

GRADO EN INGENIERÍA EN TECNOLOGÍA INDUSTRIAL
TRABAJO FIN DE GRADO

***HERRAMIENTA DE CÁLCULO DE
TRANSFORMADORES DE POTENCIA.
CAÍDAS DE TENSIÓN, PÉRDIDAS Y
RENDIMIENTO.***

Alumno: Salvador, Vidal, Álvaro

Director: Valverde, Santiago, Víctor

Curso: 2018-2019

Fecha: En Bilbao, 27, Junio, 2019

DATOS BÁSICOS DEL TRABAJO FIN DE GRADO

- Alumno: Salvador Vidal, Álvaro.
- Director: Valverde Santiago, Víctor.
- Departamento: Ingeniería Eléctrica.

- Título del Trabajo: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.
- Resumen: El objetivo del presente trabajo es desarrollar una aplicación software con una interfaz intuitiva y de fácil manejo, que permita ejecutar los cálculos básicos de caída de tensión, pérdidas y rendimiento de un transformador de potencia en base a sus datos nominales y sus ensayos de vacío y cortocircuito. Para ello se utilizará el software comercial “Matlab” y su aplicación “GUIDE” para el desarrollo de interfaces gráficas de usuario (“GUI”).
- Palabras clave: interfaz gráfica, transformador, condiciones de carga, parámetros de diseño, caídas de tensión.

- Izenburua: Potentzia transformadore calculu erreminta. Tentsio jaitsierak, galerak eta errendimendua.
- Laburpena: Lan honen helburu nagusia tentsio jaitsiera, galerak eta potentzia transformadore baten errendimenduaren, bere hutseko eta zirkuitulaburreko eta bere datu nominalak kontuan izanda, interfaze intuitiboa duen eta erabilera errazeko software aplikazio bat garatzea da. Horretarako erabiltzailearen (“GUI”) interfaze grafikoen garapenerako software komertzial “Matlab” eta bere aplikazio “GUIDE” erabiliko dira.
- Hitzgakoak: interfaze grafikoa, transformadore, karga egoera, diseinu parametroak, tentsio jaitsierak.

- Title: Power transformer calculation tool. Voltage drops, losses and performance.
- Abstract: The main goal of this Project is to develop a software application with an intuitive and easy to use interface, wich allows to execute the basic calculations of voltaje drop, energy losses and performance of a power transformer based on its nominal data, vacuum and short-circuit tests. The software “Matlab” and its “GUIDE” application for the development of graphical user interfaces (“GUI”) will be used.
- Keywords: graphical interface, transformer, load conditions, design parameters, voltaje drop.

ÍNDICE

1. INTRODUCCIÓN.....	8
2. CONTEXTO.....	9
2.1. Sistema eléctrico de potencia.....	9
2.2. El transformador.....	11
2.3. Caídas de tensión, pérdidas y rendimiento del transformador.	12
3. OBJETIVOS Y ALCANCE.....	14
4. BENEFICIOS DEL PROYECTO.....	15
4.1. Beneficios técnicos.....	15
4.2. Beneficios sociales.....	16
4.3. Beneficios económicos.....	16
5. ESTADO DEL ARTE.....	18
5.1. El transformador monofásico.....	18
5.1.1. Principio de funcionamiento.....	19
5.2. Transformador de potencia monofásico real en vacío.....	20
5.3. Transformador de potencia monofásico real en carga.....	22
5.4. Circuito eléctrico equivalente del transformador.....	23
5.5. Circuito eléctrico equivalente simplificado del transformador.....	25
5.5.1. Conexión a carga inductiva.....	27
5.5.2. Conexión a carga capacitiva.....	27
5.6. Balance de potencias en un transformador.....	27
5.7. Ensayo de vacío.....	29
5.8. Ensayo de cortocircuito.....	29
5.9. Caída de tensión en un transformador. Coeficientes de variación de tensión.....	30
5.10. Rendimiento de un transformador.....	32
5.11. El transformador trifásico.....	33
5. 11. 1. Circuito equivalente monofásico.....	34
6. ANÁLISIS DE ALTERNATIVAS.....	36
6.1. Presentación de las alternativas.....	36
6.1.1. Pruebas de campo.....	36
6.1.2. Modelos físicos a escala.....	36
6.1.3. Simulación software.....	36
6.1.4. Criterios de selección.....	36
6.1.5. Conclusión del análisis.....	37

6.2. Elección de la herramienta software.....	39
6.2.1. ATP (<i>Alternative Transients Program</i>).....	39
6.2.2. PSCAD/EMTDC.....	39
6.2.3. Mathcad.....	40
6.2.4. MATLAB-GUIDE.....	40
6.2.5. Criterios de selección.....	40
6.2.6. Resultados del análisis.....	40
7. DESCRIPCIÓN DE LA SOLUCIÓN.....	42
7.1. Introducción a la plataforma GUIDE.....	42
7.2. Interfaz gráfica de usuario.....	43
7.2.1. Módulo 1: Datos del transformador.....	45
7.2.2. Módulo 2: Ensayos de vacío y cortocircuito.....	46
7.2.3. Modulo 3: selección del tipo de problema.....	48
7.2.4. Módulos 4 y 5: condiciones de carga del transformador.....	50
7.2.5. Presentación de resultados.....	52
7.2.6. Mensajes de error.....	55
8. EJEMPLOS DE APLICACIÓN.....	57
8.1. Ejemplo 1.....	57
8.2. Ejemplo 2.....	64
9. PLANIFICACIÓN Y DESCRIPCIÓN DE TAREAS.....	69
10. PRESUPUESTO.....	71
11. CONCLUSIONES.....	72
12. REFERENCIAS.....	73
13. ANEXO. CÓDIGO DE LA INTERFAZ.....	74

ÍNDICE DE FIGURAS

Figura 1. Esquema del SEP.....	9
Figura 2. Esquema del SEP y niveles de tensión [1].	10
Figura 3. Transformador de potencia.	12
Figura 4. Partes activas de un transformador de potencia [1].	18
Figura 5. Transformador real en vacío [1].	20
Figura 6. Representación de bobinas reales [1].	21
Figura 7. Transformador real en carga [1].	22
Figura 8. Representación del transformador real en carga [1].	24
Figura 9. Deducción del circuito equivalente [1].	25
Figura 10. Diagrama vectorial del transformador real en carga [1].	25
Figura 11. Circuito equivalente simplificado [1].	26
Figura 12. Circuito equivalente simplificado práctico [1].	26
Figura 13. Diagrama vectorial simplificado [1].	27
Figura 14. Carga capacitiva: efecto Ferranti [1].	27
Figura 15. Balance de potencias [1].	28
Figura 16. Ensayo de vacío nominal [1].	29
Figura 17. Ensayo de cortocircuito nominal [1].	30
Figura 18. Diagrama vectorial del circuito equivalente simplificado [1].....	31
Figura 19. Rendimiento [1].	33
Figura 20. Transformador de núcleo trifásico [1].	34
Figura 21. Circuito equivalente monofásico [1].	35
Figura 22. Interfaz editable GUIDE.	42
Figura 23. Interfaz gráfica de usuario.	43
Figura 24. Diagrama de flujo.	44
Figura 25. Módulo 1.	45
Figura 26. Módulo 2.	47
Figura 27. Módulo 3.	48
Figura 28. Panel habilitado.	49
Figura 29. Panel inhabilitado.	49
Figura 30. Módulo problema tipo I.	51
Figura 31. Módulo problema tipo II.	52
Figura 32. Panel de resultados.	52
Figura 33. Panel de <i>Push Button</i>	52

Figura 34. Error de tipo numérico.	55
Figura 35. Error en el factor de potencia.	56
Figura 36. Datos del transformador I.	58
Figura 37. Datos de los ensayos I.	58
Figura 38. Tipo de problema y condiciones de carga.	59
Figura 39. Resultados I.	59
Figura 40. Vista general I.	61
Figura 41. Vista general II.	62
Figura 42. Vista general III.	63
Figura 43. Datos del transformador II.	65
Figura 44. Datos de los ensayos II.	66
Figura 45. Seleccionar el tipo de problema.	66
Figura 46. Condiciones de carga y resultados.	67
Figura 47. Vista general IV.	68
Figura 48. Diagrama Gantt.	70

ÍNDICE DE TABLAS

Tabla 1. Conclusión del análisis.....	38
Tabla 2. Resultados del análisis.....	41
Tabla 3. Tareas que componen el proyecto.....	70
Tabla 4. Horas internas.....	71
Tabla 5. Amortizaciones.....	71
Tabla 6. Gastos.	71
Tabla 7. Resumen.	71

1. INTRODUCCIÓN.

Este documento contiene el desarrollo del Trabajo Fin de Grado (TFG) “Herramientas de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento”. En primer lugar y para poder resolver de forma adecuada los conceptos a tratar en el desarrollo del TFG, previamente se van a definir unos contextos de trabajo, el alcance y objetivo de dicho trabajo y los beneficios de sus resultados.

A continuación, en el apartado del estado del arte se estudiarán las características de la maquina en cuestión y sus características de interés para el futuro desarrollo de la herramienta software. Se analizará el uso del transformador en el sistema eléctrico de potencia, sus modos de funcionamiento. Finalmente se detallará la formulación matemática necesaria para poder calcular sus caídas de tensión, rendimiento, y pérdidas.

Posteriormente se muestran las diferentes alternativas posibles sobre distintas interfaces matemáticas existentes que están disponibles para desarrollar esta herramienta de calculo. Se exponen sus ventajas y desventajas, y, además, se selecciona el más adecuado para llevar a cabo este trabajo en concreto.

El objetivo es programar mediante la interfaz matemática una herramienta software de cálculo de las caídas de tensión, rendimiento y pérdidas en un transformador. En la descripción de la solución se explica paso a paso como se ha desarrollado la interfaz, explicando y analizando las diferentes partes que la componen. Después se mostrarán los resultados obtenidos, enseñando varios ejemplos que facilitan el uso de el programa y la interpretación de los datos obtenidos.

Se ha realizado un presupuesto que recoge la inversión necesaria a realizar para llevar a cabo el proyecto y un diagrama Gantt, en el que se muestra el orden en el que se han llevado a cabo las tareas del proyecto.

Para finalizar, se analizan los resultados obtenidos y se redactan a modo de conclusiones al final de este documento. Estas conclusiones tratan sobre la interpretación que se puede dar a los datos que se obtienen mediante la interfaz desarrollada.

