecciones Eléctricas.....	7
1.2.3 Tipos de Protecciones Eléctricas.....	10
1.2.4 Protección contra Cortocircuitos.....	10
1.2.5 Protección contra Sobrecargas	11
1.2.6 Relés de Protección	11
1.2.7 Tipos de relés de protección según el tipo de protección	11
1.2.8 Protecciones de las sobrecorrientes.....	12
1.2.9 Característica de operación de las protecciones de sobrecorriente	12
1.2.10 Relés de Sobrecorriente	13

1.2.11	Relés de sobrecorriente por tiempo de operación.....	13
1.2.12	Relés de sobrecorriente por la característica de operación	13
1.2.13	Modelo matemático de los relés de sobrecorriente	16
1.2.14	Criterios de coordinación de protecciones 50/51	17
1.2.15	Cálculo de cortocircuitos según normas ANSI/IEEE	18
1.2.16	Softwares de simulación de sistemas eléctricos de potencia.....	22
1.2.17	Software DIgSILENT PowerFactory.....	22
1.2.18	Compatibilidad de DIgSILENT PowerFactory	23
1.2.19	Software Python	23
1.2.20	Compatibilidad de Python.....	23
1.2.21	Visual Studio Code	24
2	CAPÍTULO II. PROPUESTA	26
2.1	Título del proyecto	26
2.2	Objetivo del proyecto	26
2.3	Descripción de la propuesta.....	26
2.4	Metodología o procedimientos empleados para el cumplimiento de los objetivos planteados.....	27
2.4.1	Comunicación Python y DIgSILENT PowerFactory	27
2.4.2	Diagramas de flujo de funcionamiento.....	28
2.4.3	Comunicación DIgSILENT PowerFactory y Python.....	35
2.4.4	Desarrollo de la interface grafica	35
2.4.5	Comandos de DIgSILENT PowerFactory	37
2.4.6	Obtener valores desde DIgSILENT PowerFactory	39
2.4.7	Proceso para generar las gráficas de los relés de sobrecorriente	43
2.5	Conclusiones Capitulo II	49
3	CAPITULO III. APLICACIÓN Y/O VALIDACIÓN DE LA PROPUESTA	50
3.1	Análisis de Resultados.....	50
3.2	Descripción del primer caso de estudio.....	50
3.2.1	39 Bus New England System.....	50
3.2.2	Análisis de Flujo de Potencia.....	51
3.2.3	Análisis de Corto Circuito	53

3.2.4	Descripción del segundo caso de estudio	54
3.2.5	Análisis de Flujo de Potencia.....	56
3.3	Conclusiones Capitulo III.....	66
4	CONCLUSIONES	67
5	RECOMENDACIONES	68
6	BIBLIOGRAFÍA	69
7	ANEXOS.....	73

ÍNDICE DE FIGURAS

Figura 1	Fundamento de Cortocircuito	11
Figura 2.	Curvas características de operación Tiempo-Corriente.	12
Figura 3.	Relé de corriente definida	14
Figura 4.	Relé de corriente definida y tiempo definido	15
Figura 5.	Relés de tiempo inverso	15
Figura 6.	Diferentes tipos de relés de tiempo inverso.....	16
Figura 7.	Coordinación de los relés de sobrecorriente.....	18
Figura 8.	Análisis de un caso de estudio.....	27
Figura 9.	Proceso de comunicación de la herramienta informática.....	28
Figura 10.	Primera ventana de la herramienta informática.....	29
Figura 11.	Ingreso del proyecto en la herramienta informática.....	30
Figura 12.	Análisis de un flujo de potencia.....	31
Figura 13.	Análisis de corto circuito en una barra.....	31
Figura 14.	Obtener los relés de sobrecorriente.....	32
Figura 15.	Obtener las barras existentes en el proyecto.....	33
Figura 16.	Obtener las constantes de las ecuaciones de los relés.....	34
Figura 17.	Ventana principal de la herramienta informática.....	36
Figura 18.	Ventana de coordinación de protecciones.....	37
Figura 19.	Configuración de un análisis de flujo de potencia.....	38
Figura 20.	Configuración de un análisis de corto circuito.....	39
Figura 21.	Listado de proyectos en PowerFactory.....	39
Figura 22.	Listado de relés en PowerFactory.....	41
Figura 23.	Lectura de valores desde la herramienta informática.....	42
Figura 24.	Librerías de relés de sobrecorriente PowerFactory.....	43
Figura 25.	Parámetros del relé de fase	43
Figura 26.	Parámetros del relé de fase	44
Figura 27.	Parámetros de comportamiento del relé de fase	45
Figura 28.	Graficas del relé de fase	46
Figura 29.	Datos de las ecuaciones para cada comportamiento.....	48
Figura 30.	Curvas del comportamiento del relé ANSI/IEEE.....	49
Figura 31.	Análisis del circuito de potencia de 39 barras.....	51

Figura 32. Error de medición en el voltaje.....	51
Figura 33. Error de medición en el ángulo de voltaje.....	52
Figura 34. Error de medición en la magnitud de voltaje.....	52
Figura 35. Error de medición la potencia de cortocircuito inicial.	53
Figura 36. Error de medición la potencia de cortocircuito inicial.	53
Figura 37. Error de medición la potencia de cortocircuito inicial.	54
Figura 38. Caso de estudio original con 3 líneas de transmisión.	54
Figura 39. Caso de estudio preparado para el análisis.	55
Figura 40. Lista de barras con nombres modificados.	55
Figura 41. Lista de relés existentes en el caso de estudio.	56
Figura 42. Tipos de relés existentes en el caso de estudio.....	56
Figura 43. Análisis de flujo de potencia en DIgSILENT PowerFactory.	57
Figura 44. Análisis de flujo de potencia en DIgSILENT PowerFactory.	57
Figura 45. Representación del error relativo obtenido.....	59
Figura 46. Análisis de Corto Circuito en la Barra C.....	60
Figura 47. Representación del error relativo obtenido.....	61
Figura 48. Comportamiento de los relés de sobrecorriente ANSI/IEEE.	62
Figura 49. Datos del Relay_Load1 en la línea B23	63
Figura 50. Configuración del TC para el Relay_Load1.....	63
Figura 51. Datos del Relay_L23 en la línea B23.....	64
Figura 52. Configuración del TC para el Relay_L23	64
Figura 53. Coordinación 1 Relay_Load1 vs Relay_L23.....	65
Figura 54. Coordinación 2 Relay_L24 vs Relay_Load2.....	65
Figura 55. Coordinación 3 entre Relay_L25 vs Relay_Load3.....	66
Figura 56. Pantalla principal de instalación PowerFactory.....	73
Figura 57. Instalación de completos de PowerFactory.	74
Figura 58. Instalación de licencias de servidor.	74
Figura 59. Selección del idioma en PowerFactory.	75
Figura 60. Software instalado de PowerFactory.....	75
Figura 61. Instalador de Python 3.3.2.....	76
Figura 62. Proceso de instalación terminado.	77
Figura 63. Acceso de Propiedades del Sistema.	77

Figura 64. Variable de entorno.....	78
Figura 65. Enlace de PATH de Python.....	78
Figura 66. Inicio de la instalación de Visual Studio Code.....	79
Figura 67. Agregar PATH en Visual Studio Code.	80
Figura 68. Instalación terminada con éxito de Visual Studio Code.	80

ÍNDICE DE TABLAS

Tabla 1. Constantes de forma para ecuación exponencial según normas IEC.....	17
Tabla 2. Instrucciones para establecer comunicación entre los programas.	35
Tabla 3. Instrucciones para crear una ventana.....	35
Tabla 4. Instrucciones activar un flujo de carga.	37
Tabla 5. Instrucciones activar un corto circuito.	38
Tabla 6. Código para activar un proyecto.	40
Tabla 7. Lectura de relés desde la interface.	41
Tabla 8. Lectura de barras desde la interface.	42
Tabla 9. Ecuaciones de comportamiento de los relés de sobrecorriente.....	46
Tabla 10. Crear un archivo en Excel.....	47
Tabla 11. Obtener las variables de las constantes a_1, a_2, a_3 y a_4	47
Tabla 12. Obtener las variables de las constantes a_1, a_2, a_3 y a_4	48
Tabla 13. Valores de flujo de potencia en DIGSILENT PowerFactory.	58
Tabla 14. Valores de flujo de potencia en la Herramienta informática.	58
Tabla 15. Error de Flujo de Potencia.	59
Tabla 16. Análisis de cortocircuito en PowerFactory.	60
Tabla 17. Análisis de cortocircuito en herramienta informática.	60
Tabla 18. Error de Cortocircuito.....	61
Tabla 19. Valores de sobrecorriente	61

INTRODUCCIÓN

Los sistemas eléctricos de potencia se encuentran integrados por tres componentes principales como son : generación, distribución y transmisión, la generación de la energía eléctrica se realiza en las centrales que son locales en donde se efectúan las transformaciones de energía eólica, hidráulica, térmica, solar en energía eléctrica a través de generadores en trabajo con las turbinas, en la transmisión se encarga de sustentar grandes cantidades de energía de los centros de producción a la parte de consumo, finalmente en la distribución distribuyen de manera eficiente la energía eléctrica a los usuarios finales. Las líneas de transmisión son uno de los componentes importantes de los SEP que deben siempre estar energizadas y brindar el servicio a los usuarios, por ello es esencial detectar una falla a tiempo, con una precisión optima que tiene una respuesta inmediata para poder restablecer el suministro de energía y así mejorar la confiabilidad del sistema eléctrico de potencia [1].

Este trabajo de investigación está centrado en el desarrollo de una herramienta informática para determinar los tiempos de operación, coordinación y graficas de las curvas de operación de los relés de sobrecorriente 50/51 ante la presencia de corrientes de cortocircuitos en las líneas de transmisión en los SEP (sistema eléctrico de potencia).

En el presente trabajo de investigación se procede a desarrollar una herramienta informática elaborada en el software Python mediante el intérprete Visual Studio Code, el mismo que se comunicara con el software DIgSILENT PowerFactory.

En el software DIgSILENT PowerFactory se cargará los diferentes diseños de sistemas eléctricos que servirán como casos de estudio para comparar datos del software como flujos de carga, cortocircuitos en cada una de las barras, operación de los relés y coordinación, con los datos que presenta en la interfaz de nuestra herramienta informática. Finalmente, nuestra herramienta permitirá generar un reporte con los datos necesarios para el análisis de los relés de sobrecorriente.

Antecedentes:

Cuando se descubrió la energía eléctrica en la humanidad, este servicio se convirtió en un papel muy importante en el desarrollo de la humanidad, el medio por el cual llega la energía a los consumidores se requiere de estructuras cada vez más complejas llamadas sistemas eléctricos de potencia, por lo que es necesario mencionar que el presente trabajo de investigación se lo realizara citando diferentes fuentes bibliográficas empleadas en investigaciones similares como se detallara en los siguientes párrafos, estos deben contener términos y conceptos de acuerdo al tema de investigación [1].

En [2] el autor menciona la coordinación de relés de sobrecorriente direccionales en un sistema en anillo se formula como un problema de optimización, considerando que los tiempos totales de funcionamiento de los relés deben minimizarse manteniendo al mismo tiempo su coordinación. Como técnica de solución emplea un algoritmo evolutivo.

En [3] el autor menciona un método generalizado para evaluar la pérdida de la coordinación entre los relés de sobrecorriente en función de los límites de penetración de GD, considerando su tamaño, ubicación y tecnología. El estudio de coordinación y construcción se realiza para diferentes esquemas de protección. Para el sistema en estudio, se determina una corriente mínima de cortocircuito en donde la pérdida de coordinación puede ocurrir entre todas las protecciones (fusibles). Con este estudio se garantiza que incluso con el aumento de la corriente de cortocircuito, al mover el punto de la intersección de las curvas principales y de respaldo, se mantenga un adecuado esquema de protección.

Planteamiento del problema:

Los avances tecnológicos actualmente progresan a un ritmo formidable, por consecuencia gracias a estas investigaciones y descubrimientos la cantidad de información que se puede conseguir sobre los sistemas eléctricos de potencia es muy vasta, este tema puede abarcar también varios objetos de estudios como es la coordinación y análisis de funcionamiento de protecciones eléctricas en las líneas

de transmisión, como son los diferentes relés de sobrecorriente, para el estudio de estos temas los programas de posgrado realizan diferentes investigaciones a profundidad.

Formulación del problema

En la actualidad existe gran cantidad de softwares para el diseño y análisis de sistemas eléctricos de potencia, que se requiere de muchos estudios para poder analizar los datos, generar gráficas, reportes de los relés de sobrecorrientes, sobre todo en los sistemas eléctricos de potencia que contengan gran cantidad de elementos eléctricos.

Por este motivo se va desarrollar una herramienta informática que permita extraer los datos del software DIgSILENT PowerFactory y generar un reporte donde se presente los datos de operación de los relés de sobrecorriente de los diferentes sistemas eléctricos diseñados.

Objetivo General

Desarrollar una herramienta informática mediante la comunicación con un software de diseño de sistemas eléctricos de potencia para determinar los tiempos de operación, curvas de operación y coordinación de los relés de sobrecorriente en las líneas de transmisión.

Objetivos Específicos:

- Realizar una investigación bibliográfica sobre los relés de sobrecorriente y las formas de comunicación que se pueda realizar con los diferentes softwares.
- Desarrollar una herramienta informática que permita extraer datos del software DIgSILENT PowerFactory, para generar un reporte de funcionamiento de los relés de sobrecorriente.
- Validar la herramienta informática con un caso de estudio para determinar el error producido en la extracción de los datos de tiempos de operación, curvas de operación, coordinación de los relés de sobrecorriente, fallas y flujos de carga en cada una de las barras.

Sistemas de tareas en relación a los objetivos específicos

Objetivos específicos	Actividad (tareas)	Resultado de la actividad	Descripción de la actividad (técnicas e instrumentos)
1	<p>Realizar una investigación bibliográfica de los relés de sobrecorriente.</p> <p>Analizar las formas de comunicación que se pueda realizar con los diferentes softwares.</p>	<p>Información de la operación, coordinación y tipos de relés de sobrecorriente en los sistemas eléctricos de potencia.</p> <p>La comunicación con los diferentes de softwares de análisis de SEP, es posible mediante diferentes procesos de programación.</p>	<p>Investigación bibliográfica de fuentes confiables.</p> <p>Análisis detallado los diferentes temas que componen nuestro trabajo de titulación.</p> <p>Elaboración del 1er capítulo del trabajo de investigación (marco teórico).</p>
2	<p>Realizar una herramienta informática mediante una programación avanzada en Python con el intérprete Visual Studio Code, que comunicará el software PowerFactory con la misma.</p>	<p>Una herramienta informática que comunica el software de diseño de sistemas eléctricos para extraer los diferentes datos para el análisis.</p>	<p>Elaboración de un algoritmo de proceso para el diseño de la herramienta informática mediante el análisis de todas las variables.</p> <p>Programación avanzada en Python para comunicar PowerFactory con nuestra herramienta informática.</p>
3	<p>Validar la herramienta informática con un caso de estudio.</p>	<p>Generar un reporte en nuestra herramienta informática de un caso de estudio creado el Software PowerFactory y verificar los tiempos de operación, las curvas de operación, coordinación de los relés de sobrecorriente, fallas y flujos en cada una de las barras.</p>	<p>Verificación del funcionamiento de nuestra herramienta informática variando diferentes parámetros.</p> <p>Generar un caso de estudio para analizar los datos de comportamiento de los relés de sobrecorrientes cuando se presentan cualquier tipo de falla en las líneas de transmisión.</p> <p>Crear un reporte de manera automática con la ayuda de nuestra herramienta informática.</p>

Justificación

Con el empleo de este proyecto de investigación se realizará una programación avanzada en Python con el intérprete Visual Studio Code, que comunicará el software DIgSILENT PowerFactory con la herramienta informática, y con los datos

obtenidos realizar las curvas de operación, coordinación y el reporte de los relés de sobrecorriente en función de las fallas que se presentan en los sistemas eléctricos de potencia.

Hipótesis

El desarrollo de una herramienta informática permitirá la extracción de datos de DIGSILENT PowerFactory para generar reportes de funcionamiento de los relés de sobrecorriente en los sistemas eléctricos de potencia.

1 CAPITULO I. FUNDAMENTACIÓN TEÓRICA

1.1 Antecedentes de la investigación.

En los sistemas eléctricos de potencia (SEP) las diferentes fallas que se presentan en las líneas de transmisión son inevitables, estas diferentes fallas se presentan por diversas causas, es decir; no solo por el desgaste natural del aislamiento de los elementos del sistema eléctrico de potencia que se encuentra operando (aisladores, generadores, transformadores, cables, etc.) sino también por la mala instalación y manipulación realizada por los operadores de las diferentes centrales de energía [4].

Las protecciones como son los relés de sobrecorriente, sus principios de funcionamiento, y características de temporización, cuando la falla es grave, la protección debe despejarla inmediatamente, verificando que el sistema eléctrico de potencia no pierda estabilidad, los métodos de protección que ayudan a combatir ciertos defectos que se presentan en las líneas de transmisión, a conocer el tipo de falla y por lo tanto a conocer las consecuencias que las fallas producen en el sistema [4].

La coordinación de protecciones es importante en un sistema eléctrico de potencia ya que garantiza la protección de todos los elementos, equipos del sistema eléctrico debido a que en la gran mayoría está expuesto a fallas y cortocircuitos, por lo que se requiere que exista una correcta coordinación con el menos porcentaje de error o de tiempo y así poder garantizar la selectividad, confiabilidad y sensibilidad ante una situación [1].

Para los diferentes análisis de sistemas eléctricos de potencia existen varios programas que ayudan al diseño, estudio y análisis completo de los mismos, de forma similar existen varias herramientas de programación que pueden comunicarse con estos softwares y permiten la extracción de datos con el fin de generar reportes para los análisis posteriores.

1.2 Fundamentación Teórica

1.2.1 Protecciones Eléctricas

Los diferentes elementos de protecciones eléctricas detectan condiciones anormales de las líneas de transmisión en los SEP, estos automáticamente se activan y detectan las fallas para devolver el funcionamiento del sistema lo antes posible a su estado normal. El tiempo de reacción de estos equipos de protección a la respuesta de una falla es un factor importante en los sistemas de protección en las líneas de transmisión. La intervención humana en el funcionamiento de estos equipos de protección no es posible, ya que por cuestiones de tiempo de respuesta a las fallas debe ser automática, rápida y debe causar una mínima cantidad de desconexión de energía en el sistema de potencia [5].

1.2.2 Características de las Protecciones Eléctricas

Para las diferentes funcionalidades en los sistemas eléctricos de potencia las protecciones eléctricas deben cumplir las siguientes propiedades:

a) Confiabilidad

La confiabilidad en las líneas de transmisión es la capacidad que tiene la protección eléctrica de detectar cualquier falla que se produzca en la zona de la red y está asignada bajo condiciones de carga mínima [6].

La protección eléctrica debe saber detectar indistintamente las situaciones de falla de aquellas que no lo son. En un sistema de protecciones eléctricas tenga esta característica es necesario:

- Crear para cada tipo de protecciones eléctricas las magnitudes mínimas (corriente y/o voltaje) las mismas que son necesarias para distinguir las diferentes situaciones de falla de las situaciones normales de operación de los sistemas.
- Establecer para cada magnitud necesaria, las condiciones límite que separan las diferentes situaciones de falla de las condiciones normales de operación [6].

Al mencionar las condiciones de límite se habla de un argumento más amplio que el de valores límite ya que en ciertos casos el conocer solo el valor de una magnitud no es suficiente para determinar si es el resultado de una incidencia normal de la explotación del sistema, o de una situación anormal de funcionamiento de los sistemas [7].

Dentro del caso de las protecciones eléctricas de sobrecorriente el coeficiente de sensibilidad se calcula de la siguiente manera:

$$K_S = \frac{I_{cc\ min}}{I_{ajuste}} \quad \mathbf{Ec. (1)}$$

Donde:

K_S = Coeficiente de sensibilidad

$I_{cc\ mín}$ = Corriente mínima de Cortocircuito

I_{ajuste} = Valor de ajuste de la corriente de protección

En la Ec. (1) el coeficiente debe ser mayor que uno, solo cuando la protección es sensible, debido a que, el menor valor de la magnitud de la corriente de cortocircuito supera el ajuste en la misma establecida [8].

b) Velocidad

La velocidad dentro de los sistemas sirve para detectar una falla la protección debe actuar con tal rapidez que debe despejarla lo más rápido posible. Cuando la rapidez sea mayor, menor será la pérdida de estabilidad del sistema eléctricos de potencia y los costos de mantenimiento del sistema.

La rapidez con la cual actúa la protección está relacionada a quemarropa con la tecnología empleada en la locución y la aceleración de timbre y cuidado de los interruptores automotrices socios a las antaño citadas [9].

Las protecciones eléctricas pueden operar de manera: instantánea, con tiempo dependiente.

c) Selectividad

Es la adecuación para predisponer para qué estipulaciones se requiere la operación de un relé de sobrecorriente en función de los parámetros de configuración inicial, y aquellas para las que no debe intervenir u especular de forma temporizada [10].

Es de nota repercusión que el relé de sobrecorriente actúe en el tiempo que le corresponda, así como incluso que no actúe cuando no debe; en otras palabras, si la falla en el sistema eléctrico de potencia está internamente de la franja vigilada por el relé, éste debe sobrevenir el precepto de estrenar los interruptores que aislen el contorno en la falla [11].

Caso contrario, si la falla se produce excepto la zona vigilada, el relé debe asentir que sean otras las protecciones las que actúen para despejar la falla, ya que, si dado el caso el relé llega a desempeñarse, dejará sin favor a un gran número de circuitos, esto provoca un descenso del sistema, el mismo que es inculco [12].

En los diferentes casos en que las protecciones son expuestas a fallas ocurridas fuera de su zona de vigilancia, la selectividad puede ocurrir, mediante un adecuado ajuste de tiempos y condiciones de actuación en coordinación con el resto de protecciones relacionadas, por lo general se puede tener estos tipos de selectividad:

- **Selectividad cronométrica:** Principalmente se encarga de retrasar temporalmente la actuación de las protecciones eléctricas que se encuentran aguas arriba de la falla de operación [12].
- **Selectividad amperimétrica:** Las diferentes protecciones actúan cuando la corriente que circula por ella es superior a la corriente de falla que ha sido calculada previamente [12].
- **Selectividad lógica:** El sistema eléctrico de potencia requiere de una transferencia de señal entre los relés de protección de los interruptores automáticos de los diferentes niveles de una distribución radial.

d) Fiabilidad

Cuando una protección es fiable se encuentra funcionando correctamente y cumpliendo todos sus objetivos. Es decir, la protección debe ser segura y efectiva ante cualquier evento o falla que se produzca [9].

- **Efectividad:** Esta función garantiza que la protección actúa sólo en situación de falla.
- **Seguridad:** Esta función se caracteriza que garantiza que la protección no actúe ante causas extrañas que se presenten, es decir; evitará actuaciones incorrectas en el sistema.

Para que la protección eléctrica actúe es indispensable que todas las etapas que componen el despeje de falla sean cumplidas eficazmente [13].

La actuación de estos sistemas se vuelve incorrecta debido a: ajuste incorrecto, deterioro del servicio, diseño inadecuado del sistema eléctrico de potencia, etc [13].

1.2.3 Tipos de Protecciones Eléctricas

Las protecciones eléctricas protegen a los circuitos o líneas de transmisión de cualquier tipo de fallas o sobrecargas de tensión o corriente, estos también permiten al usuario evitar accidentes con la energía eléctrica, y también evita apagones.

1.2.4 Protección contra Cortocircuitos

Los cortocircuitos es la unión de dos conductores con una discrepancia de tensión entre sí, que no exista ninguna impedancia eléctrica entre ellos como se muestra en la Figura 1. El cortocircuito según nos menciona la ley de Ohm, la impedancia se aproxima a cero y esto permite que la intensidad tienda al infinito, estos se exponen a achaques a los pilotos y máquinas debido que la copia de calor difundido por este fenómeno puede dañarlos.

En la práctica, la fuerza generada al término de un cortocircuito, siempre queda empañada por la correa de los orientadores que, pero es de un ánimo asaz bajo, en absoluto es cero.

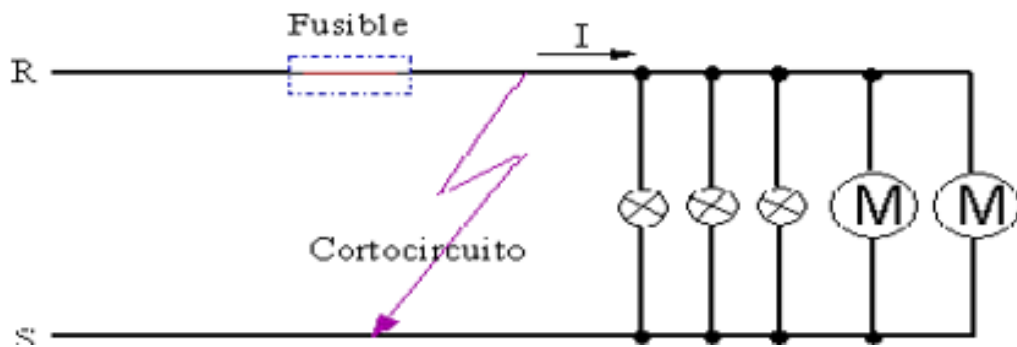


Figura 1 Fundamento de Cortocircuito

Fuente: [14]

Al momento de diseñar o armar un circuito eléctrico se debe tomar en cuenta colocar un dispositivo de protección, de acuerdo con la intensidad total para la que este hecha nuestro circuito [14].

1.2.5 Protección contra Sobrecargas

En sistema eléctrico la sobrecarga es el exceso de intensidad en un circuito eléctrico, debido a un defecto de aislamiento o una cantidad de carga nominal superior. Las sobrecargas deben tener su equipo de protección, ya que pueden dar lugar a la destrucción total de los aislamientos de una red eléctrica o de una máquina [15].

1.2.6 Relés de Protección

Es un elemento que capta los cambios bruscos de señal de voltaje o de corriente, si la dimensión de este tipo de señal que capta el relé está por fuera de un rango establecido, el relé de protección operará, este abrirá o cerrará los contactos para evitar cualquier accidente en las líneas de transmisión de los sistemas eléctricos de potencia. En conclusión, estos detectan los defectos en líneas y equipos inmediatamente detienen las operaciones que están afectando el funcionamiento de nuestro sistema eléctrico de potencia [16].

1.2.7 Tipos de relés de protección según el tipo de protección

Los tipos de relés de protección según el tipo de protección son muy importantes para proteger los diferentes sistemas estos detectan la señal y actúan inmediatamente, algunos de estos relés son: relés de potencia inversa, relés de

tiempo inverso, relés de tiempo definitivo, relés de bajo voltaje, relés de tierra, relés de sobrecorriente, relés de distancia y relés diferenciales [16].

1.2.8 Protecciones de las sobrecorrientes

En un SEP las anomalías más comunes que se presentan durante su funcionamiento, son los cortocircuitos en las líneas de transmisión, lo que ocasiona al aumento elevado de la corriente por sobre los valores que operan normalmente es decir corrientes extremadamente altas, por estas anomalías que alteran al funcionamiento del sistema eléctrico de potencia, se utilizan las protecciones de sobrecorriente que ayuda a disminuir la afección de las fallas en los equipos como en otros circuitos que actúan en el sistema eléctrico, de igual forma la cualidad principal es de proteger la integridad de las personas que se encuentran trabajando en esta área para ello son muy importantes la colocación de estos relés de sobrecorriente que evita accidentes y muertes repentinas [17].

1.2.9 Característica de operación de las protecciones de sobrecorriente

Este tipo de relés de sobrecorriente según su operación de las protecciones eléctricas, estos pueden clasificarse en tres grupos: relés de corriente definida, relés de tiempo definido, y relés de tiempo inverso, las curvas características de operación de estos relés en función de tiempo- corriente, se muestran en la Figura 2.

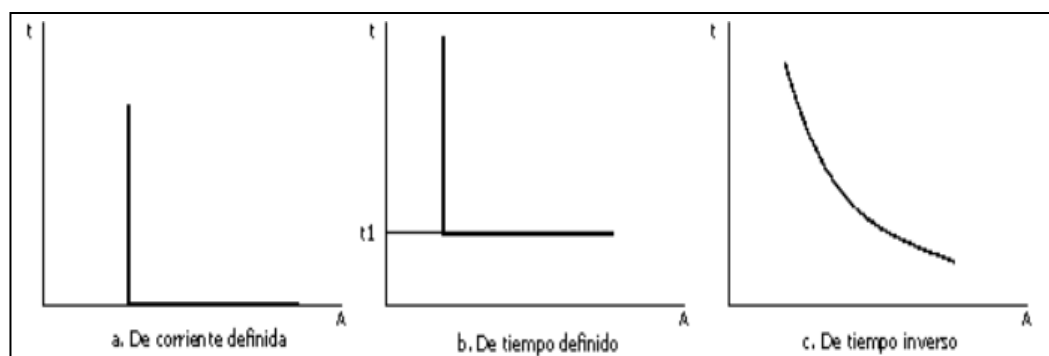


Figura 2. Curvas características de operación Tiempo-Corriente.

Fuente: [16]

1.2.10 Relés de Sobrecorriente

Son dispositivos de protección cuya cualidad es la detección de fallas dentro de una línea de transmisión y operan conjuntamente con los interruptores y otros elementos de nuestro diagrama o circuito eléctrico [16].

1.2.11 Relés de sobrecorriente por tiempo de operación

Los relés de sobrecorriente de acuerdo al tiempo de operación, la característica principal de estos relés es que al momento de su operación se toma en cuenta el tiempo de operación, estos relés se pueden clasificar en dos tipos de relés: instantáneos y temporizados.

a) Relés de sobrecorriente instantáneos

Los relés instantáneos de sobrecorriente en todo sistema eléctrico actúa en un tiempo menor o igual a 50 ms es decir actúa en ese instante de manera instantánea a la presencia de una falla o corriente de falla, cuando la corriente es mayor a la de operación este dispositivo de abre de manera instantánea evitando el paso de la corriente, la norma ANSI se designan por el número 50.

b) Relés de sobrecorriente temporizados

Los relés de temporizados operan en un tiempo después de que la corriente de falla fue detectada en los SEP, es decir cuando la corriente de falla sobrepasa los valores de corriente de operación, estos relés de sobrecorriente temporizados actúan de acuerdo a sus parámetros de configuración como el Dial, corriente de falla, tap, etc. Una de las características esenciales en este tipo de dispositivos es analizar la curva de operación tiempo-corriente ya que permitirá definir si estamos hablando de un relé de sobrecorriente de tiempo inverso o tiempo definido, la norma ANSI se designan por el número 51 [18].

1.2.12 Relés de sobrecorriente por la característica de operación

Los relés de sobrecorriente se considera las características de operación de funcionamiento, pueden haber de diferentes tipos de acuerdo a sus parámetros de

funcionamiento como relés: de corriente definida, relés de corriente, relés de tiempo definido y relés de tiempo inverso.

a) Relés de corriente definida

La operación de estos relés es de manera instantánea, es decir cuando la corriente de cortocircuito alcanza un valor establecido por el usuario (I_a), el relé operara instantáneamente de acuerdo se vaya alejando de la zona de operación.

Algo negativo de este tipo de relés de sobrecorriente es que no operan con valores altos de cortocircuitos, por lo que tienen problemas al momento de detectar una corriente de falla o cortocircuito entre varios elementos de nuestro sistema de potencia véase la Figura 3 [16].



Figura 3. Relé de corriente definida

Fuente: [16]

b) Relés de corriente definida y tiempo definido

La operación de estos relés permiten variar parámetros de funcionamiento para los diferentes valores de corriente de falla como son modificar el dial, tap entre otros y así nos presentara diferentes tiempos que actuara nuestro relé de sobrecorriente, por consiguiente los ajustes se hacen en forma tal, que el relé salte en un tiempo más corto y después se podrá configurar para diferentes tiempos cada vez mayores como se muestra en la Figura 4 [16].

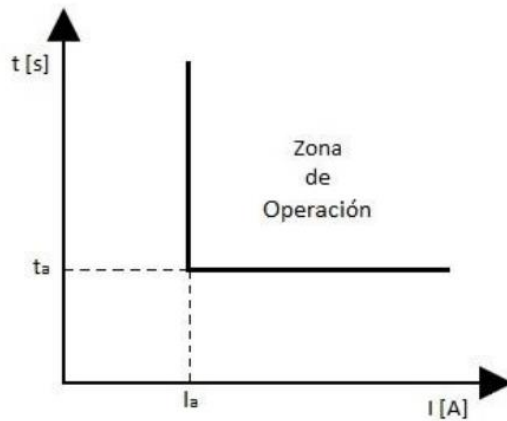


Figura 4. Relé de corriente definida y tiempo definido

Fuente: [16]

c) Relés de tiempo inverso

La operación de estos tipos de relés se da en un tiempo inversamente proporcional a la corriente de falla en función de las curvas de operación de este tipo de relés de sobrecorriente, cuando se presentan corrientes de cortocircuito altas los tiempos de operación son muy cortos de acuerdo a la ecuación matemática y parámetros que comprende cada uno de estos relés y así se protege de mejor manera al sistema eléctrico de potencia como se muestra en la Figura 5, se clasifican de acuerdo a la curva característica que viene dado por las gráficas de corriente vs tiempo y estos son : relés de tipo inverso, muy inverso, extremadamente inverso tiempo definido véase la Figura 6 [16].

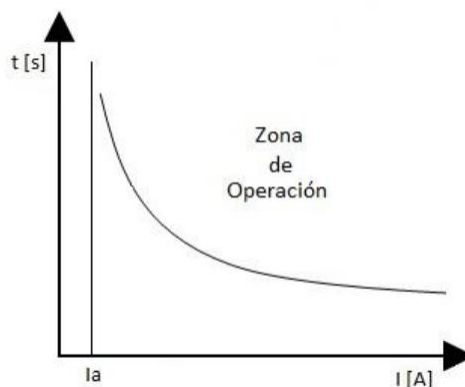


Figura 5. Relés de tiempo inverso

Fuente: [16]

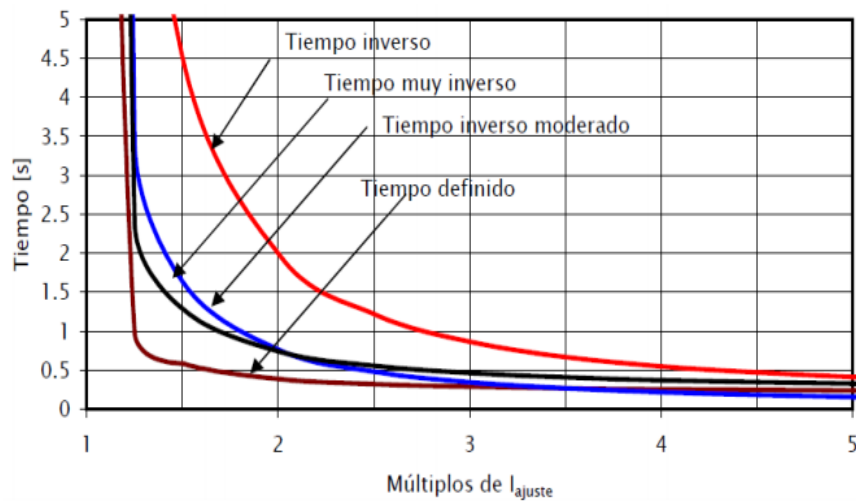


Figura 6. Diferentes tipos de relés de tiempo inverso

Fuente: [16]

1.2.13 Modelo matemático de los relés de sobrecorriente

En el modelo matemático de los relés de sobrecorriente el tiempo de operación de los relés es la función más importante del mismo para proteger los diferentes sistemas de potencia, a continuación, se presentará una ecuación que nos permite calcular los tiempos de operación de los relés de sobrecorriente en función de los diferentes parámetros [18].

$$t = DIAL \left(\frac{a}{\left(\frac{I}{I_s}\right)^n - 1} + C \right) \quad \text{Ec. (2)}$$

Donde:

t = Tiempo de operación del relé de sobrecorriente.

I = Corriente de falla inyectada al relé de sobrecorriente.

I_s = TAP o Corriente de Operación del Relé.

$DIAL$ = Constante de ajuste del Relé

n y a = Parámetro que define la curva característica de operación del Relé.

C = Constante de ajuste del Relé.

El tiempo de operación de los relés de sobrecorriente de tiempo inverso será de acuerdo a los valores de los parámetros de las constantes a y n , los cuales varían de acuerdo al tipo de relé de sobrecorriente esto se presenta en la Tabla 1.

Tabla 1. Constantes de forma para ecuación exponencial según normas IEC

Tipo de Curva	IEC 255 Y BS 142		
	N	α	C
Moderadamente inversa_IEEE	0,02	0,0515	0,114
Muy inversa_IEEE	2	19,61	0,491
Extremadamente Inversa_IEEE	2	28,2	0,1217
Extremadamente Inversamente Largo_IEEE	2,43	100	0,5
Inversa_CO8	2	5,95	0,18
Inversa de corto tiempo_CO2	0,02	0,0239	0,0169
Inversa Estandar_IEC	0,02	0,14	0
Muy inversa_IEC	1	13,5	0
Extremadamente Inversa_IEC	2	80	0

Fuente: [18]

1.2.14 Criterios de coordinación de protecciones 50/51

Para la coordinación de estos dispositivos se obtiene a través de la corriente mínima de disparo y del retardo de tiempo de operación, para que durante la operación o ejecución de un sistema eléctrico exista una sensibilidad para poder detectar las diferentes fallas que se pueden presentar en líneas de transmisión, otro de los criterios muy importantes en la coordinación es la selectividad este criterio nos permite segregar las fallas en zonas de protección primarias y secundarias.

Para que exista una correcta coordinación de los relés de sobrecorriente y sea selectiva, es preferible utilizar curvas de tiempo inverso, extremadamente inverso, inversamente largo, etc.

Primero para la coordinación de protecciones es establecer límites en las zonas de protección que dichos equipos deberán actuar es decir poner un límite máximo y mínimo de funcionamiento y todos los parámetros necesarios de los equipos de protección de nuestro sistema eléctrico, como marca, modelo, número de partes del relé, dial, tap, etc. Segundo se debe conocer las características del sistema eléctrico de potencia a proteger, es decir tener criterio de diseño y saber los parámetros de las líneas como tipo, transformadores de potencia de corriente, interruptores de potencia, dial, tipo de relé, etc [19].

En la Figura 7 se presenta un ejemplo de coordinación, existe dos tipos de relés de sobrecorriente Relé 1 y Relé 2 que son configurados con diferentes parámetros de funcionamiento como es el dial y la corriente de pickup, al presentarse una falla en nuestro sistema se puede observar cómo los relés actúan de manera coordinada con una variación de tiempo, que mientras este tiempo de operación sea menor es más beneficioso para nuestro sistema para evitar todas las anomalías y que el sistema eléctrico de potencia se encuentre funcionando y brinde todas las necesidades al usuario.

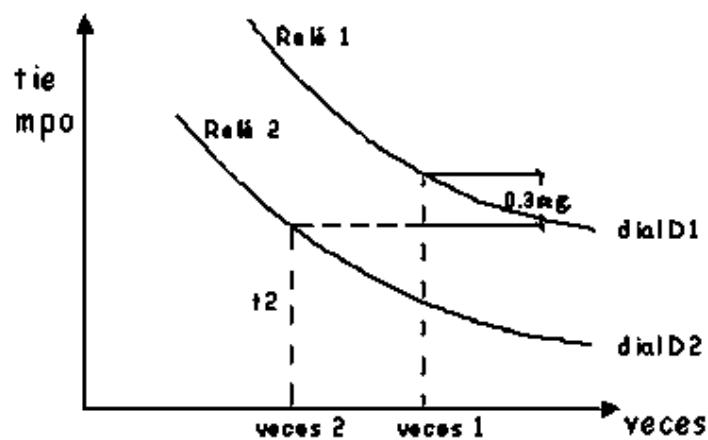


Figura 7. Coordinación de los relés de sobrecorriente.

Fuente: [19]

El primer relé desde el lado de carga toma el dial más bajo si no existen otros elementos de protección hacia el lado de carga. El cálculo del dial para los relés siguientes hacia el lado de alimentación se realiza tomando en cuenta un tiempo de 0.3 segundos, que corresponde a errores de tolerancia, sobrecarga, etc, que pudieran presentarse en estos tipos de relés.

1.2.15 Cálculo de cortocircuitos según normas ANSI/IEEE

Hoy en día existen normas relativas al cálculo de cortocircuitos con el fin de realizar los diferentes análisis de las protecciones eléctricas.

a) Corrientes Simétricas

El ajuste de su reactancia dependiendo del tipo de máquina establece la norma.

b) Corrientes Asimétricas

La norma recomienda que para obtener las corrientes de falla asimétricas se utilice multiplicadores aplicados a las corrientes de fallas simétricas, obtenidas de curvas parametrizadas relacionando el tiempo de apertura de los contactos del interruptor con la relación entre la reactancia y la resistencia de la red en el punto de falla.

c) Corriente de Retardo vs. Corriente de Estado Permanente.

Durante el estado de retardo la corriente de falla recibe el nombre de corriente de retardo, se recomienda modelar los generadores de la red en función de su reactancia transitoria o de su reactancia de estado permanente.

Para la obtención de esta corriente la norma recomienda la aplicación de la relación entre su reactancia y resistencia $\frac{X}{R}$.

La relación se obtiene a partir de la Ec. (3):

$$\frac{X}{R} = \frac{X_C}{R_C} * \frac{f}{f_c} \quad \text{Ec. (3)}$$

Donde:

f = Frecuencia del sistema

f_c = Frecuencia de la fuente equivalente

X_C = Reactancia equivalente de falla

R_C = Resistencia equivalente de falla

Con esta relación se obtiene la corriente pico a través de la corriente inicial de cortocircuito.

d) Corriente Inicial de Cortocircuito.

Es aquella corriente que se presenta luego de iniciado el cortocircuito.

e) Voltaje de Prefalla

La norma establece que el valor a considerar como voltaje de prefalla sea de 1 p.u. en las barras del sistema.

Las normas ANSI/IEEE C37.010 – 1979 y C37.5 – 1979 establecen métodos de cálculo de Corrientes de cortocircuito con base a valores simétricos y asimétricos respectivamente, para la especificación de interruptores de alto voltaje.

f) Norma ANSI/IEEE c37.010 – 1979.

Esta norma es una guía para la selección y aplicación de interruptores a nivel de alto voltaje.

Presenta dos métodos de cálculo: método simplificado E/X y el método corregido de E/X . El segundo método presenta resultados más exactos por lo tanto se enfatizará únicamente en éste.

El Método corregido de E/X llamado también “Método corregido de E/X con decrementos CA y CC” provee un planteamiento preciso para la aplicación de interruptores de alto voltaje con base a valores simétricos.

Este método determina un procedimiento para la aplicación de factores de multiplicación a los cálculos de E/X . E es el voltaje típico de prefalla (1 p.u.) y X es la reactancia equivalente vista desde el punto de falla.

La norma recomienda seguir una serie de pasos que se pueden resumir en lo siguiente:

1. Construcción del diagrama de impedancias con base a parámetros y datos del sistema, dados por los fabricantes ó en tablas.
2. Determinación del voltaje de prefalla, en general se asume $E = 1 p.u.$
3. Determinación de la red de reactancias dentro de los regímenes momentáneos y de interrupción. Las reactancias de interrupción se corrigen aplicando factores de multiplicación adecuados dadas por la norma y permiten construir la red de secuencia positiva, negativa y cero.
4. Determinación de la red de resistencias de interrupción de secuencia positiva, negativa y cero, a partir del diagrama unifilar. Las resistencias son modificadas por los mismos factores de multiplicación aplicados reactancias.

5. Reducción de las redes de secuencia a reactancias y resistencias equivalentes vistas desde el punto de vista de la falla.
6. Cálculo de corrientes de cortocircuito para las redes eléctricas se lo realiza de la siguiente manera:

Falla Monofásica:

Régimen momentáneo:
$$I_{CC10M} = 1.6 * \frac{3 * E}{X_{M1} + X_{M2} + X_{M0}} \quad \text{Ec. (4)}$$

Régimen de interrupción:
$$I_{CC10I} = \frac{3 * E}{X_{I1} + X_{I2} + X_{I0}} \quad \text{Ec. (5)}$$

Falla Trifásica:

Régimen momentáneo:
$$I_{CC30M} = 1.6 * \frac{E}{X_{M1}} \quad \text{Ec. (6)}$$

Régimen de interrupción:
$$I_{CC30I} = \frac{E}{X_{I1}} \quad \text{Ec. (7)}$$

Donde:

I_{CC10M} , I_{CC10I} = Corriente de cortocircuito monofásica de interrupción y momentánea.

I_{CC30M} , I_{CC30I} = Corriente de cortocircuito trifásica de interrupción y momentánea.

X_{M1} , X_{M2} , X_{M0} = Reactancias momentáneas de secuencia positiva, negativa y cero, respectivamente reducidas al punto de falla.

$X_{I1} + X_{I2} + X_{I0}$ = Reactancias de interrupción de secuencia positiva, negativa y cero, respectivamente reducidas al punto de falla.

La norma indica que la selección del interruptor será apropiada si la corriente de Cortocircuito calculada multiplicada por el factor de 1.6 no excede la corriente

momentánea del interruptor, es decir, se compara con la corriente de cortocircuito asimétrica.

7. Determinación de la razón X/R equivalente en el punto de falla para la red de interrupción como sigue:

Falla Monofásica:

$$\frac{X}{R} = \frac{2 \cdot X_{I1} + X_{I0}}{2 \cdot R_{I1} + R_{I0}} \quad \text{Ec. (8)}$$

Falla Trifásica:

$$\frac{X}{R} = \frac{X_{I1}}{R_{I1}} \quad \text{Ec. (9)}$$

Donde:

R_{I1} = Resistencia de interrupción y momentánea de secuencia positiva respectivamente, reducidas al punto de falla.

8. Selección del factor de multiplicación (FM) para la corrección de E/X, obtenida en el paso 6 correspondiente a la red de interrupción.
9. Obtenido el factor FM, se multiplica a la relación E/X para tener la corriente de interrupción impuesta al interruptor.
10. Finalmente se determina la capacidad de interrupción simétrica del Interruptor con base a las características de especificación dadas [20].

1.2.16 Softwares de simulación de sistemas eléctricos de potencia

En los diferentes procesos de simulación de sistemas eléctricos de potencia existen gran variedad de softwares que cada uno de ellos con sus funciones son muy útiles y permiten el análisis completo de estos sistemas, entre los más importantes tenemos DIgSILENT PowerFactory, PSS/E, ETAP.

1.2.17 Software DIgSILENT PowerFactory

Es un software de simulación de sistemas eléctricos de potencia que permite realizar diferentes análisis completos de varios casos de estudios eléctricos, en el tiempo presente este software es muy utilizado en el sector eléctrico del país por las

empresas CENACE y TRANSELECTRIC entre las más importantes, además es muy utilizado por los estudiantes para las diferentes investigaciones de los casos de estudios eléctricos.

PowerFactory dispone de múltiples funciones para la simulación de sistemas eléctricos de potencia como son; el análisis de fallas, flujos de potencia, análisis de protecciones entre otros. Este software también presenta dos lenguajes de programación DPL (DIgSILENT Programming Language) y DSL (DIgSILENT Simulation Language) [21].

1.2.18 Compatibilidad de DIgSILENT PowerFactory

DIgSILENT Power Factory es compatible con una gran variedad de softwares de programación, pero en este caso se realizará la compatibilidad que exista con Python que es un lenguaje de programación ampliamente utilizado ya que es muy eficiente y fácil de entender [21].

1.2.19 Software Python

Es un lenguaje de programación de alto nivel ya que contiene múltiples estructuras de datos y gran variedad de funciones que permite ejecutar en muchas plataformas, es utilizado en desarrollo de software, aplicaciones web machine learning entre otras. Entre los beneficios que tiene este lenguaje de programación tenemos que es un software libre, también permite ser más productivo en la programación ya que con menos líneas de programación se puede alcanzar múltiples funciones en comparación con otros lenguajes, es compatible con otros lenguajes de programación como Java, C, C++ entre otros.

Python es un lenguaje bien interpretado, un software fácil de instalar de manera gratuita que nos permite crear varios softwares o programas libres, en este caso en nuestra presente investigación utilizaremos la versión de Python 3.3.2 que es la única versión compatible con el software DIgSILENT PowerFactory [22].

1.2.20 Compatibilidad de Python

Cuando se realiza la programación en Python existen diferentes maneras de comunicar o interactuar esta herramienta como es comunicar con otros lenguajes

de programación como es Java, C, C++ para poder realizar las diferentes necesidades requeridas por el usuario, además de comunicarse con otros software Python nos da la posibilidad de interactuar o programar en varios intérpretes como lo es Visual Studio Code que es el intérprete que vamos utilizar en nuestro trabajo de investigación ya que dispone de muchas funcionalidades como el autocompletado, formateo de código, herramientas para hacer debugging y testear código Python, y el manejo de entornos virtuales [23].

1.2.21 Visual Studio Code

Visual Studio Code es un editor de código fuente elaborado por Microsoft para los sistemas operativos Linux, Windows, MacOS y web, es compatible con varios lenguajes de programación y con un sin número de diversas funciones según el lenguaje en el que sea interpretado.

La interfaz es algo amigable ya que se puede configurar de acuerdo a las necesidades del usuario, en el rol de editor de código fuente este permite cambiar la página de coditos en la que se guarda el archivo, el carácter que identifica el salto de línea y el lenguaje de programación del trabajo que se encuentre programando. Finalmente, una de las características principal de Visual Studio Code es la capacidad de crear extensiones que analizan los diferentes códigos de programación como linters y herramientas para análisis estático utilizando protocolo de servidor de idioma dentro de su programación [24].

1.4. Conclusiones Capítulo I

Los relés de sobrecorriente son dispositivos que limita el flujo de corriente de un sistema, para evitar daños en los equipos que conforman un sistema eléctrico de potencia, y brindan seguridad y protección a los usuarios, según la característica del tiempo de operación se clasifican en relés de sobrecorriente instantáneo 50 y relé de sobrecorriente inverso 51, según la norma ANSI.

En las líneas de transmisión la coordinación de protecciones eléctricas determina las características, rangos, y ajustes de los dispositivos de protección contra

sobrecorriente que aseguren que cuando exista una falla mínima en la carga se interrumpa cuando los dispositivos de protección aíslen la falla o la sobrecarga.

El lenguaje de programación Python permite comunicar y extraer los datos de los sistemas eléctricos de potencia en DIgSILENT PowerFactory.

2 CAPÍTULO II. PROPUESTA

2.1 Título del proyecto

Desarrollo de una herramienta informática que permita determinar los tiempos de operación, curvas de operación y coordinación de los relés de sobrecorriente en las líneas de transmisión.

2.2 Objetivo del proyecto

Desarrollar una herramienta informática que brinde al usuario una interfaz gráfica que permita comunicar con el Software PowerFactory y extraer los datos para el análisis de funcionamiento de los relés de sobrecorriente en los sistemas eléctricos de potencia.

2.3 Descripción de la propuesta

En el capítulo anterior se ha tratado los detalles más importantes sobre la implementación de relés de sobrecorriente para la protección de equipos eléctricos dentro de un sistema de alta potencia, también se han analizado los parámetros más importantes como corriente de pickup y dial necesarios para una correcta coordinación de estos elementos de protección. Para un correcto entendimiento del proceso de coordinación de protección de sobrecorriente es necesario que ilustración mediante una propuesta de desarrollo de una interface entre Python y PowerFactory y aplicada a dos estudios de caso.

En la **Figura 8** se observa las etapas aplicadas durante el desarrollo de la herramienta informática que comunica una interface implementada en el software Python con el programa de simulación de circuitos eléctricos PowerFactory. Esta interfaz es capaz de analizar el circuito eléctrico para obtener valores de corriente, voltaje, corriente de corto circuito en la barra, número de relés existentes y sus parámetros de coordinación.

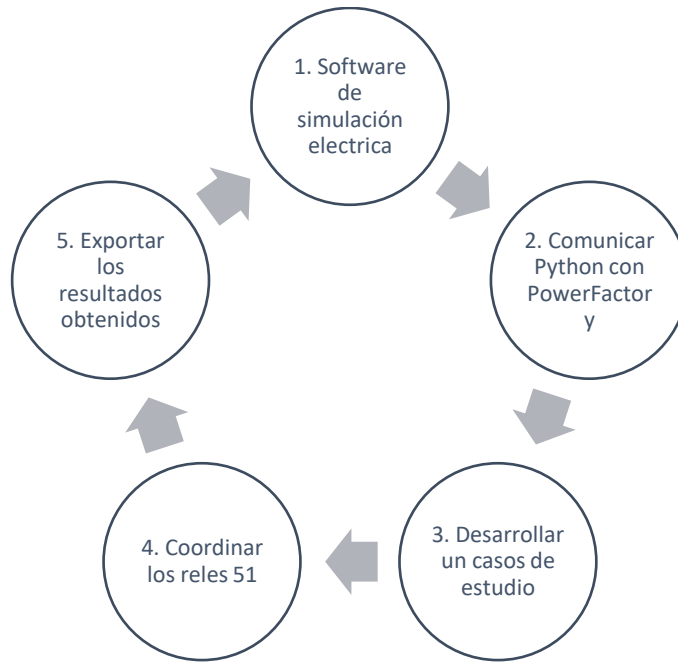


Figura 8. Análisis de un caso de estudio.

2.4 Metodología o procedimientos empleados para el cumplimiento de los objetivos planteados

Con el fin de cumplir con los objetivos específicos planteados en la presente investigación en primera instancia se requiere analizar los diferentes softwares disponibles en el mercado y que permitan simular circuitos eléctricos de alta potencia con una amplia librería de relés de sobre corriente. A continuación, se procede a comunicar este programa con el software Python para extraer los parámetros más importantes del sistema eléctrico para desarrollar una buena coordinación de las protecciones. Finalmente, esta información debe ser exportada a un archivo Excel para que el usuario pueda interpretar los datos obtenidos o aplicar fórmulas para validar los resultados, todo este proceso se realiza mediante una programación en Python. Ver Anexo 6

2.4.1 Comunicación Python y DIgSILENT PowerFactory

Para establecer una comunicación entre el programa de simulación de circuitos eléctricos DIgSILENT PowerFactory y Python se debe considerar los requisitos de compatibilidad expuestos, a continuación:

- DIgSILENT PowerFactory
- Python 3.3.2

Posteriormente se procede a seguir los siguientes pasos de instalación, cabe recalcar que para que exista una correcta interpretación de la codificación por parte de todos los programas involucrados en el desarrollo de la herramienta informática se requiere seguir el proceso descrito en la Figura 9.

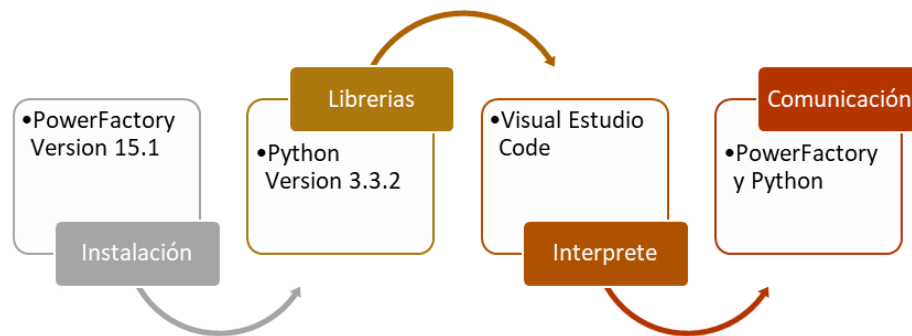


Figura 9. Proceso de comunicación de la herramienta informática.