2. CONTEXTO.

Este Trabajo Fin de Grado ha sido realizado en la Escuela de Ingenieros de Bilbao (EIB) en el departamento de Ingeniería Eléctrica con el título “Herramientas de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento”.

El trabajo consiste en el desarrollo de una interfaz software para el cálculo de caídas de tensión, rendimiento y pérdidas en transformadores de potencia. Para una correcta comprensión del trabajo desarrollado, en este apartado contextualizaremos la máquina y sus parámetros a estudiar.

2.1. Sistema eléctrico de potencia.

El Sistema Eléctrico de Potencia (SEP) es el conjunto de instalaciones y equipos que generan y suministran energía eléctrica a gran escala. Sus partes fundamentales son: la generación, el transporte, la distribución y el consumo. Constituyen un sistema trifásico a frecuencia constante (50 Hz en Europa y 60 Hz en América) aunque cada una de sus partes posee su propio nivel de tensión [1].

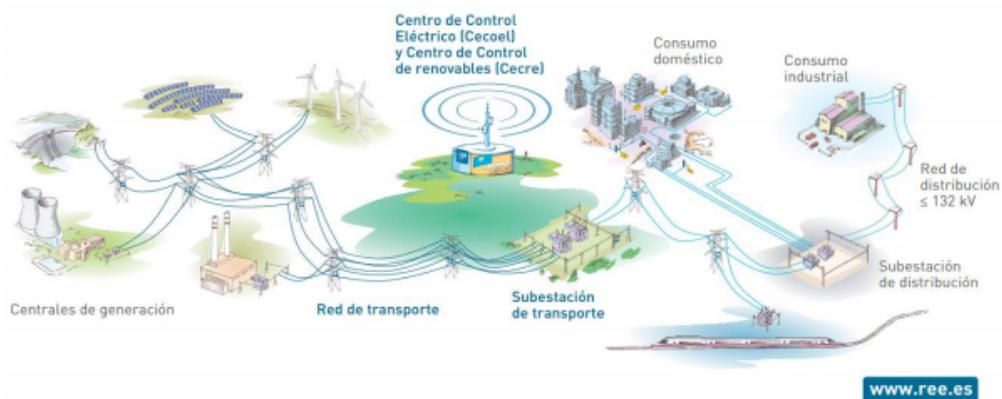


Figura 1. Esquema del SEP.

La generación tiene su origen en las centrales eléctricas a partir de otras formas de energía existentes [1].

Según la función de la central dentro del sistema, las centrales se pueden clasificar en: centrales de base, centrales de punta, centrales de reserva (rodante, rápida y lenta) y centrales de acumulación o de bombeo. También se pueden clasificar en función del tipo de fuente primaria que estas utilizan para producir la energía eléctrica, se clasifican entonces en: térmicas, hidroeléctricas, eólicas, solares,

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

nucleares, etc. Siendo las hidroeléctricas, las térmicas y las nucleares las más empleadas y las que se sitúan en los grandes centros de transformación.

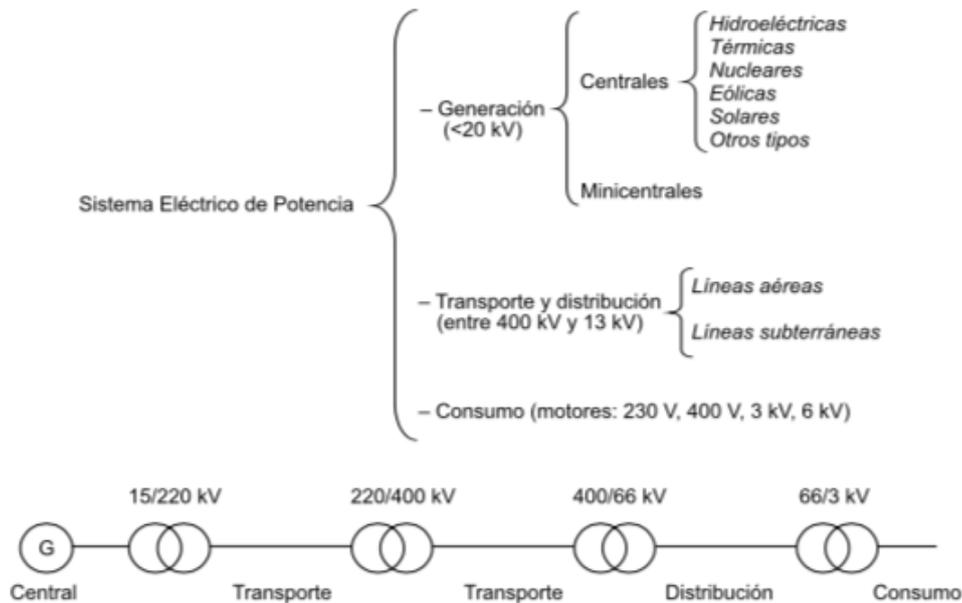


Figura 2. Esquema del SEP y niveles de tensión [1].

La energía eléctrica generada debe suministrarse a los consumidores a través de redes de distribución y transporte.

Las redes de transporte son las redes que comunican los centros de generación con las subestaciones principales o núcleos de carga. Estas líneas transmiten la energía eléctrica a unos niveles de tensión iguales o superiores a 220kV (220kV y 400kV en España) [1].

La red de distribución es la red de reparto de la energía eléctrica que abastece a los consumidores. Esta distribución se realiza a niveles de tensión inferiores utilizando líneas aéreas y subterráneas, siendo estas últimas las utilizadas en los núcleos urbanos.

En la actualidad, debido al impulso creciente de las energías renovables y a factores de búsqueda de mejora de la eficiencia técnica existen centros de generación que vuelcan su energía eléctrica directamente a la Red de Distribución [2].

Por último, en las zonas de consumo se recibe y emplea la energía en función de las características de cada centro. El consumo se realiza a diferentes niveles de tensión dependiendo de las potencias demandadas y los niveles de tensión de la zona de este modo, grandes complejos industriales con consumos del orden de 30 MVA o superiores están conectados directamente a 132 kV, mientras que consumidores del orden de 5 MVA estarán conectados a 30 kV y otros consumidores del orden de centenares de kVA se alimentarán en baja tensión (230/400 V) al igual que los consumidores domésticos [1].

Por tanto, para que el SEP funcione de una forma adecuada, será necesario convertir la energía para que sea transportada y reconvertirla para que sea consumida a niveles de tensión óptimos para cada uso. Para realizar estas tareas, se emplean diferentes máquinas eléctricas de corriente alterna.

2.2. El transformador.

Es la máquina eléctrica estática más común. Esta máquina se utiliza para convertir la energía eléctrica a distintos niveles de tensión. Está compuesta por dos circuitos eléctricos unidos entre sí por un circuito magnético. El circuito eléctrico que recibe la potencia es el denominado circuito primario. El que transmite la potencia a la salida es el circuito eléctrico secundario. El núcleo magnético es el encargado de unir ambos circuitos eléctricos y generar el circuito magnético. Sobre este núcleo magnético se arrollan los bobinados de los circuitos eléctricos.

La necesidad de emplear esta máquina se debe a la imposibilidad de trabajar en un mismo nivel de tensión a lo largo de todo el SEP, es decir, es necesario convertir la energía a diferentes niveles de tensión para un aprovechamiento óptimo y para lograr una mayor facilidad a la hora de realizar distintas maniobras sobre el sistema. Será necesario entonces operar a distintos niveles de tensión por razones de seguridad, protegiendo dispositivos, personas e infraestructuras.

Para ello distinguimos dos tipos de transformadores:

-Transformadores de potencia: transmiten la potencia eléctrica a diferentes niveles de tensión, variando la intensidad pero manteniendo la frecuencia.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

-Transformadores especiales: entre ellos están los transformadores de medida, protección y autotransformadores.

Por tanto, la gran ventaja radica en la capacidad de estas máquinas a la hora de variar la intensidad y la tensión de la energía eléctrica para adecuarlas a cada uso (altas tensiones y bajas intensidades en redes transporte y bajas tensiones e intensidades considerables en redes de consumo); logrando un rendimiento superior al de otras formas de conversión de la energía existentes [3].



Figura 3. Transformador de potencia.

2.3. Caídas de tensión, pérdidas y rendimiento del transformador.

Denominamos caída de tensión a la diferencia entre la tensión en vacío y la tensión en bornes de la carga, y se expresa en parámetros de forma porcentual. Cuando alimentamos un transformador de potencia a cierta tensión por su devanado primario, debido a las resistencias internas de los bobinados de los circuitos eléctricos y al carácter de las cargas conectadas, la tensión en bornes de secundario no será la misma que si se alimenta al transformador a esa misma tensión pero en vacío, es decir, sin cargas conectadas [4].

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Como ya se ha comentado, la caída de tensión la provocan distintos factores: por un lado las resistencias internas de los conductores de los devanados primario y secundario de los circuitos eléctricos que se arrollan sobre el circuito magnético; por otro lado, el carácter de la carga (inductivo, resistivo o capacitivo) también influye en la caída de tensión, tanto aumentándola como disminuyéndola.

Estos factores tienen su influencia sobre el rendimiento del transformador. El transformador no siempre podrá trabajar a su máximo rendimiento debido a las cargas que tenga acopladas en un determinado instante.

3. OBJETIVOS Y ALCANCE.

El principal objetivo de este Trabajo Fin de Grado es el desarrollo de una interfaz gráfica mediante la aplicación "GUIDE" que ofrece el programa "MATLAB" para calcular las caídas de tensión, pérdidas y rendimiento de transformadores monofásicos y trifásicos.

A su vez, el desarrollo de este trabajo conlleva mas objetivos a parte del que ha sido mencionado:

- Analizar y conocer los parámetros internos de un transformador y calcular sus caídas de tensión, sus pérdidas de potencia y su rendimiento.
- Estudiar la respuesta de la máquina ante diferentes situaciones de carga y alimentación, tanto en transformadores monofásicos como trifásicos.
- Programar las funciones necesarias para realizar los cálculos anteriormente mencionados en lenguaje de "MATLAB" para el correcto funcionamiento de la interfaz gráfica programada en "GUIDE" y la correcta obtención de resultados.
- Reducir los tiempos a la hora de realizar los cálculos de caída de tensión, pérdidas y rendimiento en transformadores monofásicos y trifásicos y así optimizar tanto recursos materiales como humanos sin necesidad de manipular otros dispositivos de medida sobre el transformador.