- Cabe recalcar que el proceso necesario para la instalación de DIgSILENT PowerFactory, Python 3.3.2 y Visual Studio Code se puede encontrar en los Anexo 1, Anexo 2 y Anexo 3, respectivamente.

2.4.2 Diagramas de flujo de funcionamiento

El funcionamiento del programa inicia con la ejecución de la herramienta informática, la primera venta permite que el usuario seleccione la dirección de instalación de DIgSILENT PowerFactory y Python. Una vez que se valida que estas direcciones son correctas se procede activar el programa. En la Figura 10 se puede observar el diagrama de flujo para esta función.

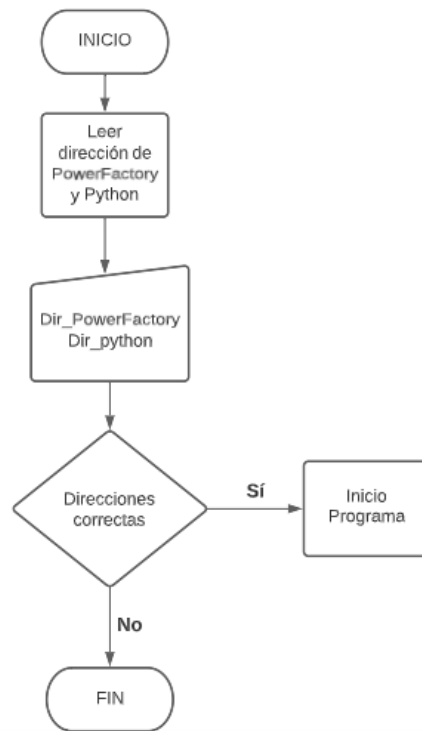


Figura 10. Primera ventana de la herramienta informática.

Posteriormente se genera una segunda ventana donde se puede encontrar todo el funcionamiento y las variables de la herramienta informática. En este caso se requiere que el usuario ingrese de forma manual el nombre de un proyecto existente dentro de las funciones de DIGSILENT PowerFactory. Cabe recalcar que este nombre tiene que ser idéntico tanto en la herramienta informática como en DIGSILENT PowerFactory. La herramienta informática realiza una validación de esta información revisando la base de datos de todos los proyectos existentes en PowerFactory para proceder a la obtención de los datos, como se muestra en la Figura 11.

Una vez que se activado exitosamente el proyecto que se requiere analizar se procede activar las funciones de análisis de flujo de potencia y el cálculo de corto circuito donde el usuario lo requiera. También se activa la lectura de los relés, barras y parámetro de las ecuaciones de comportamiento del relé de sobrecorriente instalados en el circuito eléctrico de potencia.

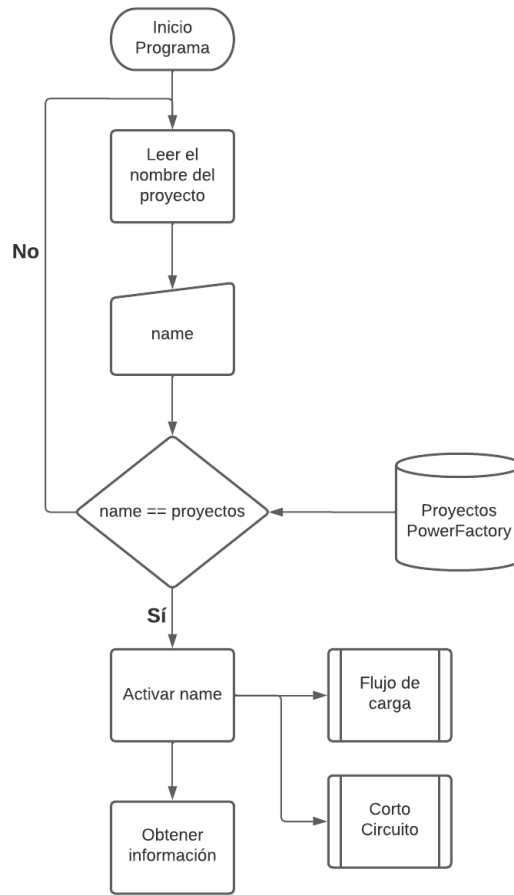


Figura 11. Ingreso del proyecto en la herramienta informática.

La función de análisis de un flujo de potencia se activa cuando el usuario activa el botón correspondiente (ON), este proceso se repite cada vez que el usuario lo requiera; para lo cual se requiere activar el comando “ComLdf” y posteriormente el comando “Execute”. Cuando esta función se realiza la herramienta informática esta lista para obtener los valores de simulación del circuito eléctrico de potencia. El diagrama de flujo para este proceso se puede observar en la Figura 12.

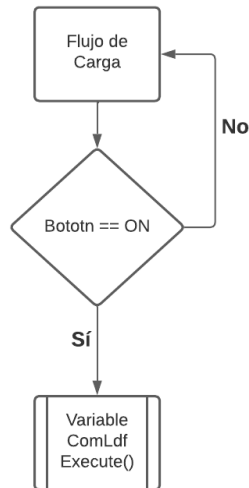


Figura 12. Análisis de un flujo de potencia.

De la misma manera el análisis de un corto circuito inicia con la activación del botón correspondiente para esta función. Sin embargo, previo a este proceso el usuario debe seleccionar la barra donde se requiere aplicar un análisis de corto circuito. Los valores de configuración son por defecto y se muestran en la Tabla 5, además la Figura 13 muestra el procedimiento de este caso.

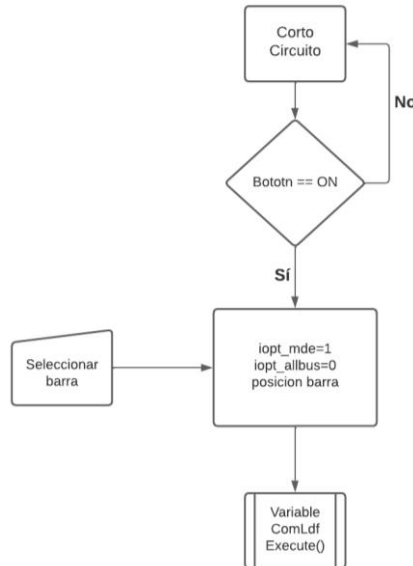


Figura 13. Análisis de corto circuito en una barra.

Para leer los relés de sobrecorriente instalados en el circuito eléctrico de potencia se requiere utilizar la variable “RelIoc” en el filtro de objetos, la herramienta informática realiza una lectura de la base de datos en DIgSILENT PowerFactory

para determinar cuántos relés están instalados y sus respectivos nombres. A continuación, se crea una lista con los nombres de estos elementos para que el usuario los pueda visualizar y de esta manera extraer los valores de “Pickup” y “Dial”; esta información también es enviada a la interface gráfica, como se muestra en la Figura 14.

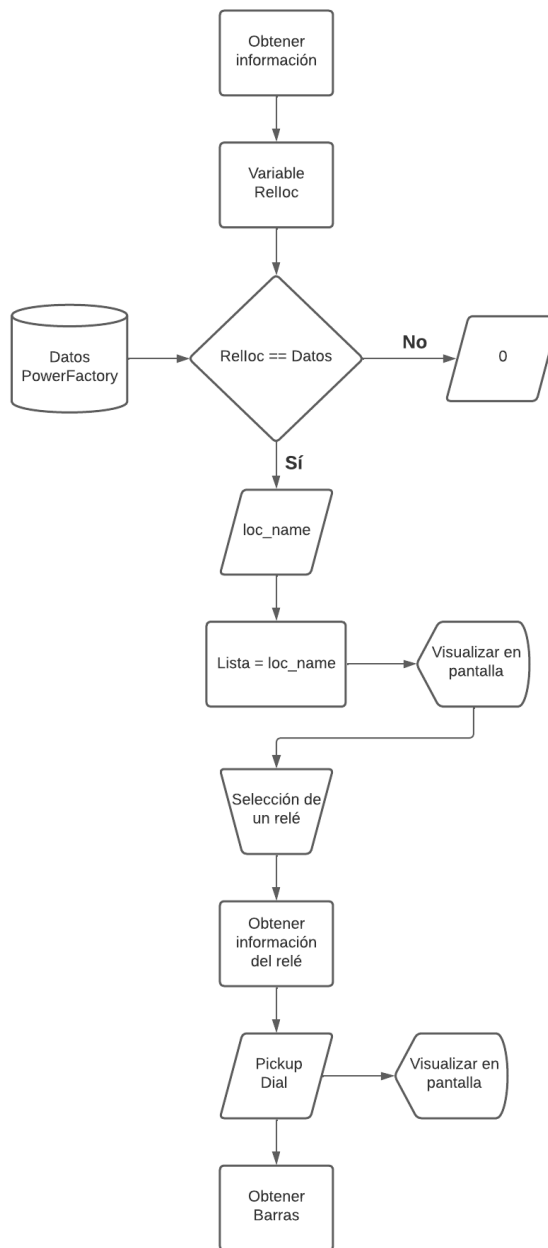


Figura 14. Obtener los relés de sobrecorriente.

El proceso de lectura de las barras existentes en el proyecto activado es similar a la lectura de relés, con la diferencia que se debe utilizar la variable “ElmTerm” y la herramienta informática únicamente extrae los nombres de las variables que inician con “Barra” caso contrario no obtiene dicha información. En la Figura 15 se muestra el proceso de lectura de barras y la obtención de las variables de simulación “loc_name, m:UI, m:phiu, m:u”).

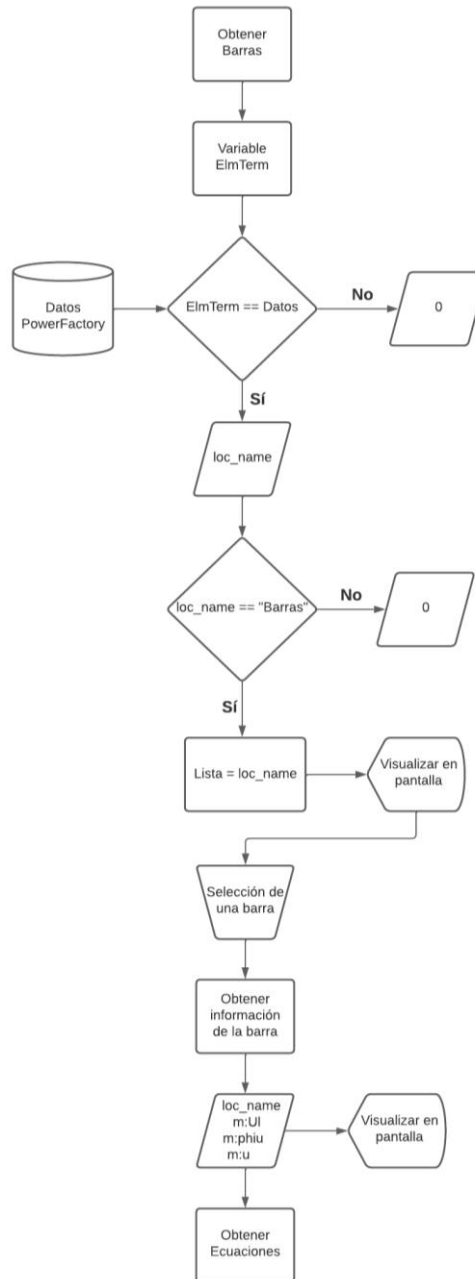


Figura 15. Obtener las barras existentes en el proyecto.

La lectura de las constantes de las ecuaciones del comportamiento de los relés ANSI/IEEE se obtiene utilizando la variable “TypChatoc” en el filtro de objetos, esta información se valida con la base de datos de DIGSILENT PowerFactory y se obtienen las constantes a_1, a_2, a_3 y a_4 . La herramienta informática reemplaza dichos valores en las ecuaciones y los datos son enviados hacia una hoja de cálculo en Excel donde se tabulan y generan gráficos con escalas logarítmicas.

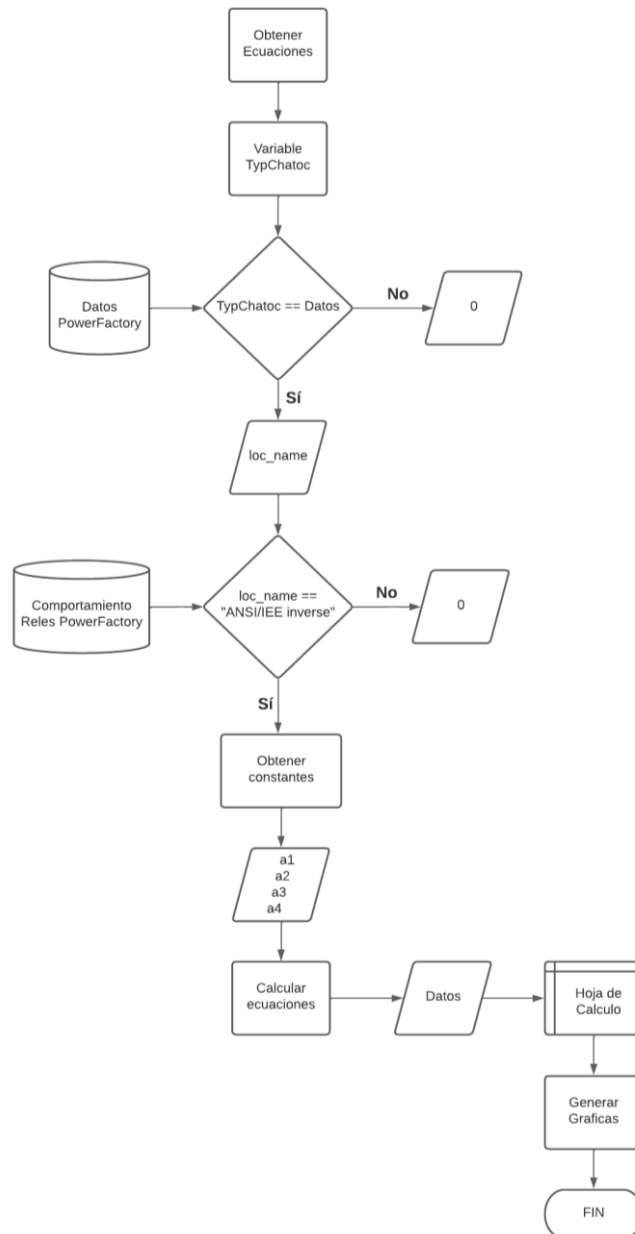


Figura 16. Obtener las constantes de las ecuaciones de los relés.

2.4.3 Comunicación DIgSILENT PowerFactory y Python

En la Tabla 2 se procede a identificación todas las librerías y la codificación utilizada para establecer una comunicación adecuada entre DIgSILENT PowerFactory y Python. Cane recalcar que es muy importante identificar la dirección de instalación de PowerFactory en el Windows, así como también la dirección de la carpeta con las librerías de Python.

Tabla 2. Instrucciones para establecer comunicación entre los programas.

	Visual Studio Code	Descripción
Librerías	<code>import os</code>	Cargar interfaces misceláneas del sistema operativo.
	<code>import sys</code>	Importar parámetros y funciones específicas del sistema.
Código	<code>os.environ["PATH"]=r"C:\Program Files (x86)\DIgSILENT\PowerFactory 15.1"+os.environ["PATH"]</code>	Instrucción para localizar la carpeta de instalación de PowerFactory
	<code>sys.path.append(r"C:\Program Files (x86)\DIgSILENT\PowerFactory 15.1\python")</code>	Instrucción para localizar la carpeta de Python dentro de la instalación de DIgSILENT.

2.4.4 Desarrollo de la interface grafica

Para el desarrollo de una interface grafica se debe considerar la versión de Python 3.3.2 y las librerías disponibles para esta función. Tkinter es una librería que contiene el kit de herramientas GUI Tcl/Tk, con esta librería se puede crear ventanas, agregar cuadros de texto o labels, botones, entre otros; como se muestra en la Tabla 3.

Tabla 3. Instrucciones para crear una ventana.

	Visual Studio Code	Descripción
Librerías	<code>from tkinter import messagebox, from tkinter import ttk from tkinter import filedialog from tkinter import *</code>	Librerías Tkinter
Código	<code>ventana = tkinter.Tk() ventana.title('Mi titulo') ventana.geometry("1050x400") ventana.config(bg="white")</code>	Crear una ventana
Código	<code>btn_guardar_datos = tkinter.Button(ventana, text ="Guardar datos", command=aviso_rele, width=20,</code>	Crear un botón

	<code>bg="#5585f5", fg="white").place(x=547, y=200)</code>	
Código	<code>Label(ventana, text="Dial del relé:", borderwidth=1, relief="solid", fg="black", width=20) entry_dial = ttk.Entry(width=20) entry_dial.place(x=750, y=170)</code>	Crear un label

Por otro lado, una vez aplicada la codificación se procede a crear una ventana principal de la herramienta informática. En esta ventana se solicita al usuario que ingrese la dirección donde se ha instalado DIgSILENT PowerFactory y a su vez la dirección de la carpeta Python donde se encuentra la respectiva librería. Además, contiene un botón que inicializa la comunicación entre la herramienta informática y el simulador de circuitos eléctricos de potencia PowerFactory. En la Figura 17 se muestra los elementos que contiene esta ventana principal.

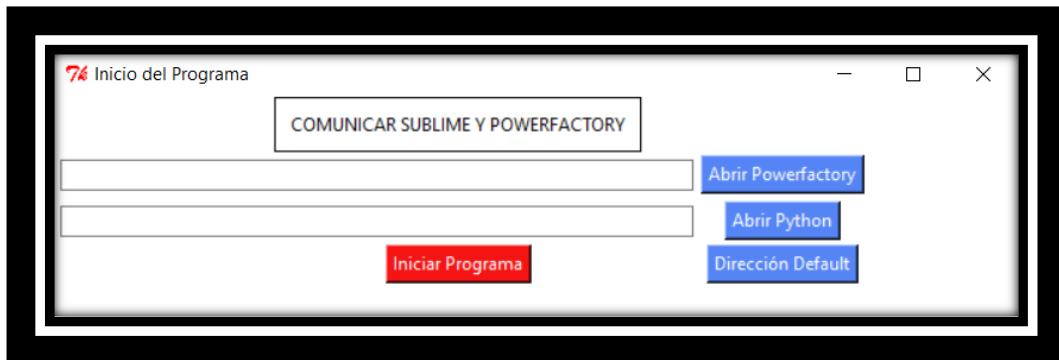


Figura 17. Ventana principal de la herramienta informática.

En la Figura 18 se muestra una ventana secundaria contiene toda la funcionalidad de la herramienta informática, donde se puede identificar los siguientes elementos importantes:

- Ingreso del nombre del circuito eléctrico que se desea analizar
- Número y nombre de los relés encontrados
- Barras encontradas con el nombre inicial de “Barra”
- Parámetros de los relés como pickup y dial
- Botones para ejecutar simulación en corto circuito
- Graficas para representar el comportamiento de los relés



Figura 18. Ventana de coordinación de protecciones.

2.4.5 Comandos de DIgSILENT PowerFactory

En esta sección se procede a detallar los comandos que permiten ejecutar un análisis de flujos de carga y un estudio de corto circuito, cabe recalcar que la activación de estos comandos requiere de una configuración adecuada para cada uno de los casos. En la Tabla 4 se observa el comando utilizado para activar un flujo de carga, sin embargo, los parámetros de configuración para esta simulación son por defecto.

Tabla 4. Instrucciones activar un flujo de carga.

	Visual Studio Code	Descripción
Código	<code>app.GetFromStudyCase('ComLdf')</code> <code>Execute()</code>	ComLdf comando para activar un flujo de carga con parámetros por defecto.

Por otro lado, los parámetros de configuración para el análisis de flujo de carga se muestran en la Figura 19, donde se puede observar la siguiente configuración:

- Flujo de carga AC, Balanceado, Secuencia positiva
- Dependencia de temperatura a 20°C
- Acumuladores nocturnos al 100%
- Balance de referencia de maquina
- Control de potencia activa tal como se envía

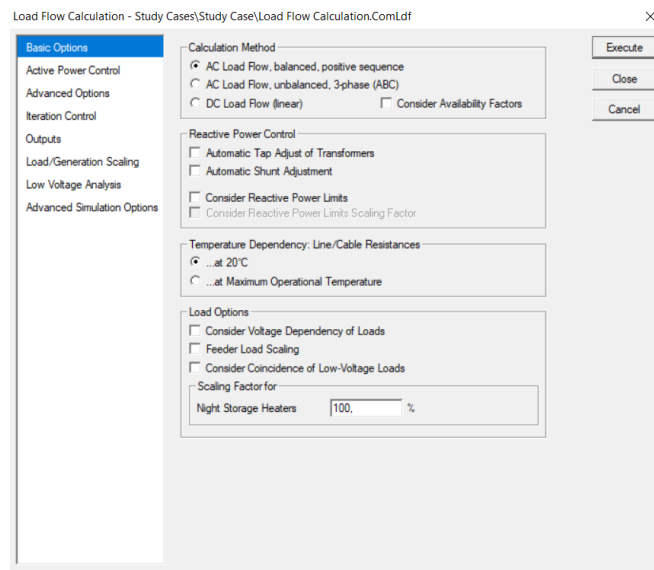


Figura 19. Configuración de un análisis de flujo de potencia.

En lo que se refiere al cálculo de un corto circuito se deben utilizar los códigos de programación descritos en la Tabla 5, donde se muestran los valores por defecto y los posibles métodos de simulación.

Tabla 5. Instrucciones activar un corto circuito.

	Visual Studio Code	Descripción
Código	<code>ejecutarComShc = app.GetFromStudyCase('ComShc')</code>	Configurar la variable ejecutarComShc como un análisis de corto circuito.
	<code>ejecutarComShc.iopt_mde = 1</code>	Selección del método de simulación: 0 = VDE0102 1 = IEC 60909 2 = ANSI 3 = Complete 4 = IEC 61633 5 = IEC 61660DC 6 = ANSI/IEEE 946 (DC) 7 = DIN EN 61660(DC)
	<code>ejecutarComShc.iopt_allbus = 0</code>	Localización de falla 0 = Selección por el usuario 1 = Barras y nodos de unión 2 = Todas las barras
	<code>ejecutarComShc.shcobj = posicion_barra</code>	Selección de la una barra para calcular el corto circuito desde la interface
	<code>ejecutarComShc.Execute()</code>	Ejecutar el cálculo de corto circuito.

Por otro lado, en la Figura 20 se puede observar los valores de simulación de un corto circuito en el software DIGSILENT PowerFactory. Estos valores pueden ser modificados directamente desde la interface desarrollada en este proyecto de tesis y calcular automáticamente los valores de corto circuito.

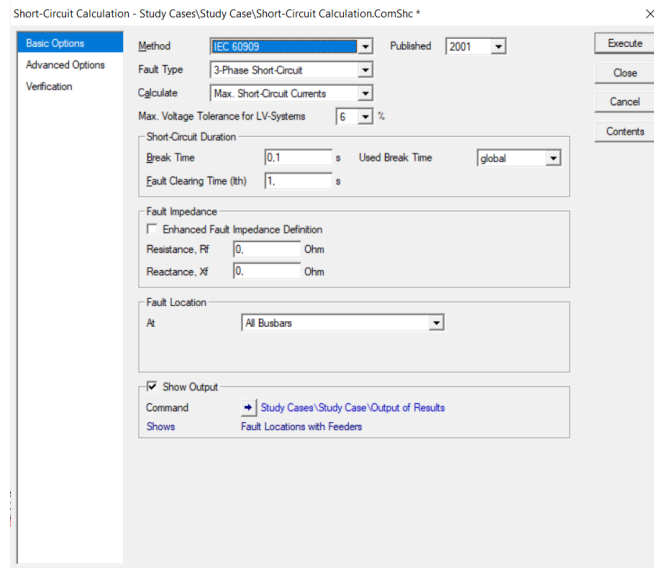


Figura 20. Configuración de un análisis de corto circuito.

2.4.6 Obtener valores desde DIGSILENT PowerFactory

Una vez que se establecido la comunicación entre la herramienta informática desarrollada en Python y el simulador de circuitos eléctricos de potencia en PowerFactory, se procede activar un caso de estudio para lo cual se debe identificar el nombre exacto del circuito que se desea analizar. En la Figura 21 se observa la lista de proyectos existentes y los que pueden vincularse directamente con la herramienta informática para su análisis.

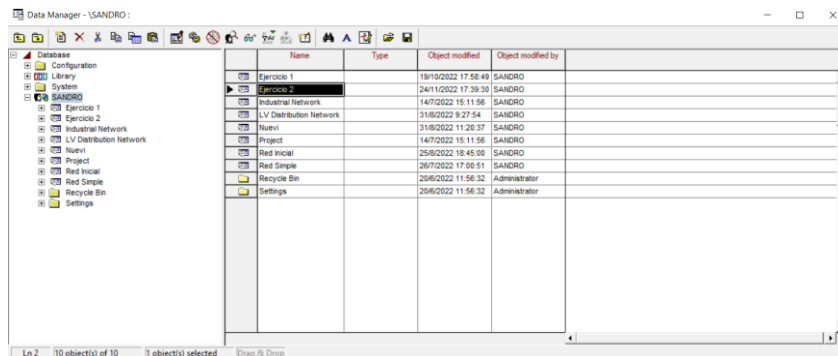


Figura 21. Listado de proyectos en PowerFactory.

En la Tabla 6 se observa el código de programación necesario para activar un proyecto desde la herramienta informática, cabe recalcar que el nombre ingresado en la interface tiene que ser idéntico al nombre existente en DIgSILENT PowerFactory. Si este no se cumple con esta condición, la herramienta informática no podrá activar el proyecto y tampoco extraer los parámetros de análisis.

Tabla 6. Código para activar un proyecto.

	Visual Studio Code	Descripción
Código	<code>project = app.ActivateProject(caso.get())</code> <code>prj = app.GetActiveProject()</code>	En esta sección se lee el nombre del proyecto desde la interface.
	<code>study_case_folder=app.GetProjectFolder('study')</code>	Ingresa a la carpeta del proyecto solicitado.
	<code>study_case=study_case_folder.GetContents('*.Intcase')[0]</code>	Obtiene la información del caso de estudio con todos sus componentes.
	<code>case.Activate()</code>	Activa el proyecto con su caso de estudio para el análisis.

La lectura de los relés existentes en el proyecto activado desde la interface se realiza median la variable “.ElmRelay”. En la Figura 22 se muestra el filtro de objetos donde se especifica todos los relés existentes dentro del proyecto en DIgSILENT PowerFactory, en este caso son los siguientes:

- Relay Model
- Rele 1
- Rele 2

Name	In Folder	Grid	Category	Relay Type	Application	Device Number	Reference	Busbar	Remote End	Connected Branch	Out of Service	Active Settings	Net Elements	CT
Relay 1	Cub_1	Grid	Overcurrent	Phase 14/15	Protection	1		Barra 1	Barra 2	T1_13/11a			CT1	CT1
Relay 2	Cub_1	Grid	Overcurrent	Phase 14/15	Protection	1		Barra 2	Barra 3	T1_11/33a			CT2	CT2

Figura 22. Listado de relés en PowerFactory.

En la Tabla 7 se muestra la programación requerida para obtener los nombres de todos los relés existentes en el proyecto activado. Esta información es enviada hacia la interface a modo de lista para el usuario pueda seleccionar una y observar sus parámetros de configuración.

Tabla 7. Lectura de relés desde la interface.

	Visual Studio Code	Descripción
Código	<code>Reles = app.GetCalcRelevantObjects('*.ElmRelay')</code>	Obtiene todos los objetos que sean de la característica ElmRelay
	<code>for rele in Reles: listado_reles[rele.loc_name]=rele name = getattr(rele, 'loc_name')</code>	Crema una lista con los nombres de cada uno de los relés encontrados.
	<code>barras_rele_cbx['values']=list(listado_reles) vuno_rele=list(listado_reles)</code>	Envía los nombres hacia la interfaz para que el usuario pueda escoger uno en una lista.