En conclusión, los objetivos fundamentales son por un lado el estudio de los transformadores de potencia para así poder llevar a cabo su programación en "MATLAB". Y por otro lado el manejo y la programación de la interfaz gráfica "GUIDE" de la que dispone "MATLAB".

4. BENEFICIOS DEL PROYECTO.

De los diferentes beneficios que se van a obtener mediante el desarrollo y la implantación de este Trabajo Fin de Grado se puede hacer una clasificación en beneficios técnicos, sociales y económicos. Debido al carácter de este proyecto, los beneficios fundamentales y mas reseñables son los beneficios técnicos. Dichos beneficios técnicos repercutirán en beneficios sociales y económicos.

4.1. Beneficios técnicos.

Como ya se ha mencionado en apartados anteriores, un “Sistema de Energía Eléctrica” es el conjunto de todas las instalaciones electrotécnicas, redes incluidas, y todas las instalaciones adicionales necesarias para la generación, transporte y utilización de la energía eléctrica dentro de una determinada unidad territorial. Para su correcto funcionamiento es necesario que el sistema sea flexible y proporcione fiabilidad y continuidad en el servicio.

El transformador, es una herramienta necesaria para poder satisfacer de forma óptima todas las necesidades del sistema, es una máquina eléctrica estática fundamental en cualquier red eléctrica. Por tanto, será necesario disponer de herramientas que permitan conocer su funcionamiento de una forma sencilla y rápida.

Conocer los parámetros de funcionamiento de nuestra máquina o conocer los parámetros que deseamos obtener al diseñar una de estas máquinas es el objetivo deseamos obtener mediante el desarrollo de la interfaz gráfica. Esta interfaz proporciona datos sobre los parámetros internos del transformador a estudiar o diseñar y cual será su respuesta ante diferentes formas de funcionamiento, todo ello sin necesidad de realizar mediciones sobre la máquina.

Los parámetros que se van a obtener son los de caída de tensión, pérdidas de energía interna, y rendimiento. Será necesario conocer estos datos para garantizar que la máquina está trabajando de forma adecuada y asegurar un correcto funcionamiento del resto de dispositivos conectados a ella y no poner en compromiso todo el sistema. Localizar unos niveles de tensión inadecuados o un exceso de pérdidas y un bajo rendimiento a tiempo otorga un mayor margen de maniobra a la hora de reparar o proteger toda la maquinaria eléctrica. Todo esto se puede lograr,

como ya se ha comentado, haciendo uso de la interfaz gráfica que se ha programado en este proyecto.

4.2. Beneficios sociales.

Como ya se ha dicho, todos los dispositivos del sistema eléctrico están diseñados para trabajar a ciertos niveles de tensión, y el sistema eléctrico tiene la obligación de mantener un servicio de calidad, fiable y asegurar su continuidad. El incumplimiento de estos parámetros no solo compromete el correcto funcionamiento de los dispositivos si no que también puede generar averías.

La capacidad de detectar las caídas de tensión, las pérdidas de potencia y el rendimiento de la máquina es una ventaja tanto para los diseñadores de estas máquinas como para los diseñadores de la red. Conocer las capacidades del transformador permite actuar a tiempo para poder resolver problemas de diseño o para realizar una mejor selección de la máquina que se va a utilizar.

La capacidad de detectar las caídas de tensión, las pérdidas de potencia y el rendimiento del transformador mediante la interfaz gráfica desarrollada en este proyecto permite realizar una toma de decisiones a tiempo para seleccionar el transformador adecuado para cada uso, asegurando una distribución energética de calidad a lo largo de todo el sistema, garantizando continuidad en el suministro y seguridad.

4.3. Beneficios económicos.

El hecho de poder calcular la caídas de tensión, pérdidas y rendimiento del transformador de potencia antes de ser instalado, de una forma rápida y sencilla supone un ahorro económico frente a la solución de realizar ensayos y toma de medidas. Todo esto repercute en un ahorro de recursos humanos, materiales y económicos.

Las labores de sustitución debido a que la máquina no cumpla los requisitos solicitados, o las labores de recambio de los dispositivos dañados por un suministro inadecuado, no solo ponen en compromiso al sistema, además, suponen un gasto elevado de recursos económicos.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Por todo esto, es una gran ventaja disponer de esta aplicación, ya que nos permite conocer al detalle la maquinaria con la que se va a trabajar y nos permite actuar con antelación a todos los problemas que podrían surgir de la elección de instalar una maquinaria inadecuada. Disponer de esta interfaz, implica un ahorro de tiempo, recursos humanos, y sobre todo económicos.

5. ESTADO DEL ARTE.

Para cumplir los objetivos y especificaciones de desarrollar una interfaz gráfica de uso sencillo y fiable, que calcula las caídas de tensión, pérdidas de potencia y rendimiento en transformadores tanto monofásicos como trifásicos, es necesario comprender el funcionamiento de estas máquinas.

A continuación se analizarán detalladamente los fundamentos teóricos bajo los que funciona un transformador y cómo se pueden calcular todos sus parámetros de interés para trabajar de forma óptima con él.

5.1. El transformador monofásico.

Esta máquina eléctrica está formada por un núcleo magnético y dos devanados eléctricos. El núcleo magnético realiza un circuito magnético, y los devanados eléctricos son circuitos eléctricos que inducen o son inducidos por campos magnéticos. El lado que recibe la energía se denomina devanado primario, y el lado que alimenta las cargas se denomina devanado secundario.

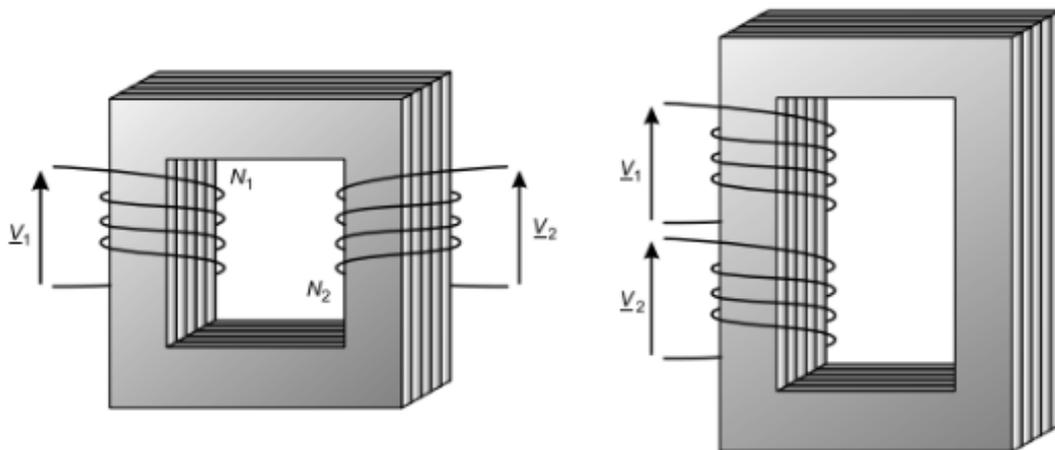


Figura 4. Partes activas de un transformador de potencia [1].

Se distinguen dos “tipos” de transformadores; los elevadores y los reductores. Los primeros son aquellos en los cuales la tensión de entrada es menos que la de salida, es decir, el devanado primario corresponde al devanado de baja tensión (BT) y el devanado secundario al de alta tensión (AT). Los segundos, se alimentan por el lado de AT y transmiten la potencia a un nivel de tensión menor que el de la entrada por el devanado secundario, siendo este en este caso el correspondiente a BT.

Además de los elementos estructurales, los elementos que constituyen un transformador son los siguientes:

-Partes activas: las partes más importantes del transformador. Son la agrupación del circuito magnético y los circuitos eléctricos, es decir, las partes anteriormente explicadas.

-Sistema de entrada/salida: son las partes del transformador que permiten conectarlo al circuito eléctrico de entrada y de salida. Tiene bornes de BT y AT, los primeros menos aislados pero con un cableado de mayor sección y los segundos con un mayor sistema de aislamiento pero cableado de menor sección.

-Sistema de refrigeración: durante el funcionamiento de esta máquina se producen ciertas pérdidas que se traducen en generación de calor, este calor será necesario evacuarlo para no provocar averías en la propia máquina ni poner en riesgo las estaciones de transformación o a los posibles operarios. La refrigeración se realiza introduciendo las partes activas en un baño de refrigerante (normalmente aceite mineral) y empleando un tanque o cuba. Se añadirá un sistema de aletas y ventiladores a la carcasa de la máquina cuando sea necesario para ayudar a la correcta refrigeración de esta.

-Sistemas de protección y de control: se emplean protecciones convencionales de tipo eléctrico como por ejemplo protecciones diferenciales y puestas a tierra. También es muy común proteger los sistemas de refrigeración mediante depósitos de expansión para el refrigerante y filtros de aire. El transformador posee un sistema de control que se utiliza para realizar un seguimiento de su estado de funcionamiento midiendo cada uno de sus parámetros de interés (temperatura, nivel de refrigerante, regulador de tensión, etc.).

5.1.1. Principio de funcionamiento.

Cuando una corriente alterna circula por el devanado de el transformador, se genera un flujo alterno que comienza a circular por el núcleo magnético, este flujo dependerá de la tensión aplicada, la frecuencia de la corriente alterna y del número de espiras del devanado. El flujo mutuo se vinculará con el devanado secundario e inducirá un voltaje que dependerá del número de espiras arrolladas en el secundario y

de la magnitud del flujo mutuo [5]. La ecuación fundamental que gobierna el funcionamiento del transformador es la siguiente:

$$E = 4.44 \cdot N \cdot \Phi \cdot f$$

Siendo “E” la tensión en bornes del devanado, “N” el número de espiras del devanado, “ Φ ” el flujo que atraviesa el núcleo magnético y “f” la frecuencia de la corriente alterna aplicada al devanado.

Como se puede comprobar, la tensión depende del número de espiras, por tanto, variando este parámetro a la hora de diseñar la máquina se pueden obtener distintos resultados de conversión de tensiones e intensidades.

Mas adelante se explicarán distintos modos de funcionamiento del transformador, esto hará que la notación de la ecuación fundamental varíe ya que las condiciones serán diferentes, no obstante, el principio de funcionamiento es el mismo para todos los transformadores de potencia.

5.2. Transformador de potencia monofásico real en vacío.

Un transformador se dice que trabaja en vacío cuando, estando alimentado por su primario, no posee ninguna carga conectada en el secundario [1].

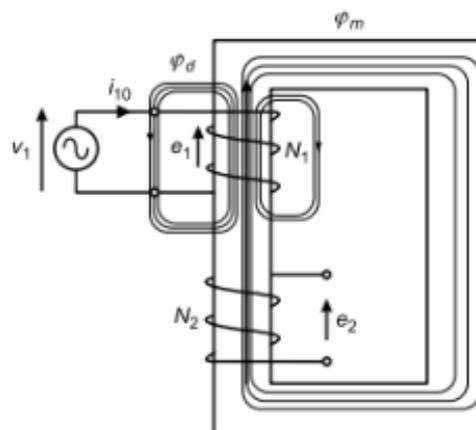


Figura 5. Transformador real en vacío [1].

En la figura que se acaba de mostrar, V_1 representa la tensión de alimentación, e_1 representa la tensión en bornes de el devanado primario, i_{10} la intensidad de corriente alterna que circula por el devanado primario en esta situación de vacío, N_1 el

número de espiras en el devanado primario, φ_d el flujo de fugas y φ_m el flujo magnético que atraviesa al núcleo magnético del transformador.

N_2 y e_2 representan el bobinado secundario y la tensión en bornes dicho bobinado respectivamente. Puede observarse que al estar funcionando en vacío no circula intensidad por este devanado, y por tanto no se produce flujo de fugas alguno.

En este modo de funcionamiento se cumple que:

$$V_1 \approx E_1 = 4.44 \cdot N_1 \cdot \Phi \cdot f$$

$$E_2 = 4.44 \cdot N_2 \cdot \Phi \cdot f$$

$$\frac{E_1}{E_2} = \frac{N_1}{N_2} = a$$

El parámetro “ a ”, es el parámetro al que denominaremos como relación de transformación, que indica la relación entre el número de espiras de los bobinados o la relación entre las tensiones de entrada y salida del transformador.

Con el fin de comprender mejor las pérdidas que se producen en una bobina, se adjunta la siguiente imagen. Una bobina real puede ser representada como una bobina ideal mas una impedancia (“ Z_1 ”) en serie que realiza la influencia del flujo de dispersión (“ X_1 ”) y las pérdidas por el efecto Joule (“ R_1 ”).

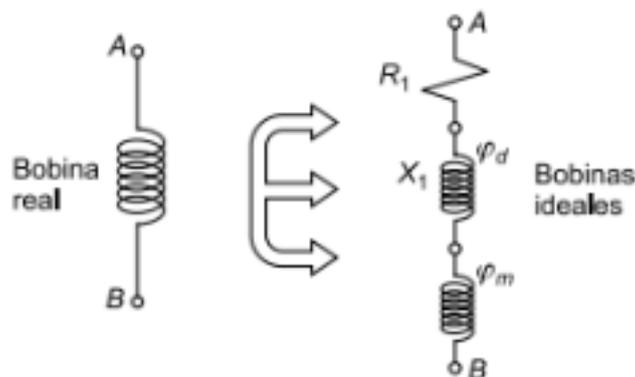


Figura 6. Representación de bobinas reales [1].

5.3. Transformador de potencia monofásico real en carga.

Se dice que un transformador está en carga cuando, estando alimentado por su primario, posee una carga conectada en el secundario. En estas condiciones existe circulación de intensidades tanto en el primario como en el secundario y, en consecuencia, flujos de dispersión o fugas en ambos bobinados [1]. Se tienen en cuenta el efecto de histéresis y saturación en el núcleo, siendo la conductividad del mismo distinta de cero apareciendo pérdidas magnéticas. También se tiene en cuenta la resistencia de los devanados que producen pérdidas [6].

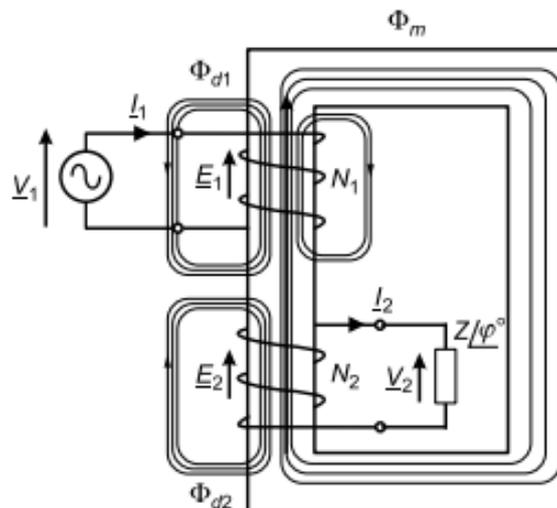


Figura 7. Transformador real en carga [1].

En la figura anterior se observa un transformador real que trabaja alimentando una carga, a continuación se detallan cuáles son los parámetros de dicha figura.

V_1 representa la tensión de alimentación, E_1 representa la tensión en bornes de el devanado primario, I_1 la intensidad de corriente alterna que circula por el devanado primario en esta situación de vacío, N_1 el número de espiras en el devanado primario, Φ_{d1} el flujo de fugas y Φ_m el flujo magnético que atraviesa al núcleo magnético del transformador.

V_2 , E_2 , I_2 , N_2 y Φ_{d2} son los mismos parámetros pero en este caso pertenecen al circuito secundario. Z_c representa a la carga conectada en bornes del circuito secundario.

Debido a las resistencias internas de los bobinados de los circuitos, las tensiones aplicadas al transformador y recibidas por la carga (V) difieren de la tensión

inducida en bornes de las espiras (E). Al estar en esta situación el secundario conectado a una carga, comienza a circular una intensidad de corriente alterna (I_2), esta será la causante de que también exista un flujo de fugas en el bobinado secundario (Φ_{d2}).

Para este caso se sigue cumpliendo lo siguiente:

$$V_1 \approx E_1 = 4.44 \cdot N_1 \cdot \Phi \cdot f$$

$$V_2 \approx E_2 = 4.44 \cdot N_2 \cdot \Phi \cdot f$$

$$\frac{V_1}{V_2} \approx \frac{E_1}{E_2} = \frac{N_1}{N_2} = a \quad (1)$$

Se puede observar que bajo esta forma de funcionamiento y en carga no se cumple la relación de transformación. Aparecen las denominadas caídas de tensión que se estudiarán en profundidad más adelante. Si el transformador fuese ideal (circuito magnético no saturado, no hay pérdidas en el hierro, ni los bobinados de resistencia y no hay flujo de dispersión) y estuviese en vacío se cumpliría que la división de E_1 entre E_2 sería exactamente igual a V_1 entre V_2 que a su vez esta división sería "a" (la relación de transformación).

5.4. Circuito eléctrico equivalente del transformador.

Con el fin de obtener un circuito equivalente válido del transformador de potencia monofásico hay que tener en cuenta las siguientes consideraciones mencionadas en la referencia [7]:

-Pérdidas en el cobre: debidas al calentamiento resistivo de los bobinados. Se representan mediante resistencias en el circuito equivalente (R_1 y R_2).

-Pérdidas por corrientes parásitas e histéresis: se conocen como pérdidas en el hierro. Las pérdidas por corrientes parásitas al calentamiento resistivo del núcleo magnético y las pérdidas por histéresis están relacionadas con el reordenamiento de los campos magnéticos. Se representan por una rama recorrida por una intensidad I_{Fe} que atraviesa una conductancia G_0 .

-Flujo de dispersión: son los flujos magnéticos anteriormente mencionados como Φ_{d1} y Φ_{d2} , que aparecen cuando circula intensidad por los bobinados primario y secundario produciendo autoinductancias que provocan pérdidas a tener en cuenta.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Se representarán en el circuito mediante dos bobinas, X_1 y X_2 , para el bobinado primario y secundario respectivamente.

Será necesario también representar los efectos que produce el flujo que atraviesa el núcleo magnético Φ_m . Este flujo crea una intensidad magnetizante I_m que retrasa 90° el voltaje aplicado. Se modela mediante la susceptancia B_0 conectada en paralelo a la fuente de tensión primaria y a la rama por la que circula la I_{Fe} . [7]

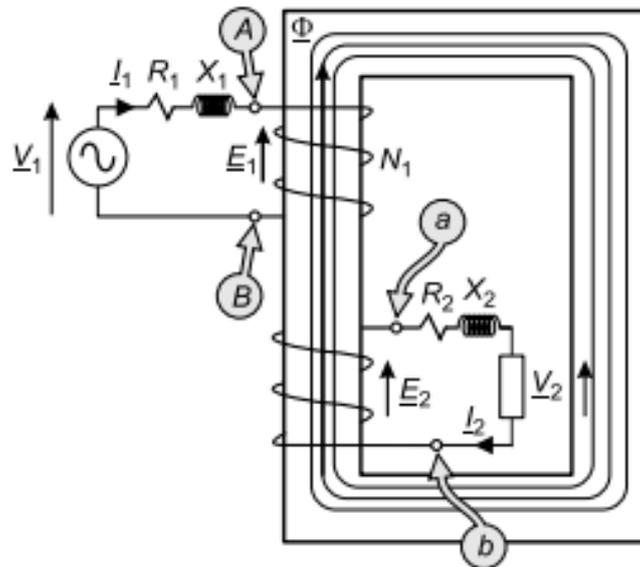


Figura 8. Representación del transformador real en carga [1].

Para poder representar las corrientes y las impedancias será necesario unir las parejas de puntos A y a de la figura y la B y b. Una vez realizada esta operación, es necesario hallar una equivalencia eléctrica; para conseguirla se refieren los valores del secundario al primario utilizando la relación de transformación consiguiendo que $E_1 = E'_2$.