Este proceso se repite para obtener el nombre de todas las barras existentes en el proyecto activado, para lo cual se debe utilizar la variable “ElmTerm”. Cabe recalcar que en la implementación del circuito eléctrico de potencia en PowerFactory se debe nombrar a las barras con el nombre de “Barra 1, Barra 2, Barra 3,, Barra #n”, respectivamente. En la Tabla 8 se puede observar la codificación requerida para este caso.

Tabla 8. Lectura de barras desde la interface.

	Visual Studio Code	Descripción
Código	<pre>Buses=app.GetCalcRelevantObjects('*.ElmTerm')</pre>	Obtiene todos los objetos que sean de la característica ElmTerm.
	<pre>for bus in Buses: name = getattr(bus, 'loc_name') if search("Barra", name): listado_barras[bus.loc_name]=bus</pre>	Crea una lista con los objetos que tienen como nombre "Barra"
	<pre>barras_cbx['values']=list(listado_barras) vuno=list(listado_barras)</pre>	Envía los nombres hacia la interfaz para que el usuario pueda escoger uno en una lista.

En la Figura 23 se muestra una representación gráfica del funcionamiento de la herramienta informática realizando una lectura de los siguientes parámetros:

- Número de relés encontrados
- Nombre de cada uno de los relés en una lista
- Valores de pickup y dial de cada relé
- Barras encontradas representadas en una lista
- Lectura de los valores de UL, PHIU, U, Skss, Ikss, Ip., del circuito



Figura 23. Lectura de valores desde la herramienta informática.

2.4.7 Proceso para generar las gráficas de los relés de sobrecorriente

En el circuito eléctrico de potencia se pueden instalar relés de sobrecorriente de diferentes marcas, gracias a las librerías que posee DIGSILENT PowerFactory. En la Figura 24 se puede observar las librerías y las marcas de los relés disponibles para la implementación en un circuito eléctrico, en este caso se optó por la utilización de los relés de la librería IEC/ANSI.

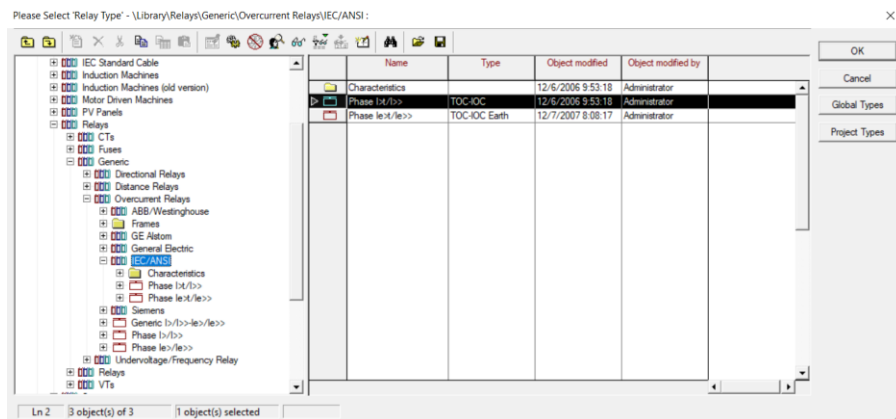


Figura 24. Librerías de relés de sobrecorriente PowerFactory.

Una vez identificada la librería de los relés de sobre corriente se procede a escoger uno de ellos, en este caso se decidió utilizar el relé de fase. Los parámetros de configuración se pueden observar en la Figura 25.

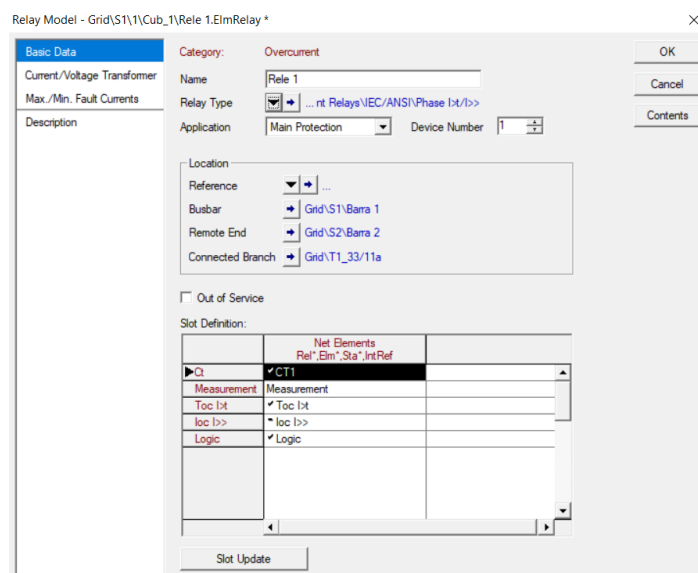


Figura 25. Parámetros del relé de fase

Si embargo, la generación de las gráficas que describen el comportamiento de los relés instalados en el circuito eléctrico de potencia en DIgSILENT PowerFactory se pueden encontrar en el filtro de objetos con el nombre de “TypChatoc”, como se muestra en la Figura 26.

Name	Object modified by	Usage	Range	Range	Like	Min Current (Ip)	Max Current (Ip)	Min Tripping Time (s)	Min Time	Max Time	Function	No. of Curves	Use Same x-Values	Draw Stat Time O	Values	Expression
ANSI/IEEE definite inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	0.4797	
ANSI/IEEE extremely inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	5.64	
ANSI/IEEE inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	8.9341	
ANSI/IEEE long inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	5.6143	
ANSI/IEEE moderately inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	0.0103	
ANSI/IEEE short inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	0.2663	
ANSI/IEEE very inverse	6/2006 9.53.10 Administrator		0/0.5-15.0.1		s	1.1	20.	0.05	0	oo	2	2	☑	☑	3.922	
IEC 255-3 extremely inverse	6/2006 9.53.10 Administrator		0/0.05-3.2.0.01		s	1.1	20.	0.05	0	oo	1	1	☑	☑	80.	
IEC 255-3 inverse	6/2006 9.53.10 Administrator		0/0.05-3.2.0.01		s	1.1	20.	0.05	0	oo	1	1	☑	☑	0.14	
IEC 255-3 long inverse	6/2006 9.53.10 Administrator		0/0.05-3.2.0.01		s	1.1	20.	0.5	0	oo	1	1	☑	☑	120.	
IEC 255-3 very inverse	6/2006 9.53.10 Administrator		0/0.05-3.2.0.01		s	1.1	20.	0.05	0	oo	1	1	☑	☑	13.5	

Figura 26. Parámetros del relé de fase

Los relés instalados en el circuito eléctrico son de la marca IEC/ANSI y pueden tener diferentes comportamientos, los mismos que se describen a continuación:

- a) ANSI/IEEE I Squared T
- b) ANSI/IEEE definitive inverse
- c) ANSI/IEEE extremely inverse
- d) ANSI/IEEE inverse
- e) ANSI/IEEE long inverse
- f) ANSI/IEEE moderately inverse
- g) ANSI/IEEE short inverse
- h) ANSI/IEEE very inverse
- i) IEC 255-3 extremely inverse
- j) IEC 255-3 inverse
- k) IEC 255-3 long inverse
- l) IEC 255-3 very inverse

Cabe recalcar que cada uno de estos comportamientos describen una ecuación diferente y a continuación se describe dichas ecuaciones.

Ecuación para el comportamiento de (a)

$$T = \frac{T_p * a_1 + a_2}{\left(\frac{I}{I_p}\right)^2} \quad \text{Ec. (10)}$$

Ecuación para el comportamiento de (b, c, d, e, f, g, h)

$$T = T_p * \left(\frac{a_1}{\left(\left(\frac{I}{I_p}\right)^{a_2} - a_3\right) + a_4} \right) \quad \text{Ec. (11)}$$

Ecuación para el comportamiento de (i, j, k, l)

$$T = T_p * \frac{a_1}{\left(\left(\frac{I}{I_p}\right)^{a_2} - a_3\right)} \quad \text{Ec. (12)}$$

Los parámetros reemplazables a_1 , a_2 , a_3 y a_4 en las ecuaciones se obtienen desde directamente desde la Figura 27.

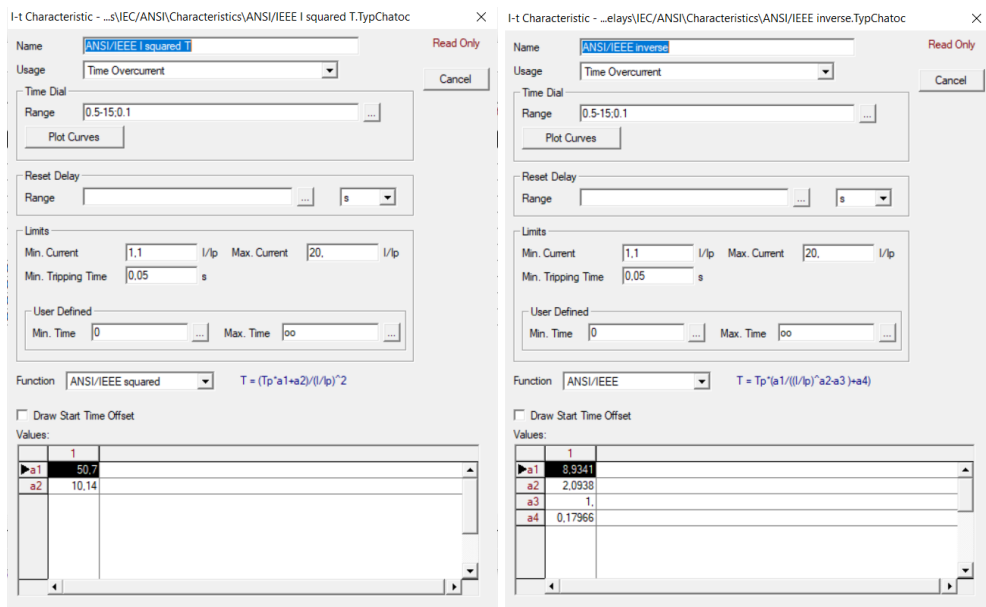


Figura 27. Parámetros de comportamiento del relé de fase

En la Figura 28 se muestran las gráficas de los relés de fase en el comportamiento de ANSI/IEEE inverso y ANSI/IEEE muy inverso, respectivamente.

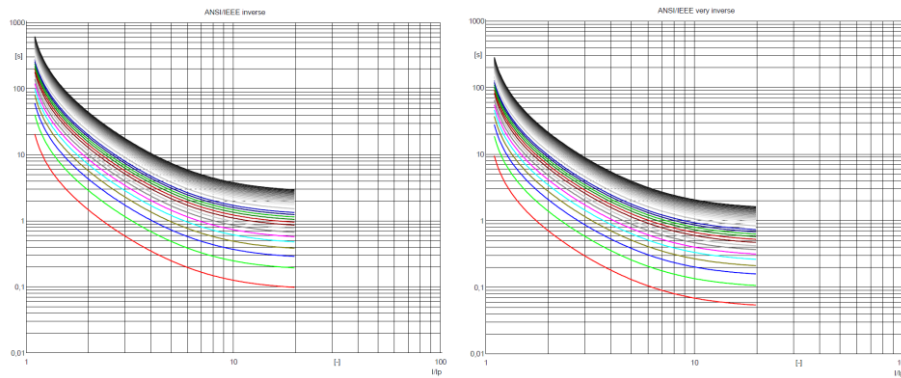


Figura 28. Graficas del relé de fase

Para el proceso de funcionamiento de la herramienta informática, se compila nuestro programa en Python interprete Visual Studio Code, y se nos presenta una interfaz con los diferentes elementos del diagrama eléctrico que se va analizar, además de presentar todos los datos necesarios para poder graficar, coordinar los relés de sobrecorriente, para este proceso se procede a realizar una transcripción de las ecuaciones del comportamiento de los relés de sobrecorriente con el objetivo de generar una representación propia de este proyecto de tesis. En la Tabla 9 se muestran las 3 ecuaciones utilizadas para el relé de marca ANSI/IEEE.

Tabla 9. Ecuaciones de comportamiento de los relés de sobrecorriente.

	Visual Studio Code	Descripción
Código	$T[x] = (TP * \sum(a1+a2)) / (Iip_min * Iip_min)$	Ecuación (1)
	$a = Iip_min ** a2$ $T[x] = (TP * (a1 / (a - a3) + a4))$	Ecuación (2)
	$a = Iip_min ** a2$ $T[x] = (TP * a1 / ((a - a3)))$	Ecuación (3)

Posteriormente se procede a crear un archivo en Excel para enviar los datos generados por la herramienta informática y de esta manera generar gráficos del comportamiento de todos los relés, para lo cual se debe aplicar las líneas de código representado en la Tabla 10.

Tabla 10. Crear un archivo en Excel.

	Visual Studio Code	Descripción
Código	<pre>workbook = xlsxwriter.Workbook('Graficas.xlsx') ec1 = workbook.add_worksheet()</pre>	Crea una hoja de cálculo en Excel con el nombre de Graficas.xlsx

A continuación, se procede a extraer los valores de las constantes de a_1 , a_2 , a_3 y a_4 desde el circuito eléctrico de potencia desarrollado en DIgSILENT PowerFactory, para lo cual se debe implementar la codificación representada en la Tabla 11. Cabe recalcar que este proceso se repite para todos los comportamientos de los relés de sobrecorriente.

Tabla 11. Obtener las variables de las constantes a_1 , a_2 , a_3 y a_4 .

	Visual Studio Code	Descripción
Código	<pre>if (vector[i] == 'ANSI/IEEE I squared T'):</pre>	Leer los valores de las constantes cuando sea afirmativa esta comparación.
	<pre>ec1.write(0,1,vector[i])</pre>	Escribir el nombre del comportamiento en la hoja de cálculo.
	<pre>valores = getattr(list_caract[vector[i]], 'vmat')</pre>	Obtener los valores de las constantes en la variable vector.
	<pre>a1=valores[0] a2=valores[1]</pre>	Asignar los valores obtenidos de las variables de a_1 , a_2 , ..., a_n .
	<pre>a=(TP*sum(a1+a2))/(Iip_min*Iip_min</pre>	Reemplazar los valores en la ecuación.
	<pre>ec1.write_number(filas,1,a)</pre>	Grabar los datos obtenidos en la hoja de cálculo.

Finalmente se procede a crear una gráfica en Excel desde la herramienta informática con los datos obtenidos del cálculo de las ecuaciones (1), (2) y (3), para lo cual se debe utilizar las instrucciones detalladas en la Tabla 12.

Tabla 12. Obtener las variables de las constantes a_1, a_2, a_3 y a_4 .

Visual Studio Code		Descripción
Código	<code>chart1 = workbook.add_chart({'type': 'line'})</code>	Crear un gráfico de dispersión desde la herramienta informática.
	<code>chart1.add_series({ 'name': '='Sheet1!\$B\$1', 'categories':='Sheet1!\$A\$2:\$A\$21', 'values': '='Sheet1!\$B\$2:\$B\$21', })</code>	Obtener el nombre del gráfico, datos en el eje x, los datos del eje y.

A través de la herramienta informática se crea automáticamente un archivo en formato xlsx con todos los datos que se requiera de nuestro sistema eléctrico. Esto servirá para que el usuario pueda manipular esta información según sea su necesidad, como se muestra en la Figura 29.

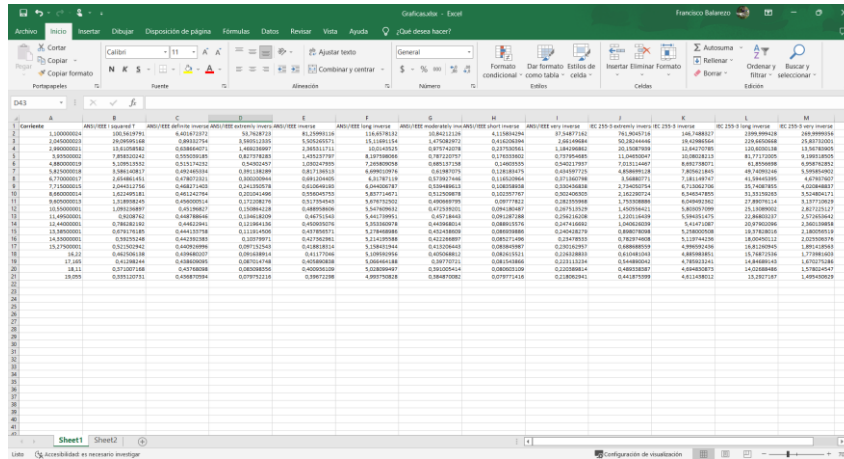


Figura 29. Datos de las ecuaciones para cada comportamiento.

Finalmente, en la Figura 30 se puede observar todos los comportamientos de los relés de marca ANSI/IEEE generados a través de codificación de la herramienta informática en una hoja de cálculo en Excel.

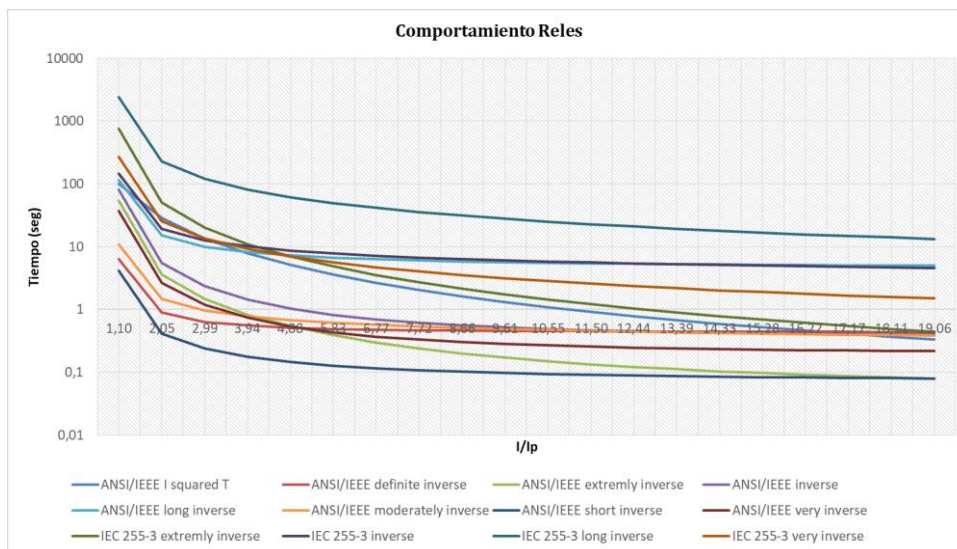


Figura 30. Curvas del comportamiento del relé ANSI/IEEE.

2.5 Conclusiones Capítulo II

Los problemas de comunicación de los softwares de diseños eléctricos con otros programas se deben a las versiones y licencias de los mismos, la información sobre esto este tema es limitada, con Python y DIgSILENT PowerFactory se presentaron problemas de compatibilidad de las versiones que limitaron algunas funciones de la herramienta informática.

La herramienta informática extrae datos de DIgSILENT PowerFactory de la operación de los relés de sobrecorriente en los sistemas eléctricos, y genera un reporte en un archivo Excel en donde se encontrará los tiempos de operación, curvas de operación, y coordinación de estos tipos de protecciones.

Esta herramienta permite ejecutar de forma automática los flujos de carga en todo el circuito para obtener los valores de voltaje y corriente en cada una de las barras; también se puede realizar un análisis de cortocircuito y el usuario puede observar dichos valores en la interface grafica.

3 CAPITULO III. APLICACIÓN Y/O VALIDACIÓN DE LA PROPUESTA

3.1 Análisis de Resultados

En esta sección se procede a realizar un análisis de flujo de carga, cortocircuito en todas las barras y la coordinación de los relés existentes en el caso de estudio, utilizando la herramienta informática; para lo cual se iniciará por aplicar un análisis de flujo en DIGSILENT PowerFactory y su respectivo procedimiento en la herramienta informática.

3.2 Descripción del primer caso de estudio

Una vez que se realizado el análisis del circuito eléctrico de potencia existente en las bases de datos de la Universidad Técnica de Cotopaxi en la carrera de Electricidad, se procede probar el funcionamiento de la herramienta informática desarrollada con el circuito existente en la base de datos de PowerFactory denominada como “*39 Bus New England Sustum*” con los datos de referencia publicados en [25].

3.2.1 39 Bus New England System

El circuito eléctrico de potencia “*39 Bus New England System*” consta de 39 buses (nodos), 10 generadores, 19 cargas, 34 líneas y 12 transformadores. Se muestra en el ANEXO 11 el diagrama unifilar. Es un modelo simplificado del sistema de transmisión en el área de Nueva Inglaterra en el noreste de los EE.UU.

La frecuencia nominal del sistema de transmisión de Nueva Inglaterra es de 60 Hz y el nivel de tensión de la red es de 345 kV (tensión nominal). Para los nodos con un nivel de voltaje diferente, se han asumido los siguientes voltajes nominales para el modelo PowerFactory:

- Barra 12: 138 kV
- Barra 20: 230 kV
- Barra 30 – Barra 38: 16,5 kV

Para la lectura de las 39 barras instaladas en este circuito de potencia se utilizó la palabra “Bus” para poder acceder los datos. En la Figura 31 se puede observar los resultados de la ejecución de la herramienta informática con este circuito de potencia.



Figura 31. Análisis del circuito de potencia de 39 barras.

3.2.2 Análisis de Flujo de Potencia

Los valores obtenidos para el flujo de potencia se encuentran descritos en el Anexo 7 y Anexo 8, donde se puede observar los datos obtenidos en el software PowerFactory y Python respectivamente. En la Figura 32 se puede observar el error generado por la magnitud del voltaje línea a línea del circuito de potencia de 39 barras entre el software PowerFactory y Python.

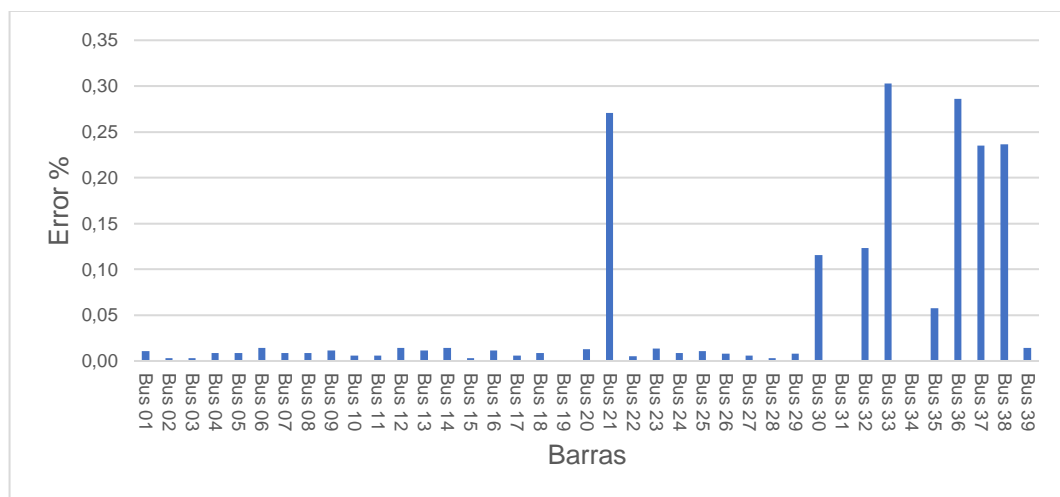


Figura 32. Error de medición en el voltaje.

Por otro lado, en la Figura 33 se puede observar el error en la variable del ángulo de voltaje generado entre el software PowerFactory y la herramienta informática; cabe recalcar que el máximo error en este caso es de 6% en la barra 23.

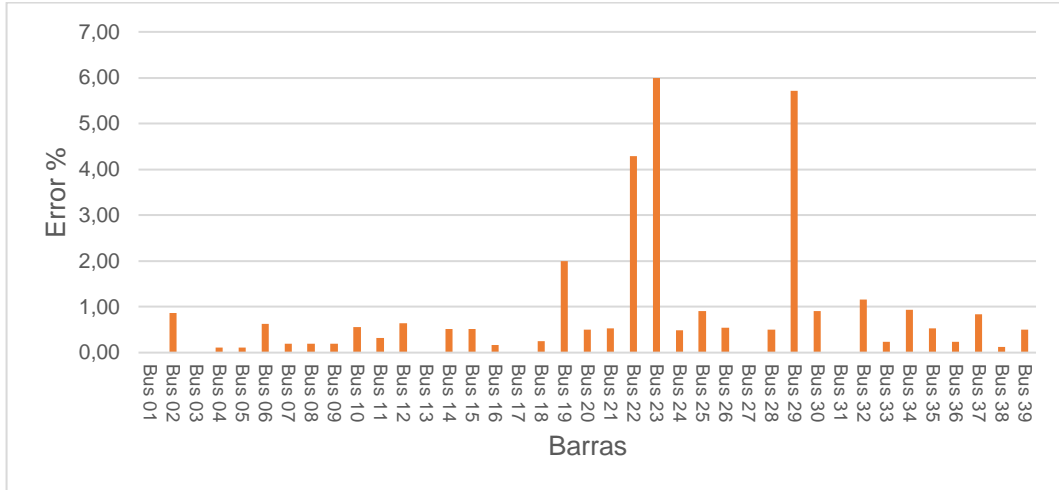


Figura 33. Error de medición en el ángulo de voltaje.

Finalmente, en la Figura 34 se muestra el error generado por la medición de la magnitud de voltaje del circuito de potencia; donde se puede identificar el máximo error de 0.5%.

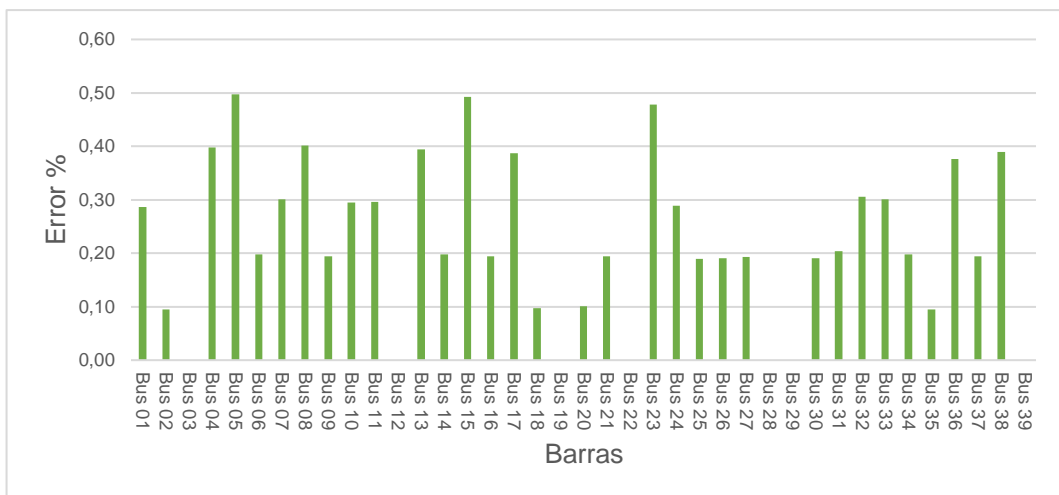


Figura 34. Error de medición en la magnitud de voltaje.

3.2.3 Análisis de Corto Circuito

En esta sección se procede a realizar un análisis de corto circuito al caso de estudio titulado “39 Bus New England System” con el objetivo de determinar el error existente en este análisis entre los dos programas utilizados; cabe recalcar que los valores de las tablas se encuentran en los Anexos 9 y 10. En la Figura 35 se muestra la potencia de cortocircuito inicial de este caso de estudio.