El circuito equivalente es el siguiente:

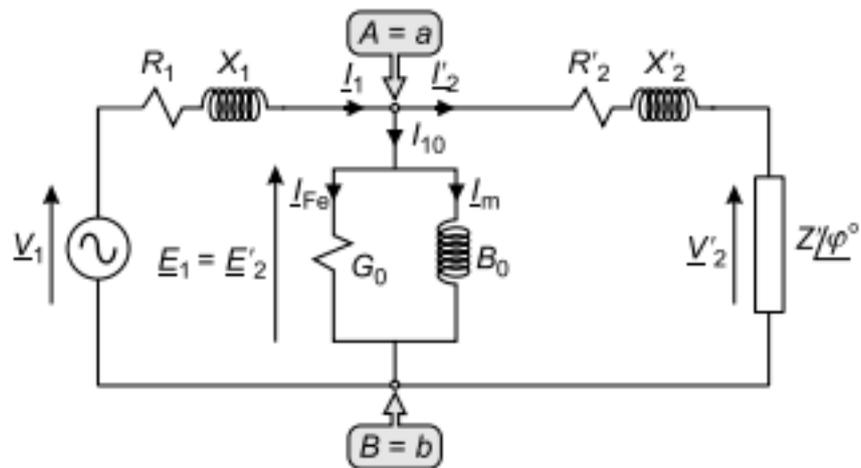


Figura 9. Deducción del circuito equivalente [1].

Debido a las distintas pérdidas se produce una caída de tensión, esto se aprecia con detalle en el diagrama vectorial del transformador.

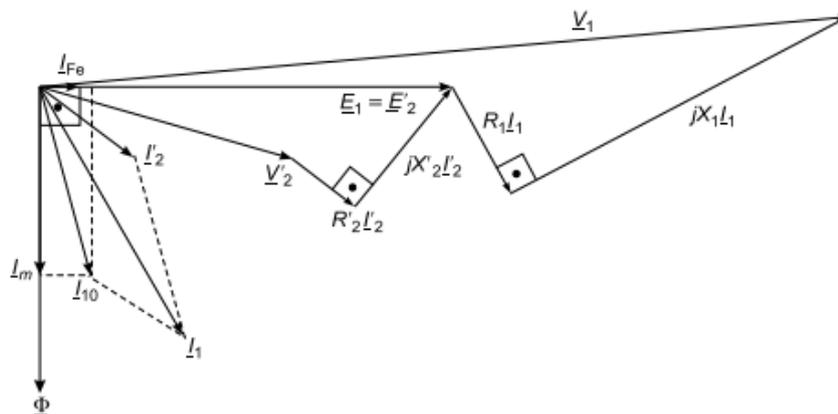


Figura 10. Diagrama vectorial del transformador real en carga [1].

Se puede apreciar como tanto los valores E_1 y V_1 y E'_2 y V'_2 no cumplen la relación de transformación ya mencionada.

5.5. Circuito eléctrico equivalente simplificado del transformador.

Teniendo en cuenta las relaciones existentes entre las diversas magnitudes que intervienen en el transformador, es norma habitual realizar una serie de simplificaciones que no ponen en peligro los resultados obtenidos y facilitan la representación y estudio del transformador [1].

Por un lado, cuando el transformador no está conectado a ninguna carga I_2 es nulo, como consecuencia $I_0 = I_{Fe} + I_m$ es la única intensidad que recorre la resistencia e

inductancia del primario [8]. Estas impedancias son tan pequeñas que se puede considerar que la caída de tensión que producen es prácticamente nula, por lo tanto, se supone que $V_1 \approx E_1$. Por otro lado, se observa que la corriente I_1 es por lo menos veinte veces más mayor que la corriente $I_0 = I_{Fe} + I_m$. Por lo tanto, despreciando la intensidad de vacío frente a la intensidad del primario se puede considerar que $I_1 \approx I'_2$ [8].

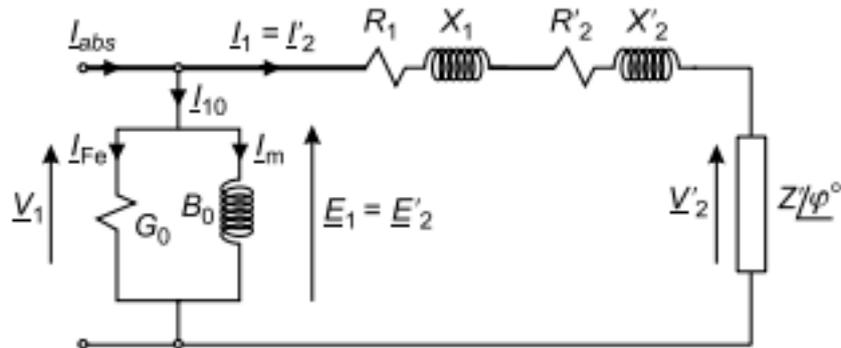


Figura 11. Circuito equivalente simplificado [1].

Donde I_{abs} es la intensidad absorbida por el transformador y quedan definidas la impedancia, resistencia y reactancia equivalentes del transformador (1).

$$R_e = R_1 + R'_2$$

$$X_e = X_1 + X'_2$$

$$Z_e = R_e + jX_e$$

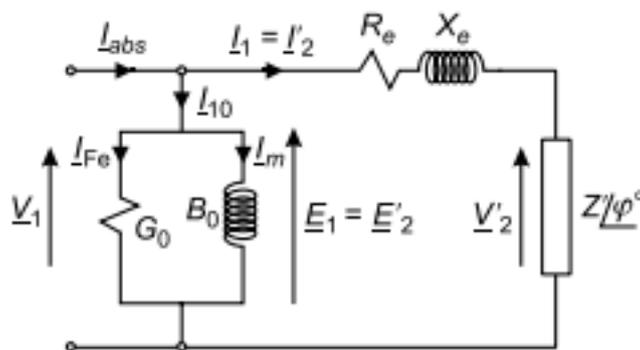


Figura 12. Circuito equivalente simplificado práctico [1].

Los diagramas vectoriales del transformador en base a este último circuito equivalente simplificado son los siguientes:

5.5.1. Conexión a carga inductiva.

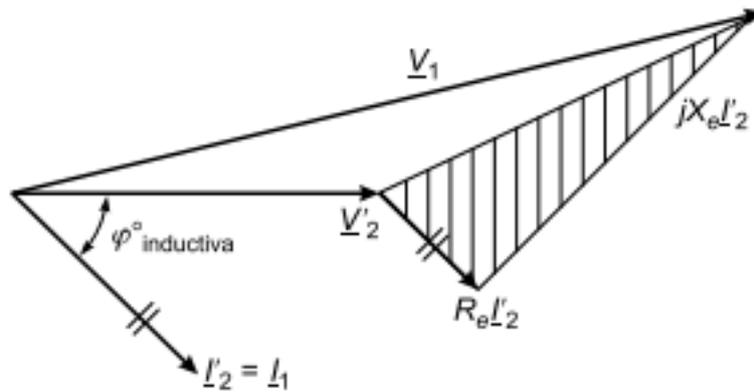


Figura 13. Diagrama vectorial simplificado [1].

Como se observa, con cargas inductivas siempre se cumple que $|V_1| > |V'_2|$.

5.5.2. Conexión a carga capacitiva.

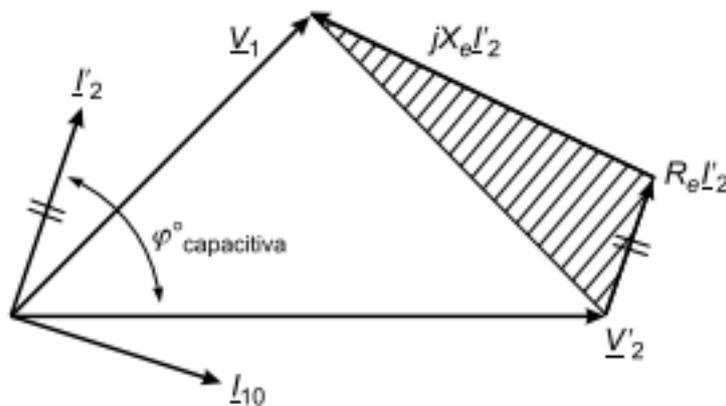


Figura 14. Carga capacitiva: efecto Ferranti [1].

Ante cargas de carácter muy capacitivo, puede llegar a darse el caso en que $|V_1| < |V'_2|$. Esto se conoce como efecto Ferranti.

5.6. Balance de potencias en un transformador.

Las potencias que actúan en condiciones normales de funcionamiento del transformador son las que aparecen en la siguiente figura:

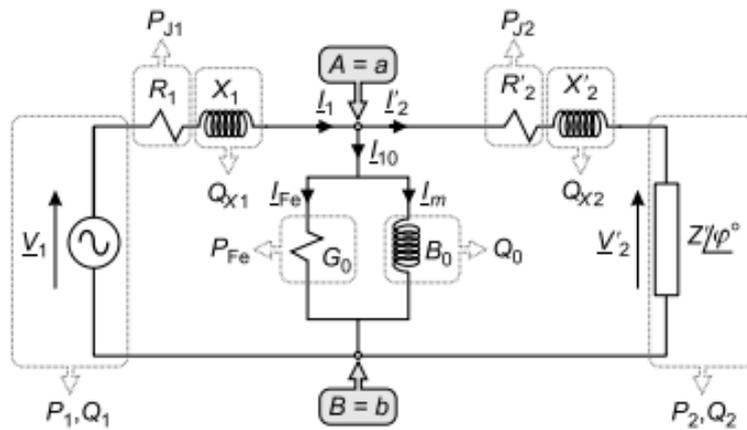


Figura 15. Balance de potencias [1].

A continuación diferenciamos por un lado el balance de potencias activas y por otro el de potencias reactivas.

Balance de potencias activas:

$$P_1 = P_2 + P_{J1} + P_{J2} + P_{Fe} \quad (2)$$

De tal manera que:

- P_{Fe} : Pérdidas en el hierro del circuito magnético.

- P_{J1} y P_{J2} : Pérdidas por efecto Joule en bobinados primario y secundario respectivamente.

- P_1 : Potencia absorbida de la red.

- P_2 : Potencia entregada a la carga.

Balance de potencias reactivas:

$$Q_1 = Q_2 + Q_{X1} + Q_{X2} + Q_0 \quad (3)$$

- Q_0 : Potencia asociada al flujo útil.

- Q_{X1} y Q_{X2} : Potencia asociada al flujo de dispersión primario y secundario respectivamente.

- Q_1 : Potencia absorbida de la red.

- Q_2 : Potencia entregada a la carga.

5.7. Ensayo de vacío.

Este ensayo, como su propio nombre indica se realiza con la máquina trabajando en vacío para obtener los parámetros de la rama de vacío B_0 y G_0 .