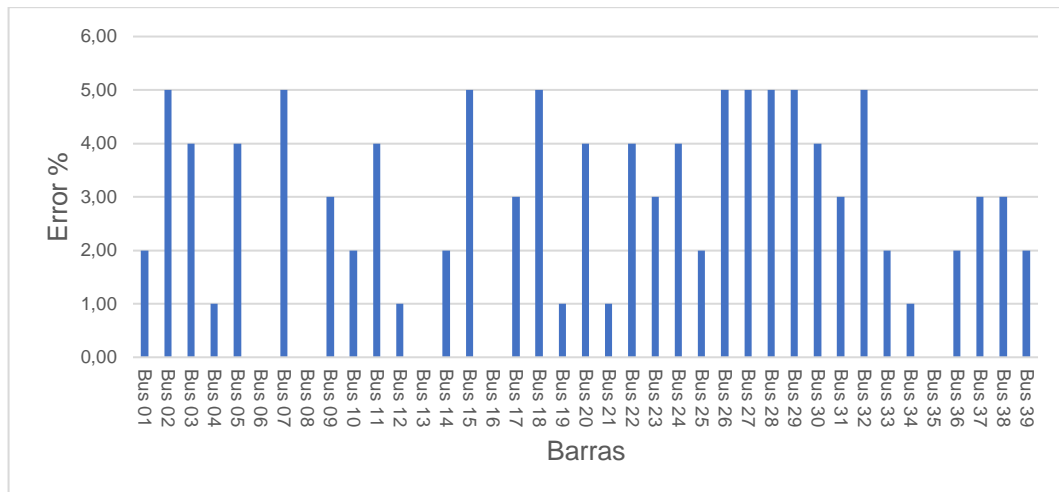


Figura 35. Error de medición la potencia de cortocircuito inicial.

En la Figura 36 se puede observar el error generado en la medición de la corriente de cortocircuito inicial del circuito de 39 barras.

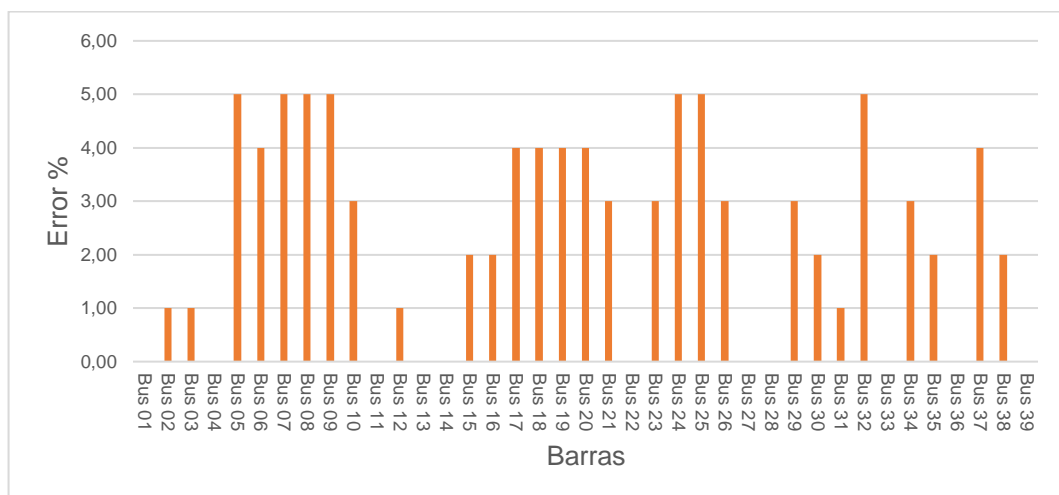


Figura 36. Error de medición la potencia de cortocircuito inicial.

Por otro lado, la Figura 37 representa la corriente pico de cortocircuito existente en el circuito de potencia de 39 barras.

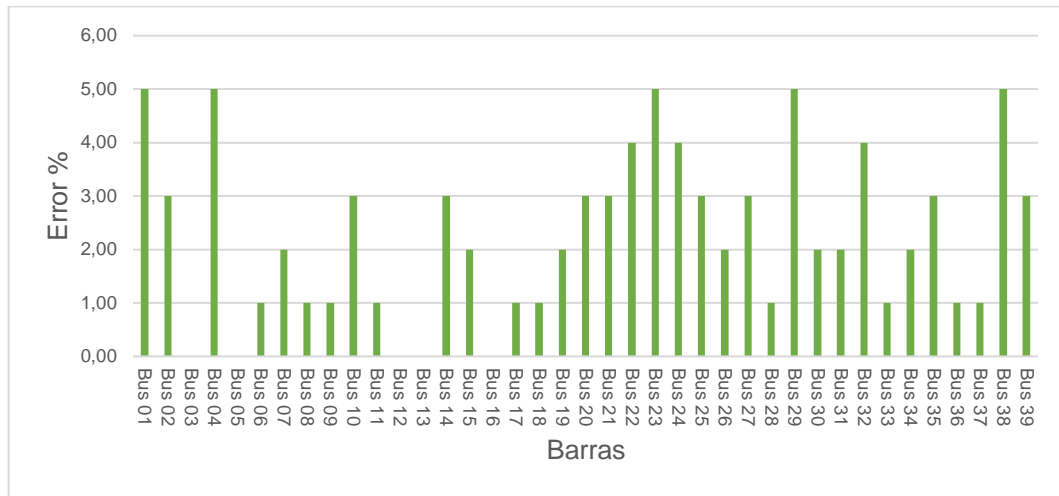


Figura 37. Error de medición la potencia de cortocircuito inicial.

3.2.4 Descripción del segundo caso de estudio

Este caso de estudio corresponde a un circuito eléctrico de potencia existente en las bases de datos de la Universidad Técnica de Cotopaxi en la carrera de Electricidad, dispone de 8 relés de sobrecorriente instalados, 2 relés en cada una de las 3 líneas de transmisión y 2 relés en el transformador como se muestra en la Figura 38.

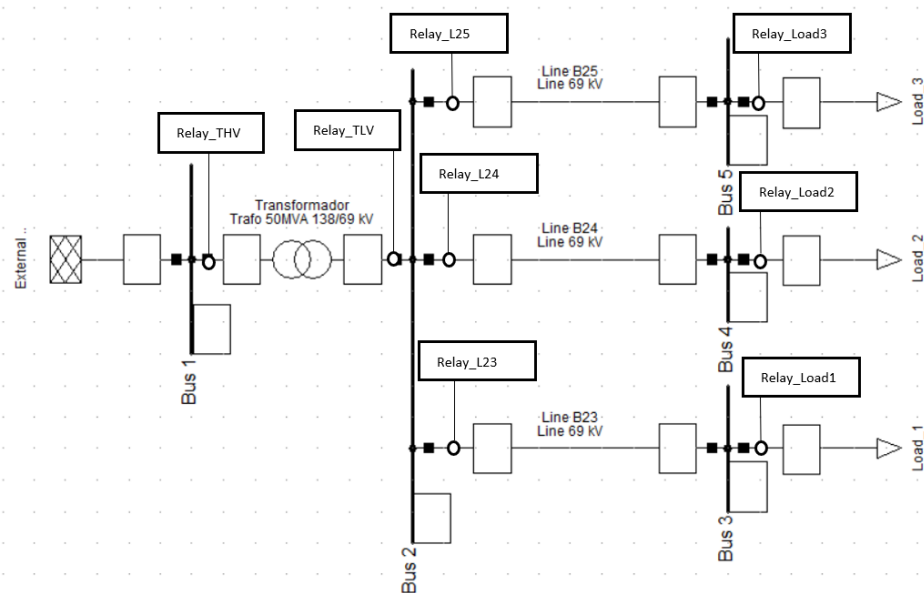


Figura 38. Caso de estudio original con 3 líneas de transmisión.

Antes de proceder a aplicar un análisis con la herramienta informática desarrollada en este proyecto de tesis, se procede a renombrar las barras de todo el circuito eléctrico, utilizando la palabra inicial de “Bus”; de esta manera se podrá identificar cada una de ellas y aplicar un análisis de flujo de carga o un análisis de corto circuito. Los resultados de este proceso se pueden observar en la Figura 39

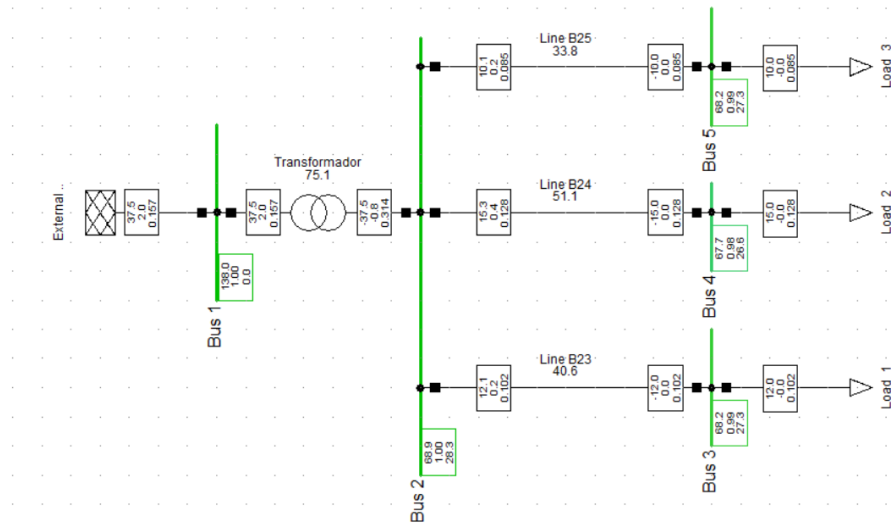


Figura 39. Caso de estudio preparado para el análisis.

En la Figura 40 se muestra la lista de las barras con los nombres modificados para que la herramienta informática pueda identificarlos y extraer los datos de simulación.

Name	Grid	Type TypBar	Zone ElmZone	Area ElmArea	Out of Service	System Type	Usage	Phase Technology	Nom.L-L Volt. kV	Nom.L-G Volt. kV	Negative Voltage	Position on Line km
BarraA	Grid				<input type="checkbox"/>	AC	Busbar	ABC	138	79.67434	<input type="checkbox"/>	0.
BarraB	Grid				<input type="checkbox"/>	AC	Busbar	ABC	138	79.67434	<input type="checkbox"/>	0.
BarraC	Grid				<input type="checkbox"/>	AC	Busbar	ABC	138	79.67434	<input type="checkbox"/>	0.
BarraD	Grid				<input type="checkbox"/>	AC	Busbar	ABC	69	39.83717	<input type="checkbox"/>	0.
BarraE	Grid				<input type="checkbox"/>	AC	Busbar	ABC	69	39.83717	<input type="checkbox"/>	0.
BarraF	Grid				<input type="checkbox"/>	AC	Busbar	ABC	69	39.83717	<input type="checkbox"/>	0.
BarraG	Grid				<input type="checkbox"/>	AC	Busbar	ABC	69	39.83717	<input type="checkbox"/>	0.

Figura 40. Lista de barras con nombres modificados.

A continuación, se procede a realizar la verificación de la lista de relés existentes en el circuito eléctrico con sus respectivos nombres, como se muestra en la Figura 41 se tiene un total de 8 relés.

Name	In Folder	Grid	IEC Symbol	ANSI Symbol	Measure Type	Type Typloc*	Out of Service	Tripping Direction	Pickup Current p.u.	Pickup Current sec.A	Pickup Current pri.A	Time
loc>>	Relay_THV	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	35	35	1750.	
loc>>	Relay_Load2	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	28.	28.	840.	
loc>>	Relay_Load3	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	29.	29.	870.	
loc>>	Relay_L24	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	53.	53.	2120.	
loc>>	Relay_L25	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	55.	55.	2200.	
loc>>	Relay_L23	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	58.	58.	2320.	
loc>>	Relay_TLV	Grid	I>>	50	Phase Current (loc>>		<input checked="" type="checkbox"/>	None	0.01	0.01	0.9	
loc>>	Relay_Load1	Grid	I>>	50	Phase Current (loc>>		<input type="checkbox"/>	None	31.	31.	930.	

Figura 41. Lista de relés existentes en el caso de estudio.

Posteriormente se procede a la identificación del tipo de relé instalado en el circuito, en este caso se tiene relés del tipo ANSI/IEEE, como se muestra en la Figura 42.

Name	In Folder	Grid	Object modified	Object modified by	Usage	Range	Range	Unit	Min. Current I/lp	Max. Current I/lp	Min. Tripping Time s	Min. Time	Max. Time
ANSI/IEEE I squared T	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE definite inver	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE extremely inver	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE long inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE moderately in	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE short inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
ANSI/IEEE very inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.5-15.0.1		s	1.1	20.	0.05	0	oo
IEC 255-3 extremely inver	Characteristics		1/2022 13:57:50	Usuario		0 0.05-3.2.0.01		s	1.1	20.	0.05	0	oo
IEC 255-3 inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.05-3.2.0.01		s	1.1	20.	0.05	0	oo
IEC 255-3 long inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.05-3.2.0.01		s	1.1	20.	0.5	0	oo
IEC 255-3 very inverse	Characteristics		1/2022 13:57:50	Usuario		0 0.05-3.2.0.01		s	1.1	20.	0.05	0	oo

Figura 42. Tipos de relés existentes en el caso de estudio.

Una vez finalizado el proceso de análisis del circuito de potencia mediante un caso de estudio se procede a la generación de un informe técnico que contiene los valores para graficar las curvas de comportamiento de todos los tipos de relés ANSI/IEEE, además se describen los valores de Dial, Pickup de todos los relés instalados en el circuito con sus respectivos. Estos datos se pueden observar en el Anexo 4.

3.2.5 Análisis de Flujo de Potencia

En la Figura 43 se muestran los resultados obtenidos en cada una de las barras con la aplicación de un análisis de flujo de potencia en DIgSILENT PowerFactory.

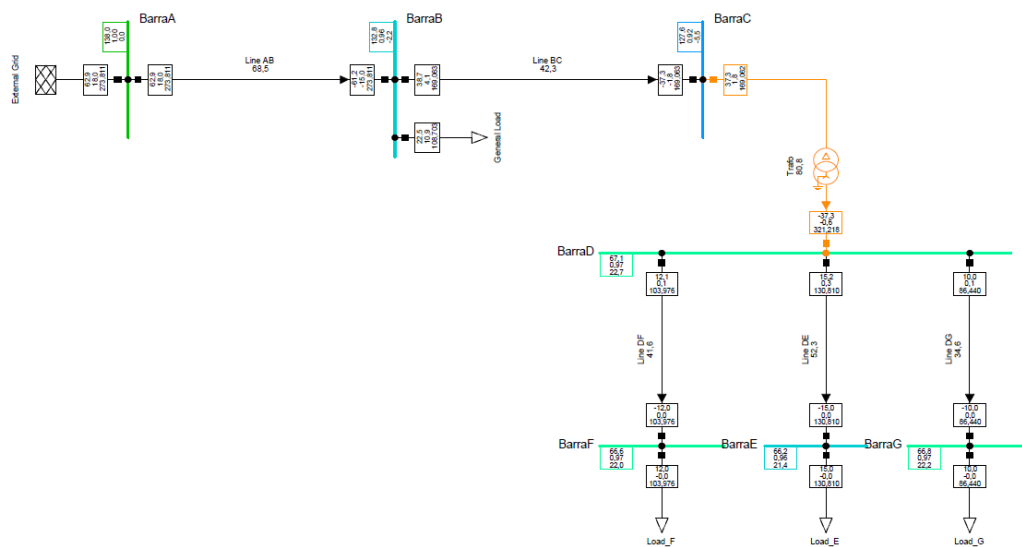


Figura 43. Análisis de flujo de potencia en DIgSILENT PowerFactory.

Por otra parte, en la Figura 44 se puede observar el resultado de la ejecución de un flujo de potencia desde la interface grafica desarrollada en este proyecto de tesis, cabe recalcar que se debe ingresar el nombre exacto del circuito eléctrico de potencia creado en DIgSILENT PowerFactory; en este caso es el nombre de “Caso de Estudio.pfd”.



Figura 44. Análisis de flujo de potencia en DIgSILENT PowerFactory.

A continuación, se procede a realizar una comparación de los valores obtenidos de los dos casos de simulación para las variables de m:UI (Voltaje Línea a Línea), m:u (Magnitud de Voltaje) y m:phiu (Angulo de Voltaje). En la Tabla 13 se muestra los valores obtenidos desde el software DIgSILENT PowerFactory para todas las barras existentes en el circuito eléctrico.

Tabla 13. Valores de flujo de potencia en DIgSILENT PowerFactory.

Flujo de Potencia en DIgSILENT PowerFactory							
	BarraA	BarraB	BarraC	BarraD	BarraE	BarraF	BarraG
m:UI (KV)	138,00	132,80	127,60	67,10	66,20	66,60	66,80
m:u (p.u.)	1,00	0,96	0,92	0,97	0,96	0,97	0,97
m:phiu (deg)	0,00	-2,20	-5,50	22,70	21,40	22,00	22,20

En la Tabla 14 se muestran los valores obtenidos desde la interface grafica de análisis de circuitos eléctricos de potencia, cabe recalcar que estos valores tienen una ligera diferencia con la tabla anterior.

Tabla 14. Valores de flujo de potencia en la Herramienta informática.

Flujo de Potencia en la Herramienta informática							
	BarraA	BarraB	BarraC	BarraD	BarraE	BarraF	BarraG
m:UI (KV)	138,00	132,78	127,63	67,11	66,20	66,63	66,79
m:u (p.u.)	1,00	0,96	0,92	0,97	0,96	0,97	0,97
m:phiu (deg)	0,00	-2,22	-5,53	22,66	21,35	21,97	22,20

Aplicando la formula Ec. (10) correspondiente al cálculo del error relativo se puede cuantificar la diferencia de los datos obtenidos entre la interface grafica de desarrollada en Python con el software PowerFactory.

$$\%Error = \frac{|error_{real} - error_{medido}|}{error_{real}} \quad \text{Ec. (10)}$$

Donde:

$\%Error$: Porcentaje del error existente en la obtención de datos.

$error_{real}$: Valores de simulación obtenido en PowerFactory.

$Error_{medido}$: Valores de simulación obtenido en Python.

Los resultados del cálculo de error se muestran en la Tabla 15, donde se puede observar un valor máximo de 0.91% de error entre las mediciones realizadas y un valor mínimo de 0%.

Tabla 15. Error de Flujo de Potencia.

Error Flujo de potencia							
	BarraA	BarraB	BarraC	BarraD	BarraE	BarraF	BarraG
Error m:UI	0,00	0,02	0,02	0,01	0,00	0,05	0,01
Error m:u	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Error m:phiu	0,00	0,91	0,55	0,18	0,23	0,14	0,00

Otra forma de representar el error relativo existente entre las mediciones es a través de un gráfico, como es muestra en la Figura 45.

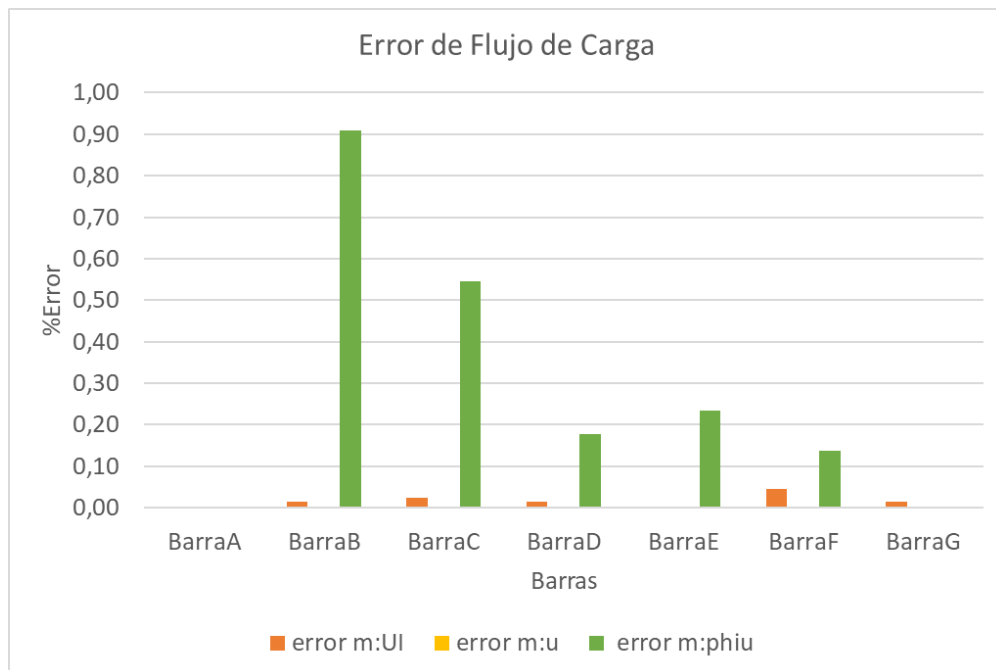


Figura 45. Representación del error relativo obtenido.

De la misma manera se procede a realizar un análisis de corto circuito en cada una de las barras existentes en el diagrama eléctrico de potencia. En la Figura 46 se muestra los resultados de la simulación de un corto circuito en la Barra C.

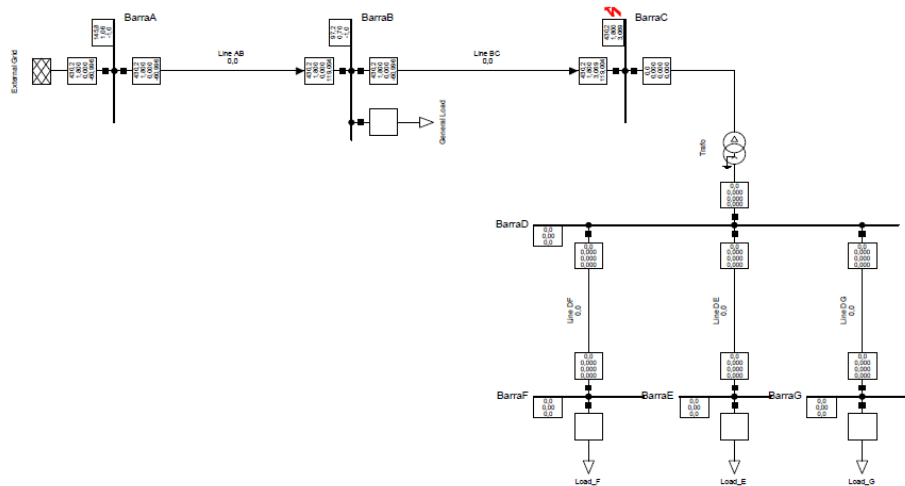


Figura 46. Análisis de Corto Circuito en la Barra C.

Este proceso anterior se repite para cada una de las Barras desde la A hasta la G, obteniendo los valores de las variables $m:SK_{ss}$ (Potencia de cortocircito inicial), $m:IK_{ss}$ (Corriente de cortocircito inicial), $m:ip$ (Corriente máxima de cortocircito), los resultados se pueden observar en la Tabla 16.

Tabla 16. Análisis de cortocircito en PowerFactory.

Corto Circuito DIgSILENT PowerFactory							
	BarraA	BarraB	BarraC	BarraD	BarraE	BarraF	BarraG
m:SK_{ss} (MVA)	10000,00	1195,60	430,20	333,90	223,50	251,20	264,30
m:IK_{ss} (kA)	41,84	5,00	1,80	2,79	1,87	2,10	2,21
m:ip (kA)	103,31	8,70	3,07	5,18	3,35	3,80	4,01

En el caso de la Tabla 17 se muestra los valores correspondientes a la simulación de un corto circuito en cada una de las barras, pero accionado desde la interface gráfica.

Tabla 17. Análisis de cortocircito en herramienta informática.

Corto Circuito Herramienta informática							
	BarraA	BarraB	BarraC	BarraD	BarraE	BarraF	BarraG
m:SK_{ss} (MVA)	10000,00	1195,57	430,23	333,88	223,50	251,23	264,34
m:IK_{ss} (kA)	41,84	5,00	1,80	2,79	1,87	2,10	2,21
m:ip (kA)	103,30	8,70	3,07	5,17	3,35	3,80	4,01

Aplicando la formula Ec. (10) correspondiente al error relativo para los valores obtenidos de cortocircito en PowerFactory y en la interface grafica se obtienen los

valores representados en la Tabla 18, donde se identifica un valor máximo de error del 0.02% y un mínimo del 0%.

Tabla 18. Error de Cortocircuito.

Error Corto Circuito							
	BarraA	BarraB	BarraC	BarraD	BarraE	BarraF	BarraG
Error m:SKss	0,00	0,00	0,01	0,01	0,00	0,01	0,02
Error m:Ikss	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Error m:ip	0,00	0,00	0,00	0,10	0,00	0,00	0,00

En la Figura 47 se muestra la representación gráfica del error relativo en las mediciones de cortocircuito.

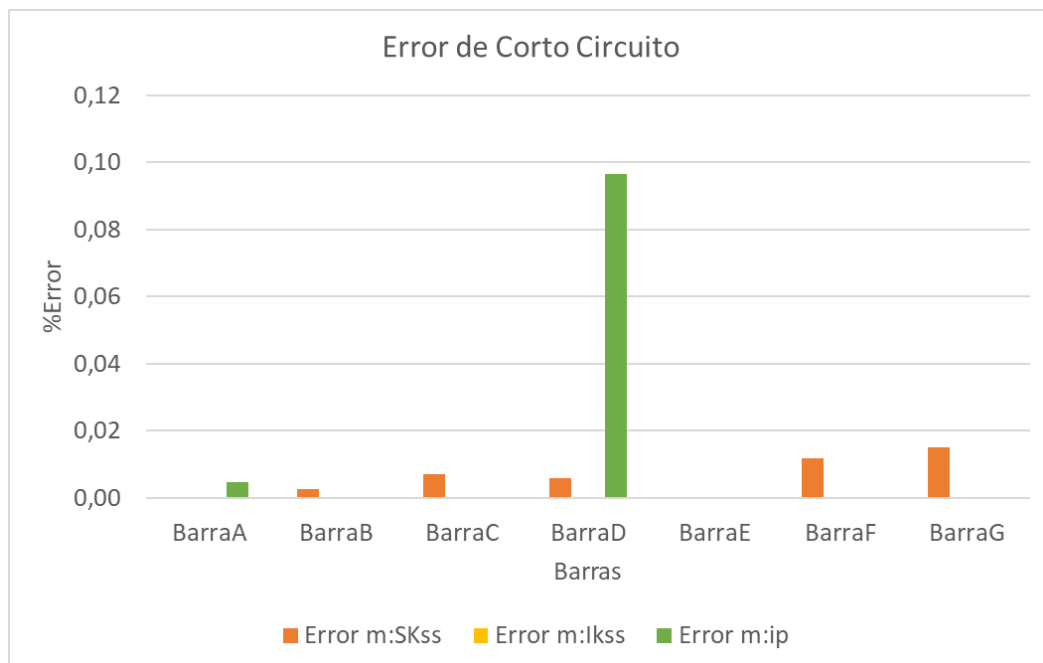


Figura 47. Representación del error relativo obtenido.

La lectura de los valores correspondientes al Pickup y Dial de los relés de sobrecorriente inicia desde la lectura de su nombre en el circuito eléctrico de potencia, como se muestra en la Figura 41. Posteriormente, se procede aplicar el proceso de lectura y se obtienen los valores representados en la Tabla 19.

Tabla 19. Valores de sobrecorriente

Valores de Relés de Sobrecorriente			
	Dial	Pickup	Prim A
Relay_THV	0,119999997	35	1750

Relay_Load2	0,050000001	28	840
Relay_Load3	0,050000001	29	870
Relay_L24	0,079999998	53	2120
Relay_L25	0,090000004	55	2200
Relay_L23	0,090000004	58	2320
Relay_TLV	0,079999998	0,01	0,899999976
Relay_Load1	0,050000001	31	930

Una vez obtenido los valores de simulación se procede aplicar las fórmulas del comportamiento de cada uno de los relés del tipo ANSI/IEE (Figura 42) obteniendo las gráficas representadas en la Figura 48.

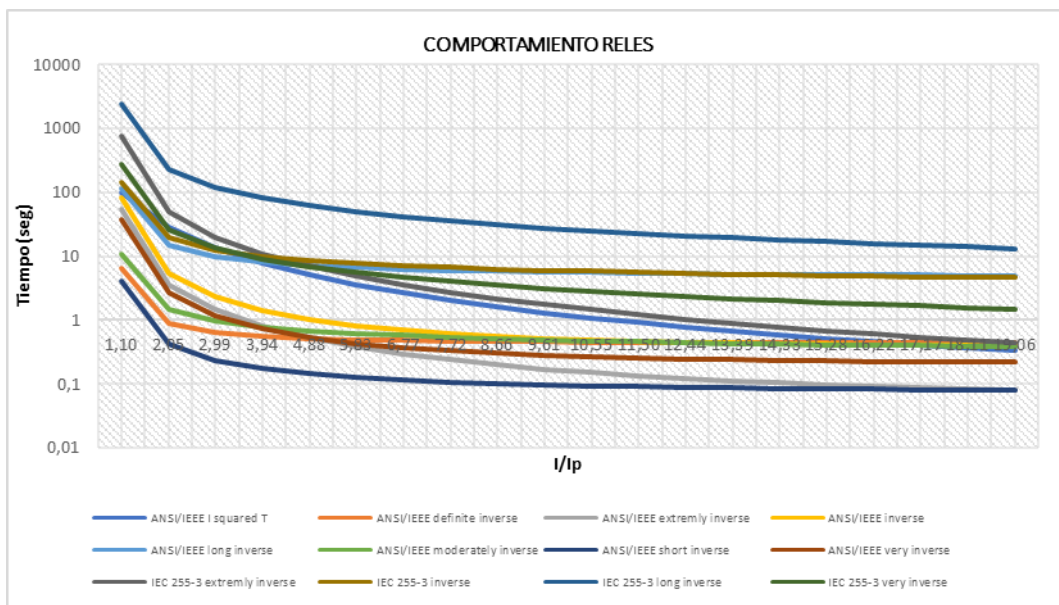


Figura 48. Comportamiento de los relés de sobrecorriente ANSI/IEEE.

El proceso de coordinación de relés se realizó comparando las gráficas de coordinación de los relés en el software DIGSILENT PowerFactory y las curvas generadas por la herramienta informática desarrollado en el presente proyecto de tesis.

a) Coordinación 1: Relay_Load1 vs Relay_L23 en la línea de transmisión B23

En la Figura 49 se muestra los parámetros de funcionamiento del Relay_Load1, que es un relé de sobrecorriente 50/51, además se puede seleccionar el tipo de relé que se requiera.

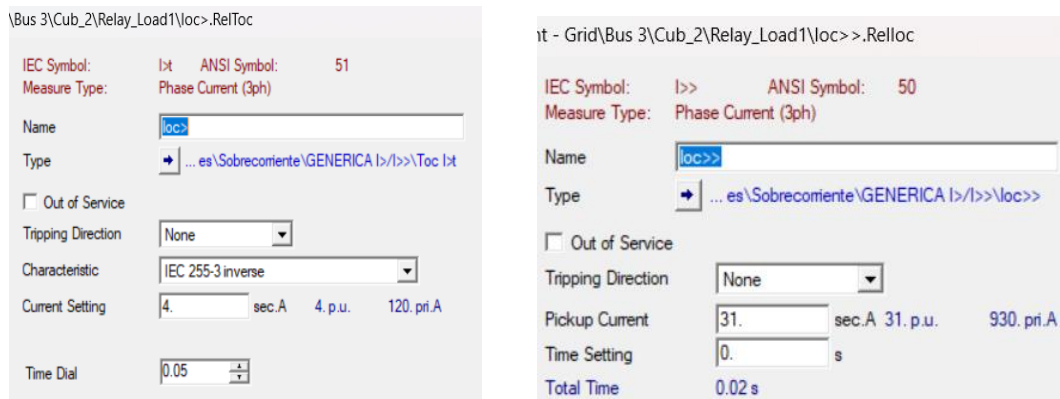


Figura 49. Datos del Relay_Load1 en la línea B23

En la Figura 50 se muestra los datos de configuración de nuestro TC del Relay_Load1, teniendo como corriente máxima de operación en el primario de 150 A y en el secundario una corriente máxima operación de 5A.

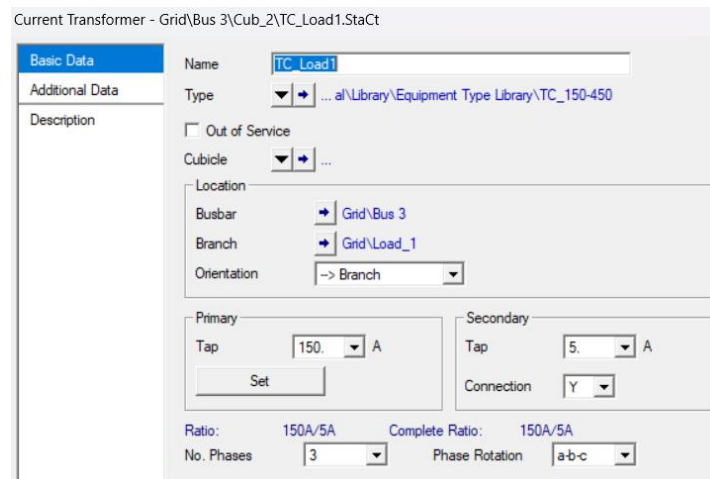


Figura 50. Configuración del TC para el Relay_Load1

En la Figura 51 se muestra los parámetros de funcionamiento del Relay_L23, que es un relé de sobrecorriente 50/51, aquí se puede seleccionar el tipo de relé que se requiera.

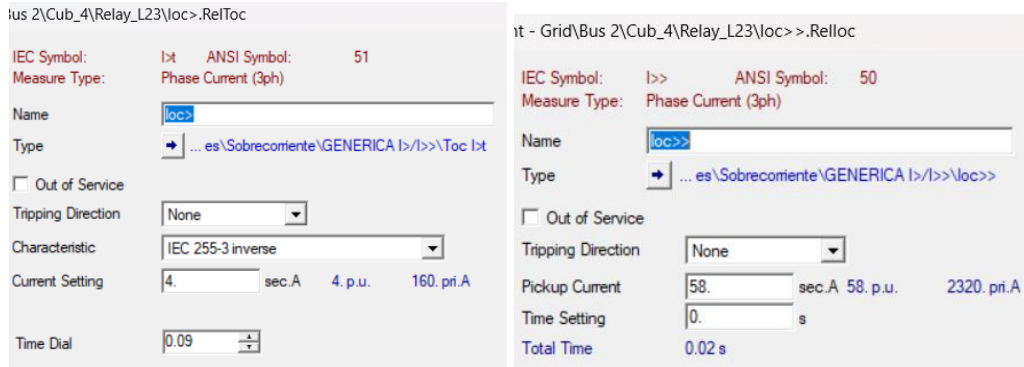


Figura 51. Datos del Relay_L23 en la línea B23

En la Figura 52 se muestra los datos de configuración de nuestro TC del Relay_L23, teniendo como corriente máxima de operación en el primario de 200 A y en el secundario una corriente máxima operación de 5A.

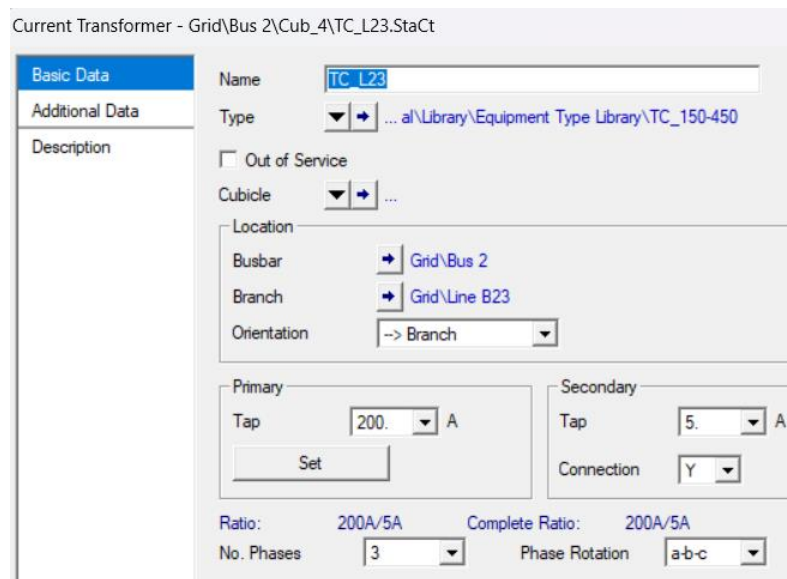


Figura 52. Configuración del TC para el Relay_L23

Una vez configurados los elementos generamos una falla en la línea B23 En esta sección se puede observar que el Relay_L23 se activa, y una correcta coordinación

con el Relay_Load1 se presenta en una corriente de línea igual a 930A, como se muestra en la Figura 53.

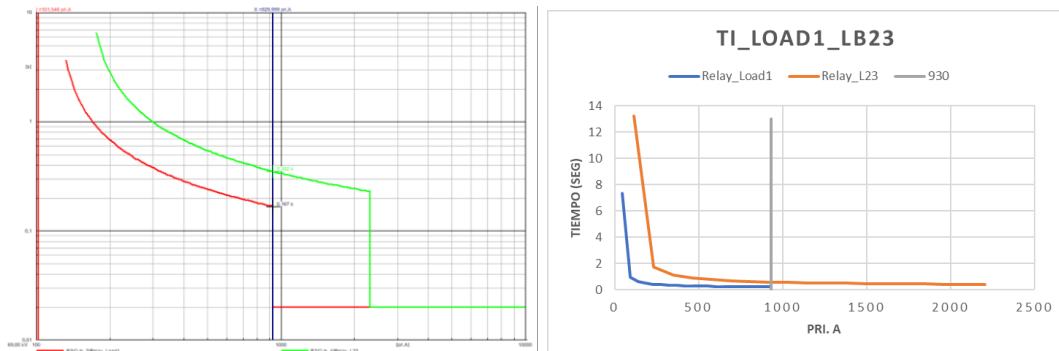


Figura 53. Coordinación 1 Relay_Load1 vs Relay_L23.

b) Coordinación 2: Relay_L24 vs Relay_load2 en la línea de transmisión B24.

Realizamos el mismo proceso de la coordinación 1 con la configuración de parámetros de los relés de sobrecorriente, instalación de los TC para la línea B24, una vez configurados los elementos generamos una falla en la línea B24 se observa que el Relay_L24 se activa, y una correcta coordinación con el Relay_load2 se presenta en una corriente de línea igual a 810A, como se muestra en Figura 54.

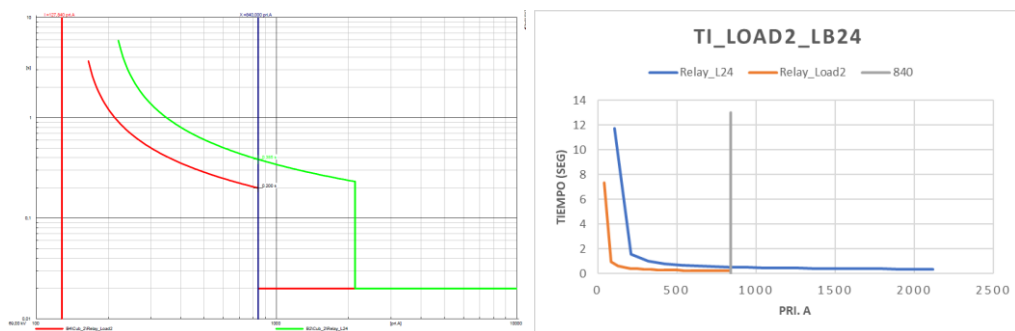


Figura 54. Coordinación 2 Relay_L24 vs Relay_Load2.

c) Coordinación 3: Relay_L25 vs Relay_Load3 en la línea de transmisión B25.

Realizamos el mismo proceso de la coordinación 1 con la configuración de parámetros de los relés de sobrecorriente, instalación de los TC para la línea B25,

una vez configurados los elementos generamos una falla en la línea B25 se observa que el Relay_L25 se activa, y una correcta coordinación con el Relay_load3 se presenta en una corriente de línea igual a 870A, como se muestra en Figura 55.

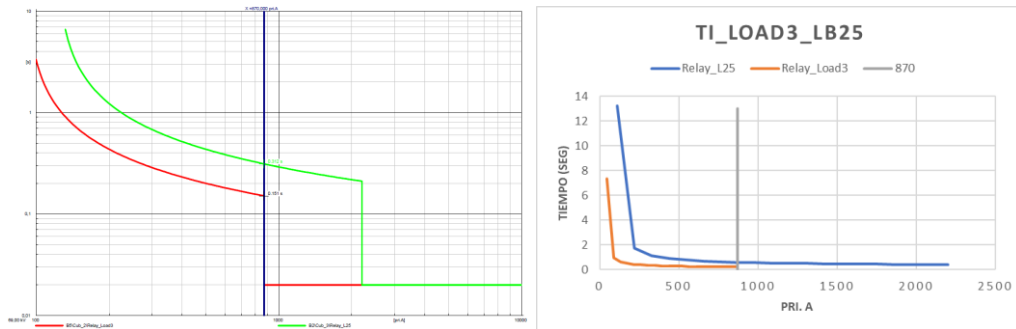


Figura 55. Coordinación 3 entre Relay_L25 vs Relay_Load3.

3.3 Conclusiones Capitulo III

En los casos de estudio aplicados se ejecutó un flujo de potencia bajo los dos métodos y se obtienen los valores de m:UI (Voltaje Línea a Línea), m:u (Magnitud de Voltaje) y m:phiu (Angulo de Voltaje); posteriormente se aplica el cálculo del error relativo entre la interface y DIgSILENT PowerFactory, obteniendo un valor máximo de 0.91% y un mínimo de 0%.

Los valores del Dial y Pickup de los relés de sobrecorriente son extraídos del software DIgSILENT PowerFactory, y con estos valores se aplica la ecuación matemática de cada uno de los relés para la gráfica de operación y coordinación.

Para la generación de reportes se crea un archivo Excel de manera automática exponiendo los datos de funcionamiento de los relés de sobrecorriente como es el tiempo de operación, graficas de operación y coordinación.

4 CONCLUSIONES

- En el presente documento se pudo observar el comportamiento los relés de sobrecorriente del tipo ANSI/IEEE ante la presencia de una falla con sus respectivas configuraciones en tiempo inverso o tiempo instantáneo. La coordinación de los relés se realizó utilizando las fórmulas de comportamiento para relés del tipo ANSI/IEEE y los valores obtenidos durante la simulación como el pickup y dial para cada uno de ellos; posteriormente y mediante una hoja de cálculo se procedió a grabar los valores de las curvas de operación. Adicionalmente, la herramienta informática permite obtener el nombre y el número de barras instaladas y relés con sus respectivas configuraciones.
- El proceso de comunicación de la herramienta informática con DIGSILENT PowerFactory permite realizar diferentes análisis de los casos de estudios que se requiera, y esto permite sintetizar el proceso de extracción de datos de los diferentes elementos de nuestro sistema eléctrico y nos ahorra tiempo.
- La herramienta informática desarrollada permite extraer datos de las protecciones eléctricas aplicadas en los distintos sistemas eléctricos realizados en DIGSILENT PowerFactory, y generan un reporte con los tiempos de operación, graficas de comportamiento y coordinación de los relés de sobrecorriente.
- El error obtenido durante la validación de nuestra herramienta se obtuvo un máximo error de 0.91% y mínimo de 0% en el flujo de potencia; por otro lado, en el estudio de corto circuito el error máximo fue de 0.02% y el mínimo de 0%.

5 RECOMENDACIONES

- Identificar adecuadamente las versiones de DIgSILENT PowerFactory y Python que sean compatibles para poder realizar la comunicación entre ellos.
- Extraer los datos del software DIgSILENT PowerFactory de forma correcta, con el fin de evitar inconvenientes en el proceso de análisis.
- Trabajar con versiones de softwares con licencia para obtener mejores resultados de la interfaz gráfica y de la generación de reportes.
- Se recomienda realizar un proceso de instalación de las versiones de softwares, librerías para un correcto funcionamiento de la herramienta informática.
- Los archivos generados en Excel se extraen en la carpeta en donde se encuentra instalado Python, por lo que es necesario identificar la ubicación de los archivos.

6 BIBLIOGRAFÍA

- [1] P. H. Analuisa Paul, «Dimensionamiento e implementación de un módulo didáctico para el análisis de curvas de operación en los relés electromecánicos de sobrecorriente inverso e instantáneo (50/51).», Universidad de las Fuerzas Armadas Espe, Latacunga, 2020.
- [2] A. T, «Coordination of Directional Overcurrent Relays Using Seeker Algorithm,» 2013.
- [3] R. A.H, «New Method on the Relays Protective Coordination Due To Presence Of Distributed Generation,» 2015.
- [4] V. S. M. Araceli, «Estudio de coordinación de protecciones en líneas de transmisión de la zona norte considerando la influencia del estado de operación del sistema de 500kv,» EPN, Quito, 2018.
- [5] C. Doñez, «Selección y análisis de equipo de protección en los sistemas de distribución,» Universidad Autonoma de Mexico, Mexico, 2015.
- [6] J. E. Estrada Centeno, «Diseño del sistema de protecciones para el sistema eléctrico de la Empresa Regional de Sucumbíos S.A,» Repositorio Digital EPN – FIEE, Quito, 2001.
- [7] J. A. Sueiro Dominguez, «“Contribución a la coordinación de los relés de protección de redes eléctricas,» Universidad de Vigo Febrero, 2006.
- [8] Pacarenca, «Estudio de coordinación protecciones de la central hidroeléctrica,» 2003.
- [9] C. E. Soto Torres, «“Estudio de coordinación y ajuste de protecciones en tres circuitos de la subestación Turén en 13.8 kV,» Sartenejas, 2005.

- [10] A. M. N. Cella, «“Estudio de coordinación de protecciones CH LOS Hierros II,» Proyecto EE-2014-073 Informe técnico EE-ES-2014-0597 Revisión C. Estudios eléctricos.
- [11] E. C. Guambo Coello, «Análisis de calidad de voltaje en barras del S.N.I. en estado estable considerando el sistema de 500 kV,» Repositorio Digital EPN, Quito, 2014.
- [12] R. Zambrano, «Sistema de protecciones eléctricas a nivel de 500kV aplicación subestación Pifo,» Repositorio Digital EPN-FIEE, Quito, 2009.
- [13] R. Tamasco Amador, «Protecciones eléctricas,» 2007.
- [14] Electromecanica, «Electromecanica,» 2011. [En línea]. Available: https://profesormolina2.webcindario.com/electromec/prot_circ_elect.htm.
- [15] A. Rodríguez Penin, Sistemas SCADA, vol. Tercera, México D.F.: Alfaomega-Marcombo, 2013, p. 472.
- [16] S. Castaño, Protección de Sistemas Electricos, Colombia: Universidad Nacional de Colombia, 2003.
- [17] C. P. Systems, «Electrical Distribution System Protection,» Desconocida, 2005.
- [18] G. E. Mejía, «Coordinación de relés de sobrecorriente en sistemas radiales,» Universidad Tecnologica de Pereira, Pereira, 2010.
- [19] A. Lazo, «Estudio de coordinación de protecciones en alta y media tensión en la subestación “Machala” propiedad de la corporación nacional de electricidad S.A. - regional El Oro,» Ecuador, 2010.

- [20] R. D. J. A. Ramos, «Metodologías para el cálculo de las corrientes de cortocircuito,» Universidad Tecnológica de Bolívar, Colombia, 2011.
- [21] J. Nuñez, «Comparación Técnica entre los programas de Simulación de Sistemas de Potencia Digsilent PowerFactory y PSS/E,» Revista técnica de energía, 2015.
- [22] M. O. Lucero, «El lenguaje de programación Python/The programming language Python,» Ciencias Holguin, Cuba, 2014.
- [23] F. Arturo, «Python 3 al descubierto,» RC Libros, España, 2012.
- [24] M. E. i. V. S. Code, «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com/docs/editor/extension-marketplace>. [Último acceso: 02 Diciembre 2022].
- [25] D. PowerFactory, *39 Bus New England System*, 1990.
- [26] IMSE, «Ingeniería de máquinas y sistemas eléctricos,» 18 noviembre 2016. [En línea]. Available <http://imseingenieria.blogspot.com/2016/11/proteccion-contra-sobrepotencias-y.html>.
- [27] M. I. -. NOROÑA, «Planificación de expansión, estudio de coordinación y ajuste de protecciones del sistema occidental de ELEPCO S.A. (tesis de pregrado),» Universidad Técnica de Cotopaxi, Latacunga- Ecuador, 2013.
- [28] S. A. C. Correa, «Coordinación automática de protecciones de sobrecorriente para un sistema de distribución ante desastres naturales,» Quito, 2018.
- [29] M. L. Rodríguez U., «Investigación bibliográfica y documental,» 14 Mayo 2019. [En línea]. Available

<https://guiadetesis.wordpress.com/tag/investigacion-bibliografica-y-documental/>.

- [30] E. Bonilla, «Coordinación de protecciones del alimentador n.- 3 de La Mana a 13.8 Kv para elepco SA,» UTC, Latacunga, 2022.
- [31] L. C. Sañudo, «Selección y Especificación de Transformadores de Corriente y Potencial en Instalaciones Industriales,» Mexico DF, 2014.
- [32] T. S. insights, «Ejecute DIgSILENT PowerFactory a través de la API de Python: inicie la automatización de su PowerFactory,» 2021. [En línea]. Available: <https://thesmartinsights.com/run-digsilent-powerfactory-via-the-python-api-jump-start-to-your-powerfactory-automatization/>. [Último acceso: 03 12 2022].

7 ANEXOS

Anexo 1: Instalación PowerFactory 15.1

El proceso de instalación de PowerFactory 15.1 inicia con la ejecución del instalador, en la **Figura 56** se observa el asistente de DIgSILENT con las versiones que ofrece actualmente. En este caso se opta por la opción “*PowerFactory WorkStation*” que permite un acceso a todas las librerías de relés de protección de sobrecorriente y al diseño de cualquier tipo de circuitos eléctricos de alta potencia.

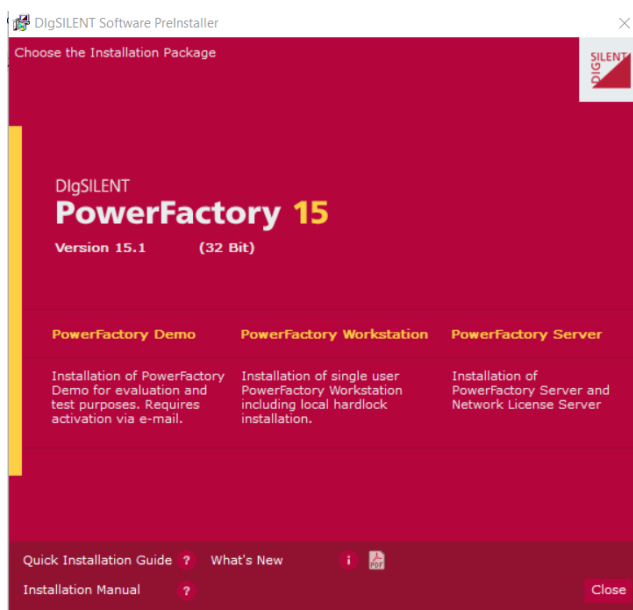


Figura 56. Pantalla principal de instalación PowerFactory.

Cabe recalcar que el programa de PowerFactory requiere de varias instalaciones previas como complementos, necesarios para un correcto funcionamiento. En la **Figura 57** se observa uno de los requisitos de instalación.

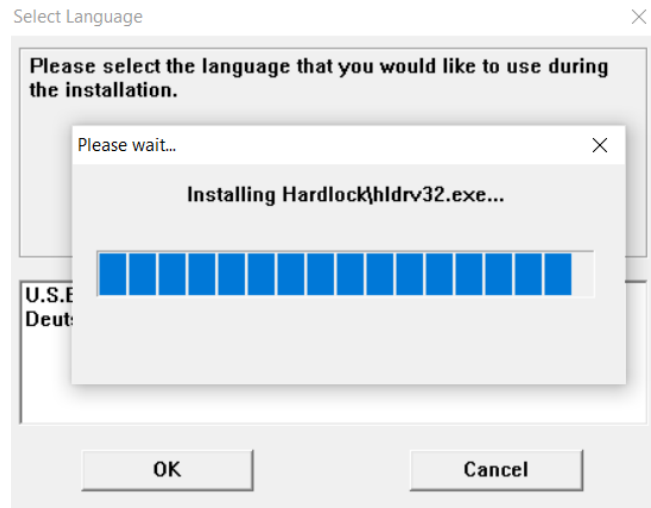


Figura 57. Instalación de completos de PowerFactory.

A continuación, se procede a la instalación de la licencia de servidor 6.1 gratuita; esta licencia permite la utilización de todas las librerías de simulación de circuitos eléctricos de alta potencia. En la **Figura 58** se muestra el proceso de instalación de la licencia de servidor para un uso continuo de PowerFactory.

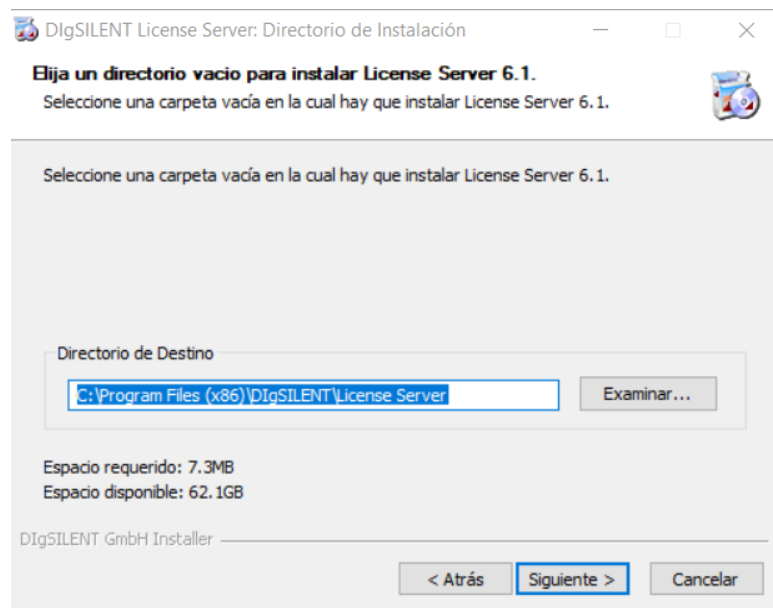


Figura 58. Instalación de licencias de servidor.

En la **Figura 59** se muestra la pantalla de selección del idioma de instalación de PowerFactory, cabe recalcar que esta sección es parte del proceso de configuración del software.

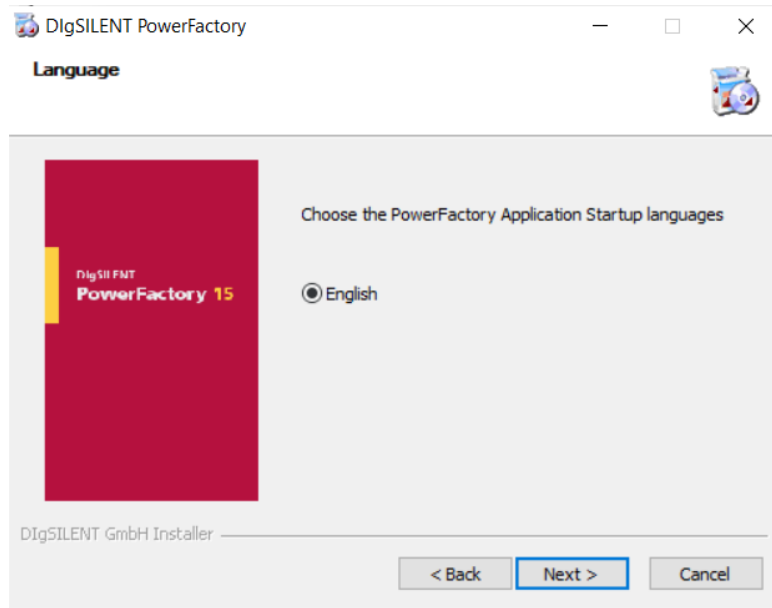


Figura 59. Selección del idioma en PowerFactory.

Una vez que se ha realizado todos los pasos de instalación correctamente, se procede a ejecutar el software PowerFactory, como se muestra en la **Figura 60**.