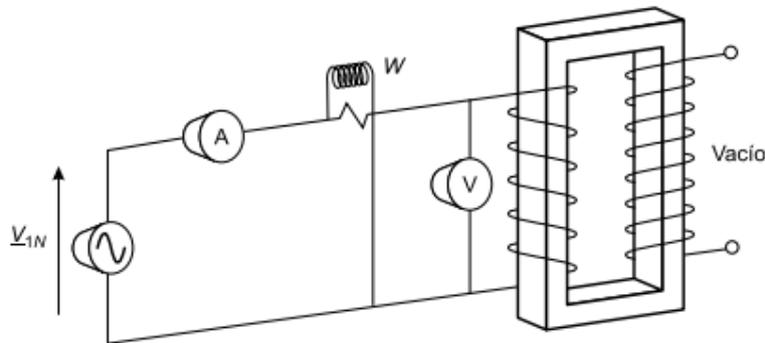


Figura 16. Ensayo de vacío nominal [1].

Como se puede ver en la figura, los dispositivos que se utilizan para realizar la toma de medidas son un vatímetro, un amperímetro y un voltímetro. Con ellos se miden los parámetros llamados W_0 , I_{10} y V_{10} respectivamente.

Realizando el balance de potencia activa se puede concluir que $W_0 \approx P_{Fe}$, y considerando que $E_1 \approx V_{10}$, se obtiene lo siguiente:

$$G_0 = \frac{W_0}{V_{10}^2} \quad (4)$$

$$Y_0 = \frac{I_{10}}{V_{10}} \quad (5)$$

$$B_0 = \sqrt{Y_0^2 - G_0^2} \quad (6)$$

Este ensayo se puede realizar tanto por el bobinado primario como por el bobinado secundario, pero ha de tenerse en cuenta que los parámetros obtenidos estarán referidos a dicho lado.

5.8. Ensayo de cortocircuito.

Este ensayo, como su propio nombre indica se realiza con la máquina trabajando en cortocircuito en uno de sus bobinados para obtener los parámetros de la rama de vacío R_e y X_e .

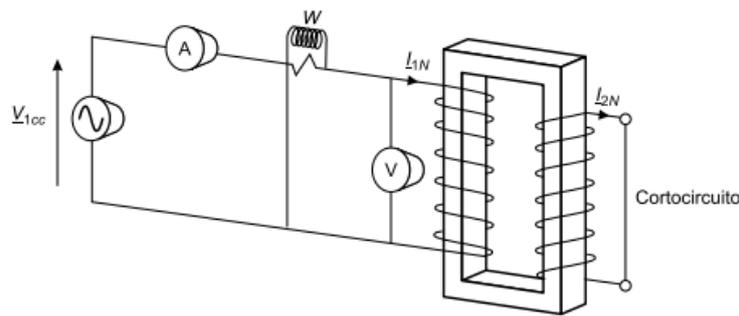


Figura 17. Ensayo de cortocircuito nominal [1].

Como se puede ver en la figura, los dispositivos que se utilizan para realizar la toma de medidas son un vatímetro, un amperímetro y un voltímetro. Con ellos se miden los parámetros llamados W_{cc} , I_{1cc} y V_{1cc} respectivamente.

Realizando nuevamente el balance de potencias se obtiene:

$$R_e = \frac{W_{cc}}{V_{1cc}^2} \quad (7)$$

$$Z_e = \frac{V_{1cc}}{I_{1cc}} \quad (8)$$

$$X_e = \sqrt{Z_e^2 - R_e^2} \quad (9)$$

5.9. Caída de tensión en un transformador. Coeficientes de variación de tensión.

Si alimentamos un transformador a su tensión nominal V_{1N} en vacío proporcionara la tensión secundaria nominal V_{2N} . Pero si este transformador estuviera conectado a una carga el valor de la tensión diferirá de V_{2N} un porcentaje denominado coeficiente de variación de tensión (u). En otras palabras, el interés se centra en conocer el valor eficaz de la tensión real en el secundario en carga respecto del que existirá en condiciones nominales [9].

Considerando el circuito eléctrico equivalente simplificado anteriormente mostrado, se tiene lo siguiente.

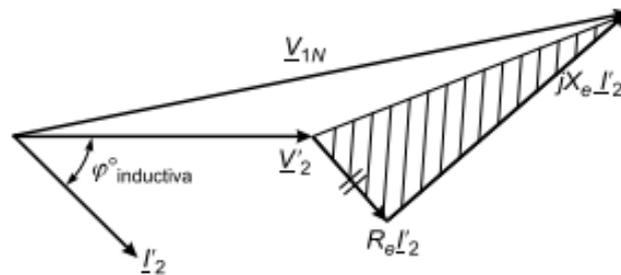


Figura 18. Diagrama vectorial del circuito equivalente simplificado [1].

$$\underline{V}_{1N} = \underline{V}'_2 + \underline{I}'_2 (R_e + jX_e) \quad (10)$$

El valor \underline{V}'_2 y el valor \underline{V}_{1N} no son exactamente iguales. Esto es debido a las caídas de tensión causadas por la intensidad que circula por R_e y X_e . El valor de la tensión será mayor o menor en función de la carga conectada. Si la carga es inductiva la tensión de la salida en el secundario será siempre menor que la tensión ideal en vacío. Si la carga es resistiva pura la tensión también será menor que la tensión ideal en vacío, pero algo mayor que en el caso de carga inductiva y, por último, si la carga es capacitiva puede que se den casos de que la tensión de salida sea mayor que la tensión nominal en vacío produciéndose efecto Ferranti.

Esta caída de tensión se cuantifica mediante el coeficiente de variación de tensión “u”, que permite calcular la tensión en bornes del secundario de la siguiente manera.

$$V_2 = V_{2N} \cdot \left(1 - \frac{u}{100}\right) \quad (11)$$

En los próximos apartados se explicará el método para calcular dicho coeficiente.

-Método analítico de Arnold para el cálculo del coeficiente de variación de tensión.

Se definen los coeficientes de Arnold como:

-Tensión de cortocircuito resistiva: $u_R = \frac{R_e \cdot I_{1N}}{V_{1N}} \cdot 100$

-Tensión de cortocircuito reactiva: $u_X = \frac{X_e \cdot I_{1N}}{V_{1N}} \cdot 100$

-Tensión de cortocircuito: $u_Z = \sqrt{u_R^2 + u_X^2}$

Se define el índice de carga (i) como el cociente entre la intensidad que recorre la línea y la intensidad nominal:

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

$$i = \frac{I_1}{I_{1N}} = \frac{I_2}{I_{2N}} = \frac{I'_2}{I_{1N}} = \frac{S}{S_N}$$

Expresión de Arnold del coeficiente de variación de tensión para cargas inductivas y resistivas:

$$u = i \cdot (u_R \cdot \cos(\varphi) + u_X \cdot \sin(\varphi)) + \frac{i^2}{200} \cdot (u_X \cdot \cos(\varphi) - u_R \cdot \sin(\varphi))^2$$

Para cargas capacitivas:

$$u = i \cdot (u_R \cdot \cos(\varphi) - u_X \cdot \sin(\varphi)) + \frac{i^2}{200} \cdot (u_X \cdot \cos(\varphi) + u_R \cdot \sin(\varphi))^2$$

La expresión de Arnold solo es aplicable si se conoce la tensión de entrada V_1 . En caso de no conocerse debe aplicarse la ecuación vectorial.

$$\underline{V}_{1N} = \underline{V}'_2 + \underline{I}'_2 \cdot (R_e + jX_e) \quad (10)$$

5.10. Rendimiento de un transformador.

Se define el rendimiento de el transformador de la siguiente forma:

$$\eta = \frac{P_{\text{útil}}}{P_{\text{útil}} + P_{\text{pérdidas}}}$$

Siendo de esta manera la $P_{\text{útil}}$ la potencia entregada a la carga, a la que antes hemos denominado como P_2 . La $P_{\text{pérdidas}}$ indican la potencia disipada debido a las pérdidas en el hierro y en el cobre (P_{Fe} y P_{Cu}).

Si se alimenta el transformador por el bobinado primario a su tensión nominal V_{1N} y se conecta a su devanado secundario una carga de carácter inductivo que absorbe una intensidad I_2 :

$$\eta = \frac{\cos(\varphi)}{\cos(\varphi) + \frac{W_0}{i \cdot S_N} + i \cdot \frac{W_{ccN}}{S_N}} \quad (18)$$

El rendimiento será máximo cuando el denominador de la ecuación anterior sea mínimo, para ello se deriva el denominador respecto al índice de carga (i). El rendimiento será máximo cuando se cumpla lo siguiente:

$$i_{\eta_{\max}} = \sqrt{\frac{W_0}{W_{ccN}}} \quad (19)$$

El transformador se suele diseñar para un índice de rendimiento máximo en el entorno de 0,75. Ello es debido a que el rendimiento mantiene un alto valor para condiciones de funcionamiento comprendidas entre el índice de carga de rendimiento máximo y la plena carga ($i=1$), que son valores habituales de funcionamiento para un transformador [10].

Por otra parte, las pérdidas en el hierro y en el cobre que se producen en el punto de máximo rendimiento son iguales.

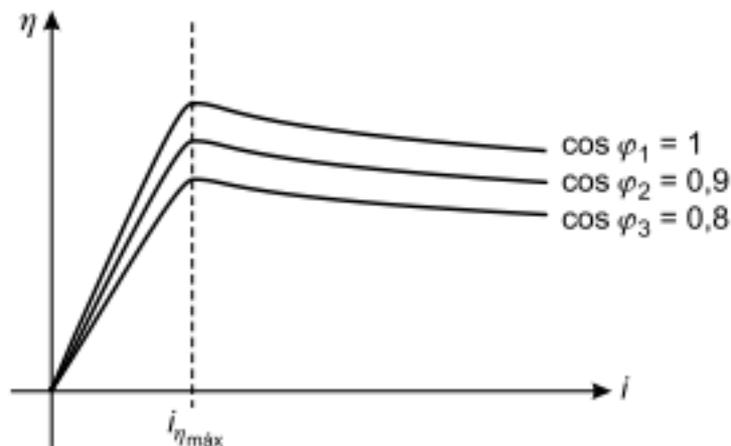


Figura 19. Rendimiento [1].

5.11. El transformador trifásico.

Existen dos formas posibles para construir un transformador trifásico; la primera consiste en construirlo a partir de tres núcleos magnéticos independientes, y la segunda mediante un único núcleo magnético. Deberán realizarse también las conexiones oportunas en los bobinados eléctricos de primario y secundario para obtener así un transformador trifásico.