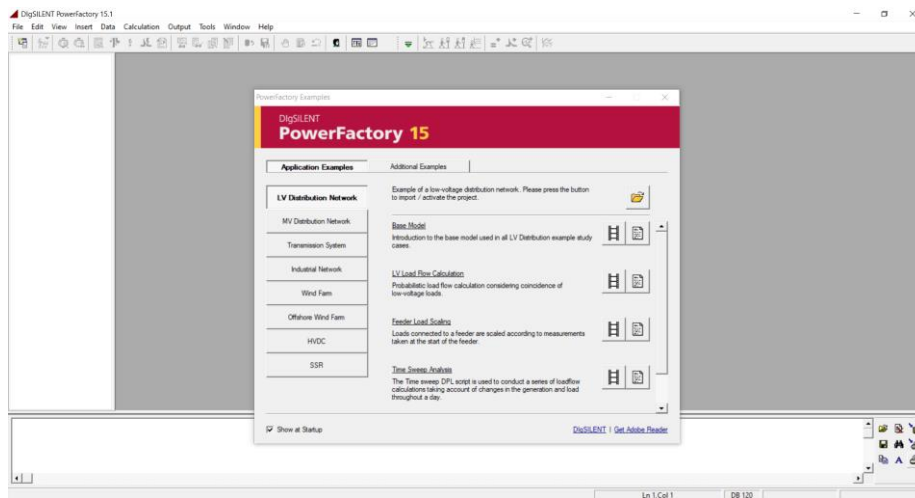


Figura 60. Software instalado de PowerFactory.

Anexo 2: Instalación Python 3.3.2

La versión de Python compatible con PowerFactory y que permite acceder a los comandos de simulación de circuitos eléctricos de forma gratuita es la 3.3.2. Para lo cual se debe acceder al siguiente enlace:

C:\Program Files (x86)\DIgSILENT\PowerFactory 15.1\python

En esta dirección se encuentra el instalador por defecto de Python que ofrece el paquete de DIgSILENT, En la **Figura 61** se puede observar el proceso de inicio de Python.



Figura 61. Instalador de Python 3.3.2.

En la **Figura 62** se puede observar la finalización del proceso de instalación de Python 3.3.2 .



Figura 62. Proceso de instalación terminado.

Generalmente Windows no reconoce directamente la funcionalidad de Python para aplicar líneas de código, para lo cual se debe agregar un PATH. En la **Figura 63** se observa las propiedades del sistema donde se debe escoger las Variables de Entorno.

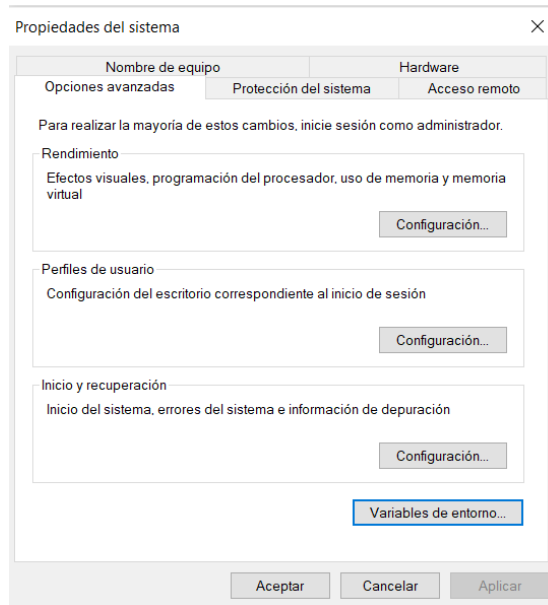


Figura 63. Acceso de Propiedades del Sistema.

Una vez dentro de las Variables de Entorno se debe proceder a editar las existentes, como se muestra en la **Figura 64**.

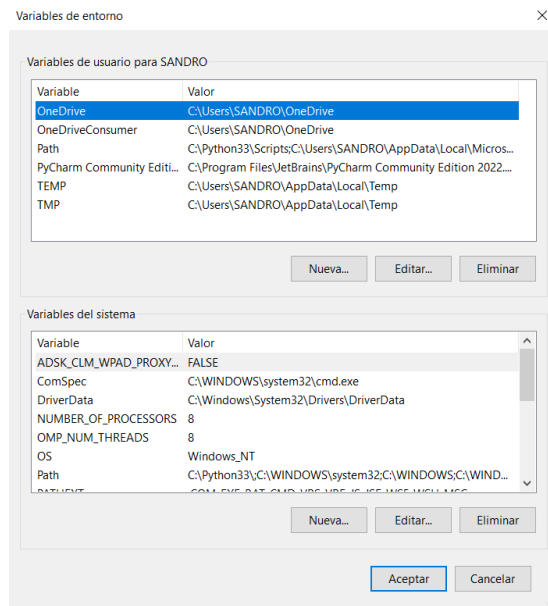


Figura 64. Variable de entorno.

Finalmente, se agrega la dirección donde se instaló Python, este en lace se muestra en la **Figura 65**.

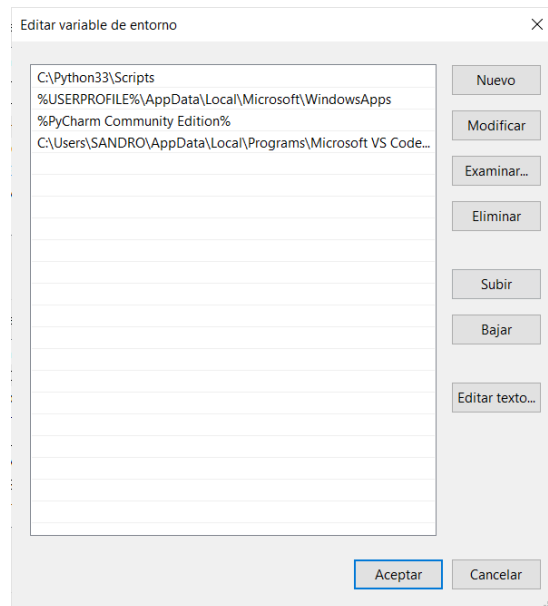


Figura 65. Enlace de PATH de Python.

Anexo 3: Instalación Visual Studio Code

Es importante la utilización de un intérprete para el lenguaje de programación basado en Python. En este caso se utilizó el recomendado por Visual Studio Code (2015) que dice:

Es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET) (pag. 10).

En la **Figura 66** se observa el proceso inicial de instalación de Visual Studio Code, donde se muestra la aprobación de la licencia del software de Microsoft.

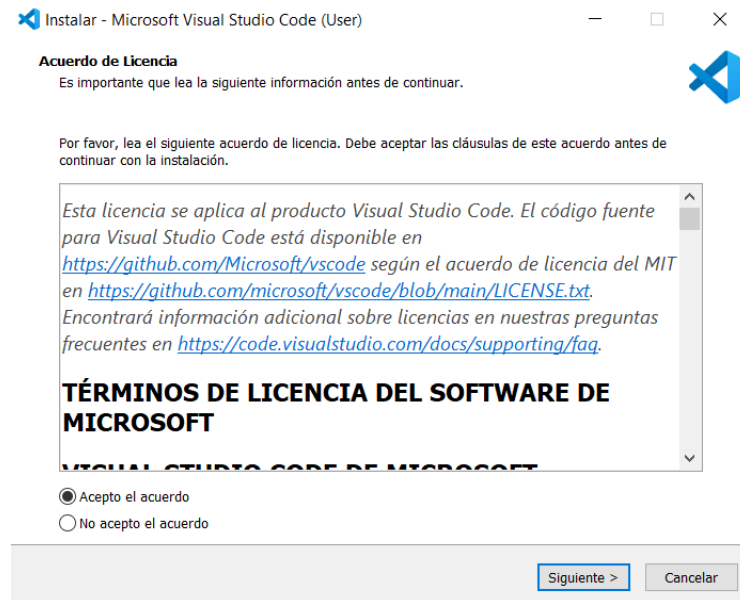


Figura 66. Inicio de la instalación de Visual Studio Code.

En la **Figura 67** se muestra la etapa de agregación del PATH para Visual Studio Code de forma automática, de esta manera el software puede reconocer directamente la funcionalidad de Python en Windows.

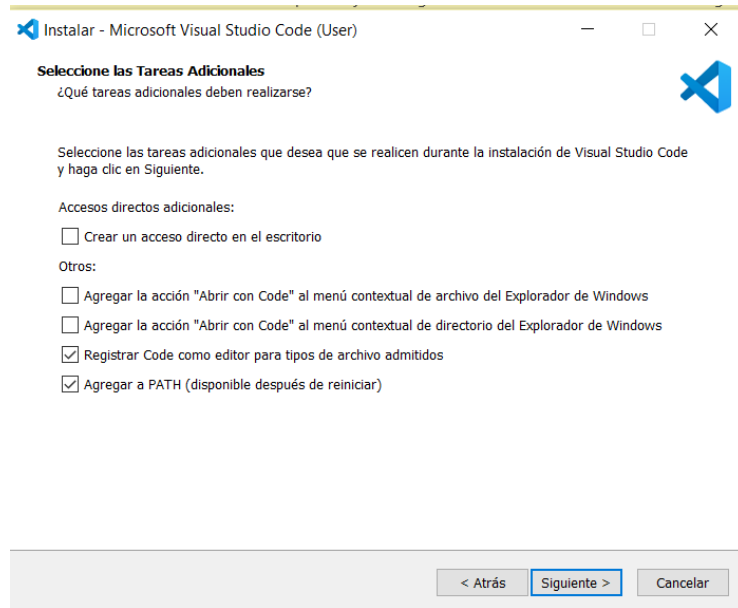


Figura 67. Agregar PATH en Visual Studio Code.

En la **Figura 68** se muestra la pantalla final de instalación de Visual Studio Code, cabe recalcar que después de finalizar este proceso se ejecuta directamente este software.

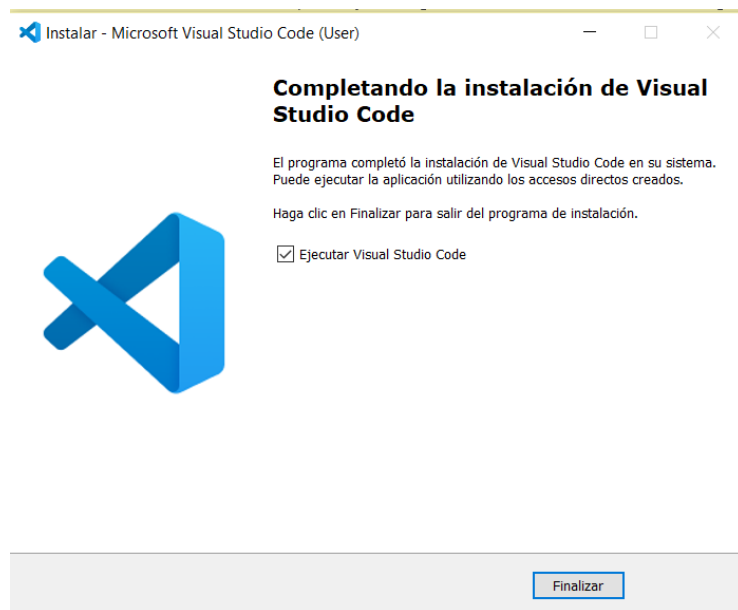
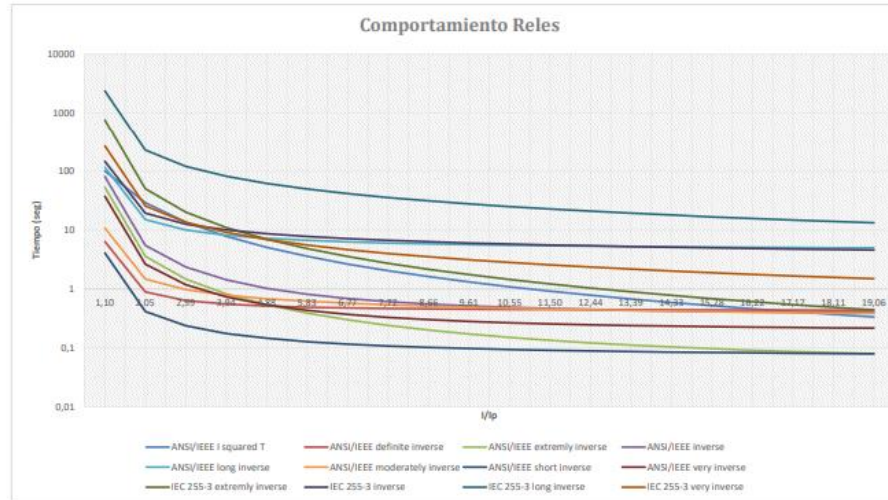


Figura 68. Instalación terminada con éxito de Visual Studio Code.

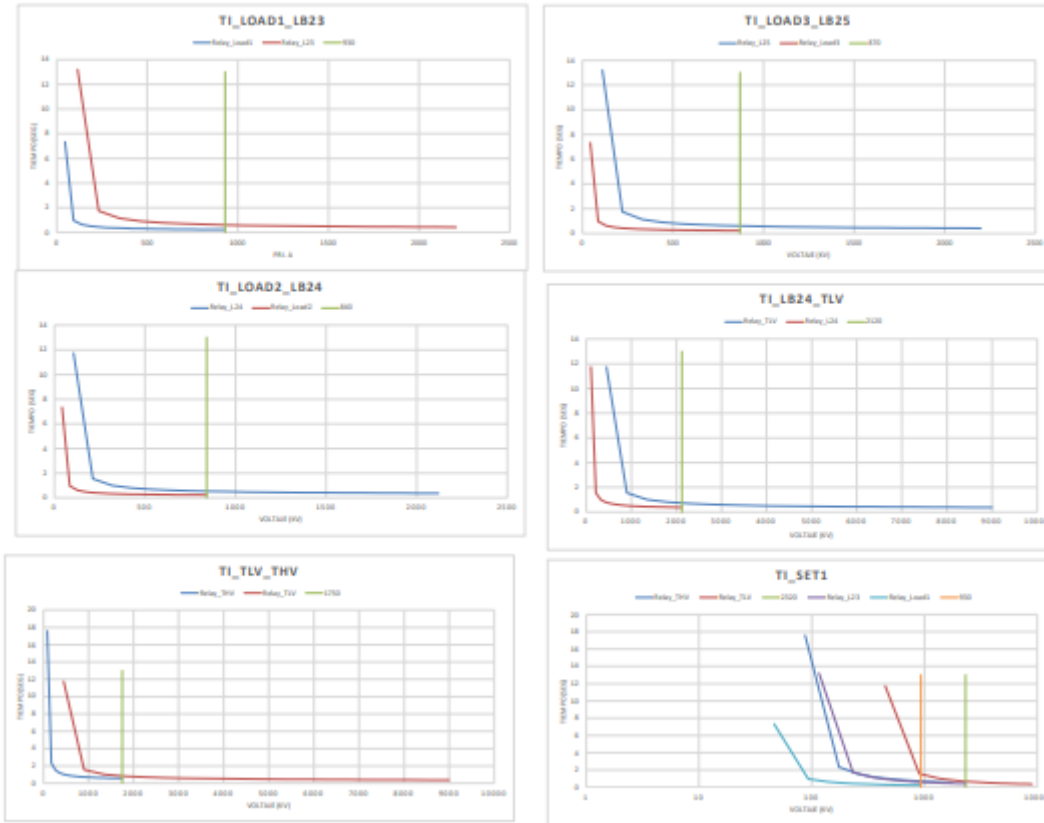
Anexo 4: Informe de coordinación de relés

Corriente	Valores de las curvas del rele ANSI/IEEE											
	ANSI/IEEE I squared T	ANSI/IEEE definite inverse	ANSI/IEEE extremely inverse	ANSI/IEEE inverse	ANSI/IEEE long inverse	ANSI/IEEE moderately inverse	ANSI/IEEE short inverse	ANSI/IEEE very inverse	IEC 255-3 extremely inverse	IEC 255-3 inverse	IEC 255-3 long inverse	IEC 255-3 very inverse
1.100000024	100.5619791	6.401672372	53.7628723	81.25993116	116.6578132	10.84212126	4.115834294	37.54877162	761.9045716	146.7488327	2399.999428	269.9999356
2.045000023	29.09595168	0.89332754	3.593512335	5.505265571	15.11691154	1.475082972	0.416206394	2.66149684	50.28244446	19.42986564	229.6650668	25.83732001
2.990000021	13.61058582	0.638664071	1.469236997	2.365311711	10.0143525	0.975742078	0.237530561	1.184296862	20.15087939	12.64270785	120.6030138	13.56783905
3.93500002	7.858320242	0.555039185	0.827378283	1.435237797	8.197598066	0.787220757	0.176333602	0.737954685	11.04650047	10.08028213	81.77172005	9.199318505
4.880000019	5.109513532	0.515174232	0.54302457	1.030247935	7.265809058	0.685137158	0.14603535	0.540217937	7.013114467	8.692738071	61.8556698	6.958762852
5.825000018	3.586140817	0.492465334	0.391138289	0.817136513	6.699010976	0.61987075	0.128183475	0.434597725	4.858699128	7.805621845	49.74093246	5.595854902
6.770000017	2.654861451	0.478072321	0.300200944	0.691204405	6.31787119	0.573927446	0.116520964	0.371360798	3.56880771	7.181149747	41.59445395	4.67937607
7.715000015	2.044312756	0.468271403	0.241350578	0.610649193	6.044006787	0.539489613	0.108358938	0.330436838	2.734050754	6.713062706	35.74087855	4.020848837
8.660000014	1.622495181	0.461242764	0.201041496	0.556045753	5.837714671	0.512509878	0.102357767	0.302406303	2.162290724	6.346347855	31.33159263	3.524804171
9.605000013	1.318938245	0.456000514	0.172208276	0.517354543	5.676732502	0.490669795	0.09777822	0.282355968	1.753308886	6.04942362	27.89076114	3.137710629
10.55000001	1.093236897	0.45196827	0.150864228	0.488958606	5.547609632	0.472539201	0.094180487	0.267513529	1.450556421	5.803057099	25.13089002	2.827225127
11.49500001	0.9208762	0.448788646	0.134618209	0.46751543	5.441739951	0.45718443	0.091287288	0.256216208	1.220116439	5.594351475	22.86803237	2.572653642
12.44000001	0.786282192	0.44622941	0.121964136	0.450935076	5.353360978	0.443968014	0.088915576	0.247416692	1.040626039	5.41471087	20.97902096	2.360139858
13.38500001	0.679176185	0.444133758	0.111914506	0.437856571	5.278468986	0.432438609	0.086939886	0.240428279	0.898078098	5.258000508	19.37828016	2.180056519
14.33000001	0.59255248	0.442392383	0.10379971	0.427362961	5.214195588	0.422266897	0.085271496	0.23478533	0.782974608	5.119744236	18.00450112	2.02506376
15.27500001	0.521502942	0.440926996	0.097152543	0.418818314	5.158431944	0.413206443	0.083845987	0.230162957	0.688688559	4.996592436	16.81260945	1.891418563
16.22	0.462506138	0.439680207	0.091638914	0.41177046	5.109592956	0.405068812	0.082615521	0.226328833	0.610481043	4.885983851	15.76872536	1.773981603
17.165	0.41298244	0.438609095	0.087014748	0.405890838	5.066464188	0.39770721	0.081543866	0.223113234	0.544890042	4.785923241	14.84689143	1.670275286
18.11	0.371007168	0.43768098	0.083098356	0.400936109	5.028099497	0.391005414	0.080603109	0.220389814	0.489338387	4.694830873	14.02688486	1.578024547
19.055	0.335120731	0.436870594	0.079752216	0.39672298	4.993750828	0.384870082	0.079771416	0.218062941	0.441875399	4.611438012	13.2927167	1.495430629



	Valores de Reles Detectados							
	Relay_THV	Relay_Load2	Relay_Load3	Relay_L24	Relay_L25	Relay_L23	Relay_TLV	Relay_Load1
Dial	0,119999997	0,050000001	0,050000001	0,079999998	0,090000004	0,090000004	0,079999998	0,050000001
Pickup	35	28	29	53	55	58	0,01	31
Prim A	1750	840	870	2120	2200	2320	0,899999976	930

Curvas de Coordinación															
Relay_THV		Relay_Load2		Relay_Load3		Relay_L24		Relay_L25		Relay_L23		Relay_TLV		Relay_Load1	
(Amp)	(seg)	(Amp)	(seg)	(Amp)	(seg)	(Amp)	(seg)	(Amp)	(seg)	(Amp)	(seg)	(Amp)	(seg)	(Amp)	(seg)
87,5	17,60986	42	7,337442	43,5	7,337442	106	11,73991	110	13,2074	116	13,2074	450	11,73991	46,5	7,337442
175	2,331584	84	0,971493	87	0,971493	212	1,554389	220	1,748688	232	1,748688	900	1,554389	93	0,971493
262,5	1,517125	126	0,632135	130,5	0,632135	318	1,011417	330	1,137844	348	1,137844	1350	1,011417	139,5	0,632135
350	1,209634	168	0,504014	174	0,504014	424	0,806423	440	0,907225	464	0,907225	1800	0,806423	186	0,504014
437,5	1,043129	210	0,434637	217,5	0,434637	530	0,695419	550	0,782346	580	0,782346	2250	0,695419	232,5	0,434637
525	0,936675	252	0,390281	261	0,390281	636	0,62445	660	0,702506	696	0,702506	2700	0,62445	279	0,390281
612,5	0,861738	294	0,359057	304,5	0,359057	742	0,574492	770	0,646304	812	0,646304	3150	0,574492	325,5	0,359057
700	0,805568	336	0,335653	348	0,335653	848	0,537045	880	0,604176	928	0,604176	3600	0,537045	372	0,335653
787,5	0,761562	378	0,317317	391,5	0,317317	954	0,507708	990	0,571171	1044	0,571171	4050	0,507708	418,5	0,317317
875	0,725939	420	0,302475	435	0,302475	1060	0,483959	1100	0,544454	1160	0,544454	4500	0,483959	465	0,302475
962,5	0,696367	462	0,290153	478,5	0,290153	1166	0,464245	1210	0,522275	1276	0,522275	4950	0,464245	511,5	0,290153
1050	0,671322	504	0,279718	522	0,279718	1272	0,447548	1320	0,503492	1392	0,503492	5400	0,447548	558	0,279718
1137,5	0,649765	546	0,270736	565,5	0,270736	1378	0,433177	1430	0,487324	1508	0,487324	5850	0,433177	604,5	0,270736
1225	0,63096	588	0,2629	609	0,2629	1484	0,42064	1540	0,47322	1624	0,47322	6300	0,42064	651	0,2629
1312,5	0,614369	630	0,255987	652,5	0,255987	1590	0,40958	1650	0,460777	1740	0,460777	6750	0,40958	697,5	0,255987
1400	0,599591	672	0,24983	696	0,24983	1696	0,399727	1760	0,449693	1856	0,449693	7200	0,399727	744	0,24983
1487,5	0,586318	714	0,244299	739,5	0,244299	1802	0,390879	1870	0,439739	1972	0,439739	7650	0,390879	790,5	0,244299
1575	0,574311	756	0,239296	783	0,239296	1908	0,382874	1980	0,430733	2088	0,430733	8100	0,382874	837	0,239296
1662,5	0,56338	798	0,234742	826,5	0,234742	2014	0,375586	2090	0,422535	2204	0,422535	8550	0,375586	883,5	0,234742
1750	0,553373	840	0,230572	870	0,230572	2120	0,368915	2200	0,415029	2320	0,415029	9000	0,368915	930	0,230572
ValorX	ValorY	ValorX	ValorY	ValorX	ValorY	ValorX	ValorY	ValorX	ValorY	ValorX	ValorY	ValorX	ValorY	ValorX	ValorY
0	1750	0	840	0	870	0	2120	0	2200	0	2320	0	0,9	0	930
13	1750	13	840	13	870	13	2120	13	2200	13	2320	13	0,9	13	930



Anexo 5: Programación desarrollada en python

```
import xlswriter
from xlutils.copy import copy # http://pypi.python.org/pypi/xlutils
from xlrd import open_workbook # http://pypi.python.org/pypi/xlrd
from xlwt import Workbook, easyfont
import xlwt

from cgitb import text
from msilib import Table
from telnetlib import SGA
import tkinter
import tkinter as tk
from functools import partial
from tkinter import messagebox, ttk, filedialog
from tkinter import *
from PIL import Image, ImageTk

import os
import sys
import random
from re import search

os.environ["PATH"]=r"C:\Program Files (x86)\DIgSILENT\PowerFactory
15.1"+os.environ["PATH"]
sys.path.append(r"C:\Program Files (x86)\DIgSILENT\PowerFactory
15.1\python")

import powerfactory as pf
app = pf.GetApplication()

inicio = tkinter.Tk()

w = copy(open_workbook('Graficas.xlsx'))
workbook = xlswriter.Workbook('Graficas.xlsx')
ec1 = workbook.add_worksheet()
ec2 = workbook.add_worksheet()
bold = workbook.add_format({'bold': 1})
headings = ['Corriente', 'Ecuacion']
ec1.write_row('A1', headings, bold)
for n in range(12):
    ec1.set_column(1, n, 22)

list_caracteristicas = {}
```

```

vector = {}

def powerfactory_dir():
    ruta_powerfactory = filedialog.askdirectory()
    if ruta_powerfactory!="":
        rut = ruta_powerfactory
        text_powerfactory.insert(0,rut)
        os.environ["PATH"]=rut+os.environ["PATH"]

def python_dir():
    ruta_python = filedialog.askdirectory()
    if ruta_python!="":
        rut = ruta_python
        text_python.insert(0,rut)
        sys.path.append(rut)

def direccion_default():
    text_powerfactory.insert(0,"C:\Program Files
(x86)\DIGSILENT\PowerFactory 15.1")
    text_python.insert(0,"C:\Program Files
(x86)\DIGSILENT\PowerFactory 15.1\python")

def ventana_secundaria():
    listado_barras = {}
    listado_reles={}
    def activar_proyecto():
        project = app.ActivateProject(caso.get())
        prj = app.GetActiveProject()

        study_case_folder=app.GetProjectFolder('study')
        study_case=study_case_folder.GetContents('*.Intcase')[0]

        #activar caso de estudio
        for case in study_case:
            case.Activate()

        #reles
        Reles = app.GetCalcRelevantObjects('*.ElmRelay')
        for rele in Reles:
            listado_reles[rele.loc_name]=rele
            name = getattr(rele, 'loc_name')
            barras_rele_cbx['values']=list(listado_reles)
            vuno_rele=list(listado_reles)

```

```

barras_rele_cbx.set(vuno_rele[0])
cargarDatoRele()

#barras
Buses=app.GetCalcRelevantObjects('*.ElmTerm')
for bus in Buses:
    name = getattr(bus, 'loc_name')
    if search("Barra", name):
        listado_barras[bus.loc_name]=bus
barras_cbx['values']=list(listado_barras)
vuno=list(listado_barras)
barras_cbx.set(vuno[0])
cargarDatoBarra()

#Ecuaciones
caracteristicas=app.GetCalcRelevantObjects('*.TypChatoc')
j=0
for caract in caracteristicas:
    name = getattr(caract, 'loc_name')
    list_caracteristicas[caract.loc_name]=caract
    vector[j]=name
    j=j+1

def cargarDatoRele():
    ejecutar()
    listado_corriente = {}
    corriente_rele = {}
    pickup={}
    tiempos = {}
    ip = app.GetCalcRelevantObjects('*.RelIoc')
    relay=getattr(listado_reles[barras_rele_cbx.get()], 'loc_name
')

    k=0
    for IP in ip:
        ip_name = getattr(IP, 'loc_name')
        if (IP.fold_id.loc_name == relay):
            if ("Ioc I>>" == ip_name):
                listado_corriente[k]=IP.loc_name
                corriente_rele[k]=IP.fold_id.loc_name
                pickup[k]=IP.Ipsetr
                tiempos[k]=IP.Tset
                k=k+1

    name =
    getattr(listado_reles[barras_rele_cbx.get()], 'loc_name')

```



```

        dev_no =
getattr(listado_reles[barras_rele_cbx.get()], 'dev_no')
        msg_rele='Resultado:\nNOMBRE: '+name+"\nNúmero device:
"+str(dev_no)
        lb_datos_rele_obtenido.config(text=msg_rele)
        entry_corriente.delete(0,END)
        entry_corriente.insert(0,str(pickup[0]))
        entry_dial.delete(0,END)
        entry_dial.insert(0,str(tiempos[0]))

def cargarDatoBarra():
    ejecutar()
    name = getattr(listado_barras[barras_cbx.get()], 'loc_name')
    ui = getattr(listado_barras[barras_cbx.get()], 'm:Ul')
    phiu = getattr(listado_barras[barras_cbx.get()], 'm:phiu')
    u = getattr(listado_barras[barras_cbx.get()], 'm:u')
    msg_barra='Resultado:\nNOMBRE: '+name+'\nUL:
'+str(round(ui,2))+ ' kV'+'\nPHIU: '+str(round(phiu,2))+ ' deg'+'\nU:
'+str(round(u,2))+ ' p.u.'
    lb_datos_barra_obtenido.config(text=msg_barra)

def ejecutar():
    ejecutar = app.GetFromStudyCase('ComLdf')
    ejecutar.Execute()

def fnCambiarBarra(event):
    cargarDatoBarra()
def fnCambiarRele(event):
    cargarDatoRele()

def ejectuar_simulacion():

    posicion_barra=listado_barras[barras_cbx.get()]
    print(posicion_barra)
    #name=getattr(posicion_barra,'loc_name')
    ejecutarComShc = app.GetFromStudyCase('ComShc')
    ejecutarComShc.iopt_mde = 1
    ejecutarComShc.iopt_allbus = 0
    ejecutarComShc.shcobj = posicion_barra
    ejecutarComShc.Execute()
    Skss=getattr(posicion_barra,'m:Skss')
    Ikss=getattr(posicion_barra,'m:Ikss')
    ip=getattr(posicion_barra,'m:ip')

```

```

        msg_corto='Resultado de corto :\nSkss: '+str(round(Skss,2))+
' MVA'+'\nIkss: '+str(round(Ikss,2))+ ' kA'+'\nip:
'+str(round(ip,2))+ ' kA'
        lb_datos_barra_obtenido_corto.config(text=msg_corto)
        cargarDatoBarra()

def obtener_datos():

    messagebox.showinfo(
        title="Actualizar",
        message="Datos actualizados"
    )

def aviso_rele():

    messagebox.showinfo(
        title="Guardado",
        message="Datos guardados"
    )

def graficas():
    print('graficas de ecuaciones')
    list_valores = {}
    Iip_min = getattr(list_caracteristicas['ANSI/IEEE I squared
T'], 'imin')
    print(Iip_min)
    Iip_max = getattr(list_caracteristicas['ANSI/IEEE I squared
T'], 'imax')
    print(Iip_max)
    TP=2
    T={}
    cont = Iip_max - Iip_min
    k=20
    columnas = 1

    for i in vector:
        Iip_min = getattr(list_caracteristicas['ANSI/IEEE I
squared T'], 'imin')
        print(Iip_min)
        Iip_max = getattr(list_caracteristicas['ANSI/IEEE I
squared T'], 'imax')
        print(Iip_max)

        T={}

```

```

print(vector[i])
if (vector[i] == 'ANSI/IEEE I squared T'):
    ec1.write(0,1,vector[i])
    valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=valores[0]
    a2=valores[1]
    filas = 1
    for x in range(k):
        T[x]=(TP*sum(a1+a2))/(Iip_min*Iip_min)
        a=(TP*sum(a1+a2))/(Iip_min*Iip_min)

        ec1.write_number(filas,1,a)
        ec1.write_number(filas,0,Iip_min)
        Iip_min=Iip_min+(cont/k)
        filas = filas +

```

1

```

if (vector[i] == 'ANSI/IEEE definite inverse'):
    ec1.write(0,2,vector[i])
    valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1
    for x in range(k):
        a = Iip_min**a2
        T[x]=(TP*(a1/(a-a3)+a4))
        b=(TP*(a1/(a-a3)+a4))
        ec1.write_number(filas,2,b)
        Iip_min=Iip_min+(cont/k)
        filas = filas + 1

```

```

if (vector[i] == 'ANSI/IEEE extremely inverse'):
    ec1.write(0,3,vector[i])
    valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1

```

```

for x in range(k):
    a = Iip_min**a2
    T[x]=(TP*(a1/(a-a3)+a4))
    b=(TP*(a1/(a-a3)+a4))
    ec1.write_number(filas,3,b)
    Iip_min=Iip_min+(cont/k)
    filas = filas + 1

if (vector[i] == 'ANSI/IEEE inverse'):
    ec1.write(0,4,vector[i])
    valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1
    for x in range(k):
        a = Iip_min**a2
        T[x]=(TP*(a1/(a-a3)+a4))
        b=(TP*(a1/(a-a3)+a4))
        ec1.write_number(filas,4,b)
        Iip_min=Iip_min+(cont/k)
        filas = filas + 1

if (vector[i] == 'ANSI/IEEE long inverse'):
    ec1.write(0,5,vector[i])
    valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1
    for x in range(k):
        a = Iip_min**a2
        T[x]=(TP*(a1/(a-a3)+a4))
        b=(TP*(a1/(a-a3)+a4))
        ec1.write_number(filas,5,b)
        Iip_min=Iip_min+(cont/k)
        filas = filas + 1

if (vector[i] == 'ANSI/IEEE moderately inverse'):
    ec1.write(0,6,vector[i])

```

```

        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1
    for x in range(k):
        a = Iip_min**a2
        T[x]=(TP*(a1/(a-a3)+a4))
        b=(TP*(a1/(a-a3)+a4))
        ec1.write_number(filas,6,b)
        Iip_min=Iip_min+(cont/k)
        filas = filas + 1

    if (vector[i] == 'ANSI/IEEE short inverse'):
        ec1.write(0,7,vector[i])
        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1
    for x in range(k):
        a = Iip_min**a2
        T[x]=(TP*(a1/(a-a3)+a4))
        b=(TP*(a1/(a-a3)+a4))
        ec1.write_number(filas,7,b)
        Iip_min=Iip_min+(cont/k)
        filas = filas + 1

    if (vector[i] == 'ANSI/IEEE very inverse'):
        ec1.write(0,8,vector[i])
        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
    a1=sum(valores[0])
    a2=sum(valores[1])
    a3=sum(valores[2])
    a4=sum(valores[3])
    filas = 1
    for x in range(k):
        a = Iip_min**a2
        T[x]=(TP*(a1/(a-a3)+a4))
        b=(TP*(a1/(a-a3)+a4))

```

```

        ec1.write_number(filas,8,b)
        Iip_min=Iip_min+(cont/k)
        filas = filas + 1

    if (vector[i] == 'IEC 255-3 extremely inverse'):
        ec1.write(0,9,vector[i])
        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
        a1=sum(valores[0])
        a2=sum(valores[1])
        a3=sum(valores[2])
        filas = 1
        for x in range(k):
            a = Iip_min**a2
            T[x]=(TP*a1/((a-a3)))
            b=(TP*a1/((a-a3)))
            ec1.write_number(filas,9,b)
            Iip_min=Iip_min+(cont/k)
            filas = filas + 1

    if (vector[i] == 'IEC 255-3 inverse'):
        ec1.write(0,10,vector[i])
        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
        a1=sum(valores[0])
        a2=sum(valores[1])
        a3=sum(valores[2])
        filas = 1
        for x in range(k):
            a = Iip_min**a2
            T[x]=(TP*a1/((a-a3)))
            b=(TP*a1/((a-a3)))
            ec1.write_number(filas,10,b)
            Iip_min=Iip_min+(cont/k)
            filas = filas + 1

    if (vector[i] == 'IEC 255-3 long inverse'):
        ec1.write(0,11,vector[i])
        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
        a1=sum(valores[0])
        a2=sum(valores[1])
        a3=sum(valores[2])
        filas = 1

```

```

        for x in range(k):
            a = Iip_min**a2
            T[x]=(TP*a1/((a-a3)))
            b=(TP*a1/((a-a3)))
            ec1.write_number(filas,11,b)
            Iip_min=Iip_min+(cont/k)
            filas = filas + 1

    if (vector[i] == 'IEC 255-3 very inverse'):
        ec1.write(0,12,vector[i])
        valores =
getattr(list_caracteristicas[vector[i]], 'vmat')
        a1=sum(valores[0])
        a2=sum(valores[1])
        a3=sum(valores[2])
        filas = 1
        for x in range(k):
            a = Iip_min**a2
            T[x]=(TP*a1/((a-a3)))
            b=(TP*a1/((a-a3)))
            ec1.write_number(filas,12,b)
            Iip_min=Iip_min+(cont/k)
            filas = filas + 1

    print(T)

chart1 = workbook.add_chart({'type': 'line'})
chart1.add_series({
    'name':      '=Sheet1!$B$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$B$2:$B$21',
})
chart1.add_series({
    'name':      '=Sheet1!$C$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$C$2:$C$21',
})

chart1.add_series({
    'name':      '=Sheet1!$D$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$D$2:$D$21',
})

chart1.add_series({

```

```

        'name':      '=Sheet1!$E$1',
        'categories': '=Sheet1!$A$2:$A$21',
        'values':    '=Sheet1!$E$2:$E$21',
    })

chart1.add_series({
    'name':      '=Sheet1!$F$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$F$2:$F$21',
})

chart1.add_series({
    'name':      '=Sheet1!$G$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$G$2:$G$21',
})

chart1.add_series({
    'name':      '=Sheet1!$H$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$H$2:$H$21',
})

chart1.add_series({
    'name':      '=Sheet1!$I$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$I$2:$I$21',
})

chart1.add_series({
    'name':      '=Sheet1!$J$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$J$2:$J$21',
})

chart1.add_series({
    'name':      '=Sheet1!$K$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$K$2:$K$21',
})

chart1.add_series({
    'name':      '=Sheet1!$L$1',
    'categories': '=Sheet1!$A$2:$A$21',
    'values':    '=Sheet1!$L$2:$L$21',
})

```



```

    })

    chart1.add_series({
        'name':      '=Sheet1!$M$1',
        'categories': '=Sheet1!$A$2:$A$21',
        'values':    '=Sheet1!$M$2:$M$21',
    })

    chart1.set_title({'name': 'Comportamiento Reles'})
    chart1.set_x_axis({'name': 'I/Ip'})
    chart1.set_y_axis({'log_base': 10})
    chart1.set_y_axis({'name': 'Tiempo (seg)'})
    chart1.set_style(10)
    ec2.insert_chart('B2', chart1, {'x_scale': 2, 'y_scale':
2})

```

```

workbook.close()

```

```

inicio.destroy()
ventana = tkinter.Tk()
ventana.title('Mi titulo')
ventana.geometry("1050x400")
ventana.config(bg="white")
etiqueta = Label(ventana, text="COORDINACIÓN DE PROTECCIONES",
bg="white", fg="black", borderwidth=1, relief="solid", padx=10,
pady=10).grid(column=2, row=0)
    btn_abrir = tkinter.Button(ventana, text ="Activar Proyecto",
command=activar_proyecto, bg="#5585f5", fg="white").grid(row=2,
columnspan=2)
    image1 = Image.open("utc.png")
    img = ImageTk.PhotoImage(image1)

    label1 = tkinter.Label(image=img, width=420, height=40)
    label1.image = img
    label1.place(x=650, y=1)

```

```

    Label(ventana, text="Ingrese el nombre del proyecto",
bg="white", fg="black", borderwidth=1, relief="solid", padx=2,
pady=2).grid(row = 1, column=0)
    caso = ttk.Entry(width=40)
    caso.grid(padx=5, pady=5, row=1, column=1)

    btn_abrir = tkinter.Button(ventana, text ="Activar Proyecto",
command=activar_proyecto, bg="#5585f5", fg="white").grid(row=2,
columnspan=2)

    #Label(ventana, text="Seleccionar un rele:", borderwidth=1,
relief="solid").grid(pady=5, row = 2, column=4)
    #combo = ttk.Combobox(ventana, state="readonly")
    #combo.bind("<<ComboboxSelected>>", aviso_rele)
    #combo.place(x=780, y=70)

    #reles
    Label(ventana, text="Relés encontrados", borderwidth=1,
relief="solid", bg="white", fg="black", width=23).grid(pady=5, row =
7, column=0)
    seleccionar_barra_rele = tk.StringVar()
    barras_rele_cbx = ttk.Combobox(ventana,
textvariable=seleccionar_barra_rele,height=5)
    barras_rele_cbx['state'] = 'readonly'
    barras_rele_cbx.place(x=180,y=100)
    barras_rele_cbx.bind('<<ComboboxSelected>>', fnCambiarRele)

    lb_datos_rele_obtenido= ttk.Label(ventana, text="Resultado:")
    lb_datos_rele_obtenido.place(x=10,y=130)

    #barras
    Label(ventana, text="Barras encontradas", borderwidth=1,
relief="solid", bg="white", fg="black", width=23).place(x=0, y=200)
    seleccionar_barra = tk.StringVar()
    barras_cbx = ttk.Combobox(ventana,
textvariable=seleccionar_barra,height=5)
    barras_cbx['state'] = 'readonly'
    barras_cbx.place(x=180,y=200)
    barras_cbx.bind('<<ComboboxSelected>>', fnCambiarBarra)

    lb_datos_barra_obtenido= ttk.Label(ventana, text="Resultado:")
    lb_datos_barra_obtenido.place(x=10,y=230)
    lb_datos_barra_obtenido_corto= ttk.Label(ventana,
text="Resultado de corto:")

```

```

lb_datos_barra_obtenido_corto.place(x=180,y=230)

frame = Frame(ventana, bg="#ffc133")
frame.config(width=480, height=150)
frame.place(x=450, y=100)
Label(ventana, text="Parámetros del Relé:", borderwidth=1,
relief="solid", fg="black").place(x=650, y=120)

Label(ventana, text="Corriente de Pick Up:", borderwidth=1,
relief="solid", fg="black", width=20).place(x=550, y=150)
entry_corriente = ttk.Entry(width=20)
entry_corriente.place(x=750, y=150)

Label(ventana, text="Dial del relé:", borderwidth=1,
relief="solid", fg="black", width=20).place(x=550, y=170)
entry_dial = ttk.Entry(width=20)
entry_dial.place(x=750, y=170)

btn_guardar_datos = tkinter.Button(ventana, text = "Guardar
datos", command=aviso_rele, width=20, bg="#5585f5",
fg="white").place(x=547, y=200)
actualizar = tkinter.Button(ventana, text = "Actualizar",
command=obtener_datos, width=20, bg="#5585f5",
fg="white").place(x=740, y=200)

btn_ejecutar_simulacion = tkinter.Button(ventana, text
="Ejecutar Simulación Corto", command=ejectuar_simulacion, width=40,
bg="#5585f5", fg="white", padx=8, pady=8).place(x=250, y=300)
btn_graficar = tkinter.Button(ventana, text = "Graficar", command
= graficas, width=40, bg="#5585f5", fg="white", padx=8,
pady=8).place(x=550, y=300)

Label(ventana, text="Autor: Henry Peralvo", borderwidth=1,
relief="solid", fg="black").place(x=100, y=350)
Label(ventana, text="Tutor: Xavier Maldonado", borderwidth=1,
relief="solid", fg="black").place(x=100, y=370)

```

```

inicio.title('Inicio del Programa')
inicio.geometry("650x150")
inicio.config(bg="white")

etiqueta = Label(inicio, text="COMUNICAR SUBLIME Y POWERFACTORY",
bg="white", fg="black", borderwidth=1, relief="solid", padx=10,
pady=10).grid(columnspan=4, row=0)

text_powerfactory = ttk.Entry(width=70)
text_powerfactory.grid(padx=5, pady=5, row=1, column=1)

dir_powerfactory = tkinter.Button(inicio, text = "Abrir
Powerfactory", command=powerfactory_dir, bg="#5585f5",
fg="white").grid(row=1, column=2)

text_python = ttk.Entry(width=70)
text_python.grid(padx=5, pady=5, row=2, column=1)
text_python.insert(0, " ")
dir_python = tkinter.Button(inicio, text = "Abrir Python",
command=python_dir, bg="#5585f5", fg="white").grid(row=2, column=2)

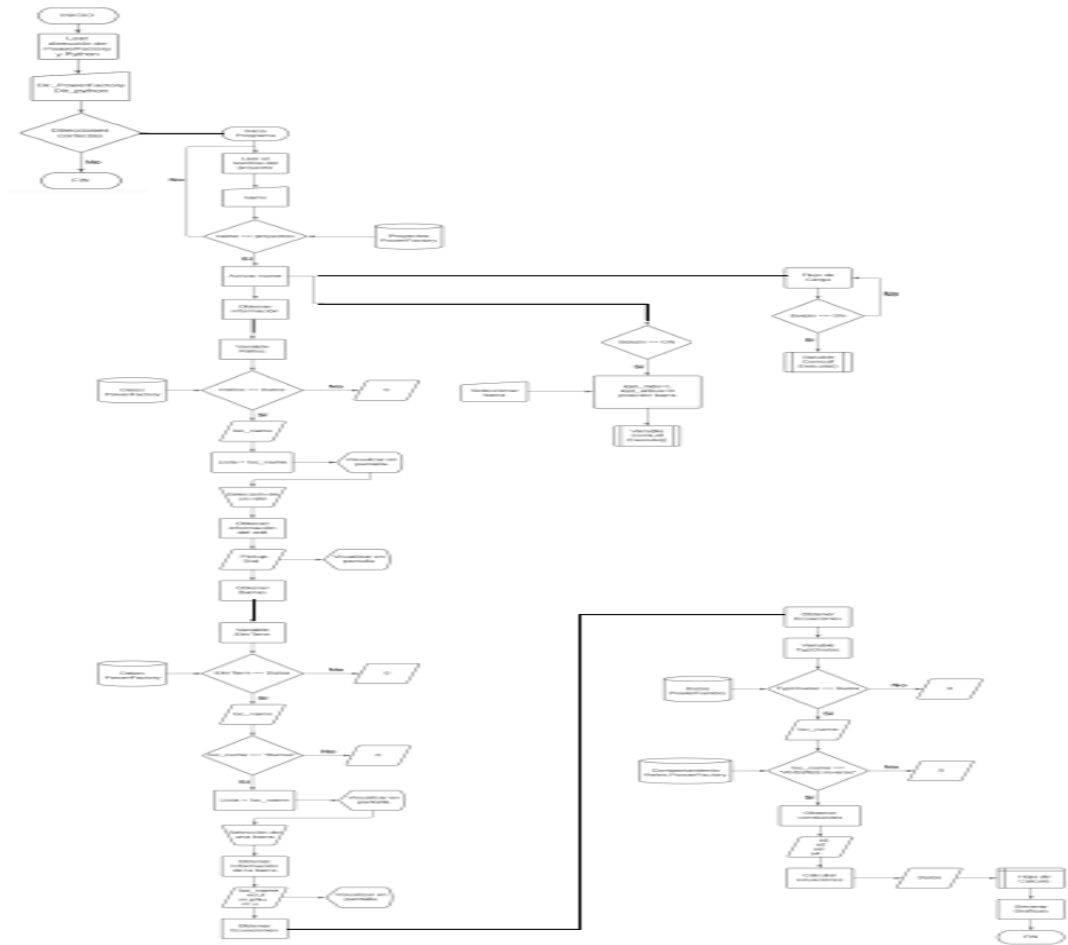
ventana_dos = tkinter.Button(inicio, text = "Iniciar Programa",
command=ventana_secundaria, bg="#f51414", fg="white").grid(row=3,
columnspan=3)
dir_default = tkinter.Button(inicio, text = "Dirección Default",
command=direccion_default, bg="#5585f5", fg="white").grid(row=3,
column=2)

"""
info_estudio = ttk.Entry(width=50)
info_estudio.place(x=200,y=80)

activar = tkinter.Button(ventana, text = "Activar Caso",
command=activar_caso)
activar.config(width=20, height=2)
activar.place(x=0,y=70)
"""
inicio.mainloop()

```

Anexo 6: Diagrama de flujo del proceso metodológico



Anexo 7: Análisis de Flujo de Carga en PowerFactory.

Flujo de Carga en PowerFactory																		
	Bus 01	Bus 02	Bus 03	Bus 04	Bus 05	Bus 06	Bus 07	Bus 08	Bus 09	Bus 10	Bus 11	Bus 12	Bus 13	Bus 14	Bus 15	Bus 16	Bus 17	Bus 18
m:UI (KV)	361,3	361,8	355,4	346,3	346,8	347,6	344	343,6	354,7	350,9	349,4	138	349,9	349	350,3	356	356,6	355,7
m:u (p.u.)	1,047	1,049	1,03	1,004	1,005	1,008	0,997	0,996	1,028	1,017	1,013	1	1,014	1,012	1,015	1,032	1,034	1,031
m:phiu (deg)	-8,4	-5,8	-8,6	-9,6	-8,6	-7,9	-10,1	-10,6	-10,3	-5,4	-6,3	-6,2	-6,1	-7,7	-7,7	-6,2	-7,3	-8,2

Flujo de Carga en PowerFactory																		
Bus 19	Bus 20	Bus 21	Bus 22	Bus 23	Bus 24	Bus 25	Bus 26	Bus 27	Bus 28	Bus 29	Bus 30	Bus 31	Bus 32	Bus 33	Bus 34	Bus 35	Bus 36	Bus 37
362,2	228	355	362,2	360,5	357,9	364,9	363	358	362,3	362,2	17,3	16,2	16,2	16,5	16,7	17,3	17,5	17
1,05	0,991	1,032	1,05	1,045	1,037	1,058	1,052	1,038	1,05	1,05	1,048	0,982	0,983	0,997	1,012	1,049	1,064	1,028
-1	-2	-3,8	0,7	0,5	-6,1	-4,4	-5,5	-7,5	-2	0,7	-3,3	0	2,6	4,2	3,2	5,6	8,3	2,4

Anexo 8: Análisis de Flujo de Carga en Python.

Flujo de Carga en Phyton																		
	Bus 01	Bus 02	Bus 03	Bus 04	Bus 05	Bus 06	Bus 07	Bus 08	Bus 09	Bus 10	Bus 11	Bus 12	Bus 13	Bus 14	Bus 15	Bus 16	Bus 17	Bus 18
m:UI (KV)	361,34	361,81	355,41	346,33	346,83	347,65	343,97	343,63	354,74	350,92	349,38	138,02	349,94	349,05	350,31	355,96	356,58	355,67
m:u (p.u.)	1,05	1,05	1,03	1	1,01	1,01	1	1	1,03	1,02	1,01	1	1,01	1,01	1,02	1,03	1,03	1,03
m:phiu (deg)	-8,44	-5,75	-8,6	-9,61	-8,61	-7,95	-10,12	-10,62	-10,32	-5,43	-6,28	-6,24	-6,1	-7,66	-7,74	-6,19	-7,3	-8,22

Flujo de Carga en Phyton																				
Bus 19	Bus 20	Bus 21	Bus 22	Bus 23	Bus 24	Bus 25	Bus 26	Bus 27	Bus 28	Bus 29	Bus 30	Bus 31	Bus 32	Bus 33	Bus 34	Bus 35	Bus 36	Bus 37	Bus 38	Bus 39
362,2	227,97	355,96	362,18	360,45	357,87	364,86	362,97	358,02	362,29	362,23	17,28	16,2	16,22	16,45	16,7	17,31	17,55	16,96	16,94	355,35
1,05	0,99	1,03	1,05	1,04	1,04	1,06	1,05	1,04	1,05	1,05	1,05	0,98	0,98	1	1,01	1,05	1,06	1,03	1,03	1,03
-1,02	-2,01	-3,78	0,67	0,47	-6,07	-4,36	-5,53	-7,5	-2,01	0,74	-3,33	0	2,57	4,19	3,17	5,63	8,32	2,42	7,81	-10,05

Anexo 9: Análisis de Corto Circuito en PowerFactory.

Corto Circuito en PowerFactory																		
	Bus 01	Bus 02	Bus 03	Bus 04	Bus 05	Bus 06	Bus 07	Bus 08	Bus 09	Bus 10	Bus 11	Bus 12	Bus 13	Bus 14	Bus 15	Bus 16	Bus 17	Bus 18
m:SKss (MVA)	6083,77	9906,82	8209,45	7785,38	8170,74	8519,75	6572,64	6804,66	5962,42	7965,91	7680,52	3226,15	7359,6	7548,33	7072,83	9981,84	8318,73	7111,19
m:Ikss (kA)	10,18	16,58	13,74	13,03	13,67	14,26	11	11,39	9,98	13,33	12,85	13,5	12,32	12,63	11,84	16,7	13,92	11,9
m:ip (kA)	26,63	41,54	35,24	33,99	35,91	37,54	28,76	29,79	26,41	34,97	33,62	35,9	32,08	32,78	30,31	43,34	35,85	30,53

Corto Circuito en PowerFactory																		
Bus 19	Bus 20	Bus 21	Bus 22	Bus 23	Bus 24	Bus 25	Bus 26	Bus 27	Bus 28	Bus 29	Bus 30	Bus 31	Bus 32	Bus 33	Bus 34	Bus 35	Bus 36	Bus 37
7942,41	5717,65	6698,36	7591,24	6892,42	7318,4	7988,31	5876,93	5569,64	3844,49	5092,04	8766,76	6455,65	7123,27	7421,33	5555,46	7337,48	6081,89	6490,59
13,29	14,35	11,21	12,7	11,53	12,25	13,37	9,83	9,32	6,43	8,52	306,76	225,89	249,25	259,68	194,39	256,75	212,81	227,11
34,61	37,61	29,35	33,64	30,46	31,95	30,58	24,67	23,7	16,35	21,98	806,21	603,82	662,44	681,09	511,08	682,34	565,2	578,08

Anexo 10: Análisis de Corto Circuito en Python.

Corto Circuito en Phytton																			
	Bus 01	Bus 02	Bus 03	Bus 04	Bus 05	Bus 06	Bus 07	Bus 08	Bus 09	Bus 10	Bus 11	Bus 12	Bus 13	Bus 14	Bus 15	Bus 16	Bus 17	Bus 18	Bus 19
m:SKss (MVA)	6205,4 454	10402, 161	8537, 828	7863,2 338	8497,5 696	8519, 75	6901, 272	6804, 66	6141,2 926	8125,2 282	7987,7 408	3258,4 115	735 9,6	7699,2 966	7426,4 715	9981, 84	8568,2 919	7466,7 495	8021,8 341
m:Ikss (kA)	10,689	17,077 4	13,74	13,681 5	13,67	14,40 26	11,22	11,50 39	10,079 8	13,729 9	12,978 5	13,5	12,3 2	13,008 9	12,076 8	16,7	14,059 2	12,019	13,555 8
m:ip (kA)	27,162 6	41,955 4	35,59 24	33,99	37,705 5	39,04 16	30,19 8	31,27 95	27,730 5	36,019 1	33,62	36,259	32,0 8	32,78	30,916 2	44,20 68	37,284	31,751 2	35,994 4

Corto Circuito en Phytton																			
Bus 20	Bus 21	Bus 22	Bus 23	Bus 24	Bus 25	Bus 26	Bus 27	Bus 28	Bus 29	Bus 30	Bus 31	Bus 32	Bus 33	Bus 34	Bus 35	Bus 36	Bus 37	Bus 38	Bus 39
5946, 356	6765,3 436	7894,8 896	7099,1 926	7611, 136	8148,0 762	6170,7 765	5848, 122	4036,7 145	5346, 642	9117,4 304	6649,3 195	7479,4 335	7569,7 566	5611,0 146	7337, 48	6203,5 278	6685,3 077	7234, 205	58938,0 276
14,78 05	11,546 3	13,208	12,106 5	12,74	13,771 1	10,026 6	9,599 6	6,4943	8,946	312,89 52	230,40 78	259,22	262,27 68	198,27 78	264,4 525	214,93 81	229,38 11	258,0 48	99,601
39,11 44	30,230 5	33,64	31,373 8	33,54 75	32,109	25,410 1	23,7	16,35	22,63 94	822,33 42	609,85 82	695,56 2	681,09	526,41 24	695,9 868	565,2	601,20 32	654,3 147	254,64