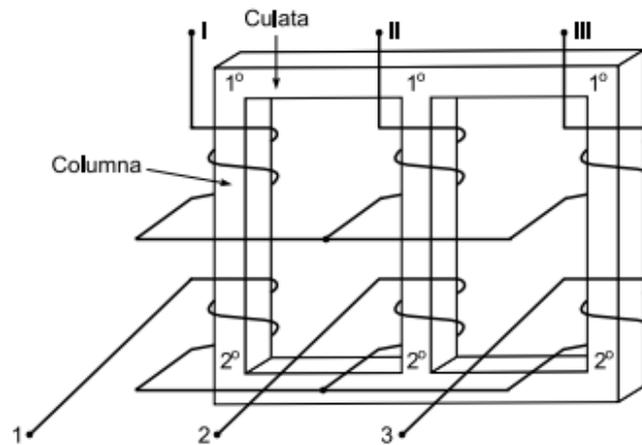


Figura 20. Transformador de núcleo trifásico [1].

La solución más económica es la solución trifásica de un solo núcleo magnético, sin embargo, a la hora de disponer de una unidad de repuesto es una desventaja, ya que es más económico tener un cuarto transformador monofásico que duplicar las existencias de los transformadores trifásicos [3].

5.11.1. Circuito equivalente monofásico.

Debido a que existe una simetría entre las fases, el transformador trifásico se puede representar y analizar mediante su circuito eléctrico monofásico equivalente de dos formas diferentes.

La primera consiste en definir el circuito monofásico equivalente en función de la conexión existente en el lado del transformador respecto al cual se refiere y la segunda opción consiste en no considerar la conexión existente y emplear siempre la hipótesis de estrella equivalente.

Salvo en los casos en que sea estrictamente necesario conocer las tensiones e intensidades reales soportadas por los bobinados, la utilización de la segunda hipótesis es mucho más sencilla, ya que permite prescindir del conocimiento de las conexiones existentes en el transformador y uniformizar el tratamiento del circuito equivalente. En todos los demás casos, cualquier otro tipo de resultado puede ser obtenido utilizando una u otra opción, razón por la cual la hipótesis estrella equivalente es la más comúnmente utilizada[1].

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

$$V_1 = \frac{U}{\sqrt{3}}; \quad (20)$$

$$P = \frac{P_{trifásica}}{3}; \quad (21)$$

$$I_1 = I_{línea}; \quad (22)$$

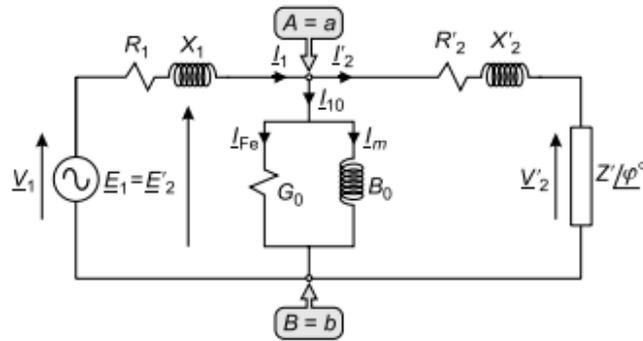


Figura 21. Circuito equivalente monofásico [1].

6. ANÁLISIS DE ALTERNATIVAS.

En este apartado se van a analizar las diferentes alternativas de las que se disponen y seleccionar la más adecuada para llevar a cabo el desarrollo del trabajo.

Como ya se ha mencionado en apartados anteriores, el principal objetivo. De este TFG es el desarrollo de una interfaz gráfica que sea capaz de calcular caídas de tensión, pérdidas de potencia y rendimiento de transformadores monofásicos y trifásicos. En este apartado se justificarán las razones por las que es más ventajoso disponer de una herramienta de simulación software que realizar otro tipo de técnicas de medición como lo son las pruebas de campo o los modelos físicos a escala.

6.1. Presentación de las alternativas.

6.1.1. Pruebas de campo.

Consiste en llevar a cabo las propias medidas sobre la máquina para realizar de forma posterior y en base a esos datos los cálculos pertinentes. Se puede poner a prueba el transformador variando su estado de trabajo (variando tensiones de entrada, carga, etc.). Los resultados se ajustan a la realidad pero conlleva unos gastos de recursos económicos y de recursos humanos.

6.1.2. Modelos físicos a escala.

Consiste en la fabricación de un modelo físico a escala del transformador a estudiar. Sobre este mismo modelo se realizarán las pruebas deseadas, para ello habrá que estudiar las relaciones de semejanza entre modelo y realidad para obtener unos resultados fiables. Será un inconveniente la alta inversión económica a realizar para la fabricación de un modelo a escala, ya que, sería necesario realizar un modelo a escala por cada transformador a analizar.

6.1.3. Simulación software.

Consiste en el diseño y desarrollo de una interfaz gráfica software que sirve como herramienta de cálculo de las caídas de tensión, pérdidas de potencia y rendimiento de un transformador ante diferentes estados de funcionamiento. Introduciendo las variables de entrada, se obtiene de forma sencilla e intuitiva los resultados deseados. Es la única opción que no realiza pruebas en un entorno real pero requiere menos inversión y los resultados no se alejan de la realidad.

6.1.4. Criterios de selección.

Los diferentes criterios a la hora de estudiar los distintos métodos de análisis han sido los siguientes:

-Coste económico: la alternativa seleccionada tendrá una mayor viabilidad y será más ventajosa cuánto menores sean sus costes. En este ámbito se estudia la viabilidad económica de cada alternativa y sus ventajas comparativas frente al resto.

-Recursos humanos y materiales: son el conjunto de personal y material utilizado para realizar el trabajo. Aquella tarea que precise menores recursos pero mantenga la calidad de los resultados será la más adecuada.

-Cantidad de resultados: la mejor elección será aquella que permita obtener la mayor cantidad de resultados en el menor tiempo posible. Cuanto mayor sea la cantidad de datos que el usuario pueda observar, mayor fiabilidad y respaldo tendrán los parámetros calculados.

-Precisión y fiabilidad de los resultados: es necesario que los resultados que se obtengan sean lo más parecido posible a la realidad. No solo es importante la cantidad de los resultados que se puedan obtener sino que también lo es su fiabilidad y su precisión. La alternativa escogida ha de tener unos fundamentos que proporcionen validez a sus resultados siendo los errores obtenidos los mínimos posibles. La fiabilidad y precisión de los resultados es la base para la correcta interpretación de los parámetros de la máquina.

6.1.5. Conclusión del análisis.

Este apartado tiene como objetivo justificar que la elección de utilizar la herramienta software es la más adecuada para el estudio de caídas de tensión, pérdidas y rendimiento de transformadores. Para demostrarlo, se realizará una comparación entre este método y el resto de alternativas atendiendo a los criterios anteriormente mencionados.

Se han recogido en una tabla las comparaciones entre las distintas alternativas para el análisis de los transformadores. Cada uno de los criterios ha sido ponderado de una manera diferente y se le otorga una puntuación a cada alternativa en función del cumplimiento de dicho criterio.

La puntuación utilizada es un rango de 0 a 10 puntos, siendo 0 puntos la peor de las puntuaciones y 10 puntos una puntuación que indica que se cumple el criterio de una manera excelente.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

	PESO	VALORACIONES SOBRE 10		
		Ensayos de campo	Modelos a escala	Simulación software
Coste económico	40%	0	4	8
Recursos humanos/materiales	10%	0	5	10
Cantidad de resultados	25%	4	5	9
Fiabilidad y precisión	25%	9	4	7
TOTAL (Sobre 10)		3,25	4,35	8,2

Tabla 1. Conclusión del análisis.

A la vista de estos resultados queda demostrado que la alternativa de la utilización de una herramienta de simulación software es la alternativa más adecuada para el análisis a realizar en los transformadores.

El objetivo de este Trabajo Fin de Grado es desarrollar una interfaz gráfica que permita determinar las caídas de tensión, pérdidas de potencia y rendimiento en transformadores monofásicos y trifásicos, introduciendo los datos característicos de la máquina y las condiciones de carga bajo las que trabaja.

Como se ha podido comprobar, en la tabla anterior queda totalmente evidenciado que el desarrollo de una simulación software es la mejor de las soluciones. Es la alternativa notablemente más económica, que mayor cantidad de resultados ofrece y menores recursos utiliza. Aún no siendo una herramienta tan fiable como los ensayos de campo, obtiene una puntuación más que aceptable.

Los modelos a escala proporcionan una cantidad de datos aceptables y hacen uso de una cantidad de recursos que no es excesiva, sin embargo el coste económico es elevado para la insuficiente precisión que ofrece.

Los ensayos de campo proporcionan unos resultados de alta fiabilidad, pero no son viables ya que ofrecen pocos resultados y los gastos de recursos humanos, materiales y los costes económicos son muy elevados.

Las puntuaciones finales son las siguientes:

- Ensayos de campo: 3,25.
- Modelos a escala: 4,35.
- Simulación software: 8,2.

La puntuación obtenida por la alternativa de el diseño de una interfaz gráfica software para el cálculo de caídas de tensión, pérdidas y rendimiento en transformadores monofásicos y trifásicos es claramente superior al resto de alternativas, por tanto, esta será la alternativa escogida. En el siguiente apartado, se analizarán las herramientas software.

6.2. Elección de la herramienta software.

Tras demostrar que la simulación software es la alternativa óptima para el análisis de las caídas de tensión, pérdidas de potencia y rendimiento de transformadores de potencia, en este apartado se analizarán diferentes software que permitirían desarrollar dicha simulación. Al igual que se ha hecho en el apartado anterior con el análisis de las alternativas, se realizará en este un estudio de las herramientas disponibles y se detallará cuál de ellas es la óptima para llevar a cabo el trabajo.

6.2.1. ATP (Alternative Transients Program).

Este programa permite la realización de cálculos electromagnéticos y electromecánicos con fines de diseño, especificaciones de equipo y definición de parámetros eléctricos. Para realizar las simulaciones se dispone de un gran número de modelos predefinidos.

El programa ATP-EMTP determina el valor en el dominio del tiempo y de la frecuencia de todas las magnitudes en un sistema con un número arbitrario de fases, modelizado a través de resistencias, inductancias y capacidades, representando también interruptores y fuentes. Cada uno de estos elementos básicos posee variaciones en su representación según las necesidades de modelización de los distintos elementos encontrados en el sistema eléctrico.

6.2.2. PSCAD/EMTDC.

Básicamente, se puede decir que los modelos y técnicas de simulación utilizadas en este software son similares a las utilizadas en el ATP-EMTP. Este software consta de dos módulos que son el EMTDC (simulador de sistemas eléctricos) y el PSCAD (interfaz gráfico de usuario)[11].

En general, es una herramienta orientada al campo de la energía eléctrica, destacando entre sus aplicaciones la modelización de líneas y cables,

transformadores saturables, dispositivos de electrónica de potencia, generación distribuida, máquinas rotativas o sistemas integrados.

6.2.3. Mathcad.

Es una herramienta informática que lleva incorporadas todas las funcionalidades de las hojas de cálculo, una gran potencia debido a su extensa librería de funciones y una alta compatibilidad con muchas de las herramientas más utilizadas en la ingeniería [11].

6.2.4. MATLAB-GUIDE.

MATLAB es un programa interactivo que combina el cálculo numérico con representaciones gráficas. De esta forma integra el análisis numérico, el cálculo matricial y el procesado de señales con gráficos que facilitan la comprensión de dichos cálculos. GUIDE (Graphical User Interface Development Environment) es una herramienta que se incluyen en MATLAB que permite diseñar interfaces gráficas de usuario añadiendo figuras, cajas de texto, botones, distintos tipos de menús, etc [11].

6.2.5. Criterios de selección.

Los criterios que se han tenido en cuenta a la hora de evaluar las diferentes herramientas para desarrollar la interfaz han sido los siguientes:

-Flexibilidad: la herramienta debe permitir realizar ciertos cambios puntuales en el diseño sin que suponga realizar una alteración completa en la aplicación diseñada.

-Tiempos de simulación: se valorará positivamente la rapidez con la que la interfaz devuelve los datos deseados.

-Posibilidad de implementar una interfaz: este criterio es fundamental ya que es la base del desarrollo de este TFG. Por tanto, la herramienta seleccionada ha de proporcionar la posibilidad de programar una interfaz gráfica.

-Simplicidad de la herramienta: el uso de la interfaz ha de ser sencillo e intuitivo.

-Coste económico: cuanto menor sea el coste de la licencia del software más positivamente se valorará.

6.2.6. Resultados del análisis.

Para realizar una correcta evaluación se asocian unos porcentajes de ponderación referidos a los criterios anteriormente mencionados y se clasifican cada una de las opciones con una nota en función del grado del cumplimiento de cada criterio. Al igual que antes, la puntuación será un valor entre los 0 y 10 puntos.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

	PESO	VALORACIONES SOBRE 10			
		ATP	PSCAD/EMTDC	Mathcad	MATLAB (GUIDE)
Flexibilidad	15%	10	10	10	10
Tiempos de simulación	10%	0	4	4	6
Posibilidad de interfaz	50%	0	6	0	10
Simplicidad de herramienta	10%	1	4	5	8
Coste económico	15%	10	5	5	5
TOTAL (Sobre 10)		3,1	6,05	3,15	8,65

Tabla 2. Resultados del análisis.

Atendiendo a la tabla, la herramienta MATLAB obtiene una puntuación de 8,65 puntos. Es la herramienta mas adecuada para cumplir con los objetivos de este trabajo. Siendo cierto que la licencia necesaria para poder utilizar este programa tiene un coste económico elevado, la herramienta proporciona la expansión (GUIDE). Mediante GUIDE no solo se puede desarrollar una interfaz fácil, intuitiva y sencilla de utilizar sino que también ofrece muchas facilidades a la hora de ser programada.

De entre las opciones mostradas, tanto ATP como Mathcad no permiten el desarrollo de una interfaz gráfica, por lo tanto quedan descartadas. A la hora de elegir entre PSCAD/EMTDC y GUIDE, se elige la herramienta GUIDE ya que esta es más sencilla e intuitiva.

Visto esto, la herramienta seleccionada para el desarrollo de una interfaz gráfica que calcule las caídas de tensión, pérdidas de potencia y rendimiento en transformadores de potencia, será la herramienta GUIDE de MATLAB.

Los resultados de las distintas opciones han sido los siguientes:

- ATP: 3,1 puntos sobre 10.
- PSCAD/EMTDC: 6,05 puntos sobre 10.
- Mathcad: 3,15 puntos sobre 10.
- MATLAB (GUIDE): 8,65 puntos sobre 10.

7. DESCRIPCIÓN DE LA SOLUCIÓN.

Como ya se ha mencionado en el apartado anterior, MATLAB es la herramienta con la que se llevará a cabo el desarrollo de la interfaz gráfica. En este apartado se describe el diseño y desarrollo de la interfaz al completo.

7.1. Introducción a la plataforma GUIDE.

GUIDE (Graphical User Interface Development Environment) es una herramienta que proporciona el software MATLAB que permite diseñar y programar interfaces gráficas.

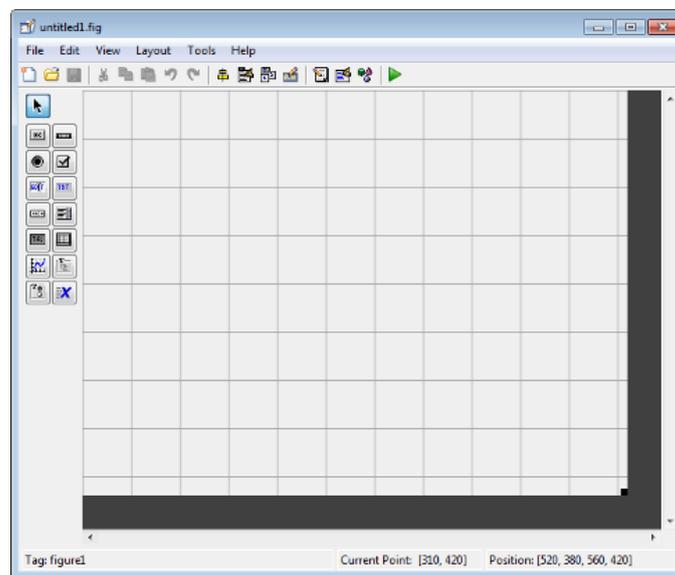


Figura 22. Interfaz editable GUIDE.

Sobre esta interfaz mostrada en la imagen anterior se añaden distintos elementos que serán programados posteriormente. Los elementos añadidos en este proyecto han sido los siguientes.

-*Paneles*: organizan la interfaz en distintos módulos agrupando distintos tipos de elementos .

-*Radio button*: permite seleccionar entre distintas opciones.

-*Static text*: celdas sobre las que el programador escribe enunciados que el usuario podrá leer cuando se ejecute el programa.

-*Edit text*: celdas sobre las que el usuario puede escribir.

-*Push Button*: ejecutan las funciones programadas asociadas al ser pulsados por el usuario.

Cada uno de los anteriores elementos tienen asociadas funciones que se deben programar. Dichas funciones serán ejecutadas al activar los distintos elementos de la interfaz.

7.2. Interfaz gráfica de usuario.

La interfaz gráfica que se ha llevado a cabo en este proyecto permite conocer las caídas de tensión, pérdidas y rendimiento de transformadores de potencia, tanto monofásicos como trifásicos. También muestra ciertos parámetros internos del transformador que son de interés para el usuario. Todos estos parámetros se pueden calcular bajo distintos modos de funcionamiento de la máquina. La interfaz gráfica de usuario tiene el siguiente aspecto:

The screenshot shows a software interface for transformer calculations. It is organized into several panels:

- Datos del transformador:** Includes a dropdown for 'Tipo de transformador' (Monofásico or Trifásico), input fields for primary/secondary voltages (U1n, U2n), power (Sn), current (In), and a 'Relación de transformación' label.
- Ensayos*:** Contains sub-sections for 'Cortocircuito' (short-circuit) and 'Vacío' (no-load) tests, each with input fields for voltage, current, and power.
- Parametros internos:** A table listing internal parameters: Req (Ohm), Xeq (Ohm), Zeq (Ohm), Go (S), Bo (S), and Yo (S).
- Tipo de problema:** Radio buttons for 'Tension en bornes' (selected) and 'Tension primario'.
- Cálculo de la tensión en bornes de la carga.:** Input fields for primary voltage, load power, and power factor, followed by a 'Tipo de carga.' dropdown (Capacitiva, Inductiva, Resistiva) and output fields for secondary voltage and current.
- Cálculo de la tensión de primario.:** Input fields for terminal voltage, load power, and power factor, followed by a 'Tipo de carga.' dropdown and output fields for primary voltage and current.
- Rendimiento:** A table for efficiency with columns for percentage and rows for 'Máximo' and 'Uz'.
- Caídas de tensión.:** A table for voltage drops with columns for percentage and rows for 'Ur', 'Ux', and 'Uz'.
- Pérdidas:** A table for losses with columns for power (W) and rows for 'Wo', 'Wcc', and 'Pérdidas totales'.
- Calcular:** Two buttons: 'Calcular' (blue) and 'Reset' (yellow).

Figura 23. Interfaz gráfica de usuario.

La herramienta está compuesta de diferentes módulos. El primero (parte superior izquierda) es donde se introducen los datos de la máquina a estudiar; si se trata de un transformador monofásico o trifásico, las tensiones nominales y su potencia aparente nominal. Este módulo devuelve los valores de intensidad nominal y relación de transformación. En el segundo módulo (parte inferior izquierda) se recogen los datos de los ensayos y se calculan los parámetros internos del transformador que se aprecian en la figura anterior. El tercer módulo se sitúa en la zona superior derecha, permite seleccionar el tipo de problema. En función de lo seleccionado en el tercer módulo se activará el cuarto módulo en el cual se introducen los datos de la carga y las tensiones de alimentación, y por último en el quinto módulo se seleccionará el carácter de la carga. Para finalizar, pulsando el botón “Calcular”, que está situado abajo a la derecha en color azul, se mostrarán los resultados indicados de pérdidas, rendimiento, caídas de tensión e intensidad. Pulsando el botón “Reset” la herramienta se reiniciará dejando en blanco todos los campos. Se adjunta el siguiente diagrama de flujo para un mejor análisis del funcionamiento de la herramienta de cálculo:

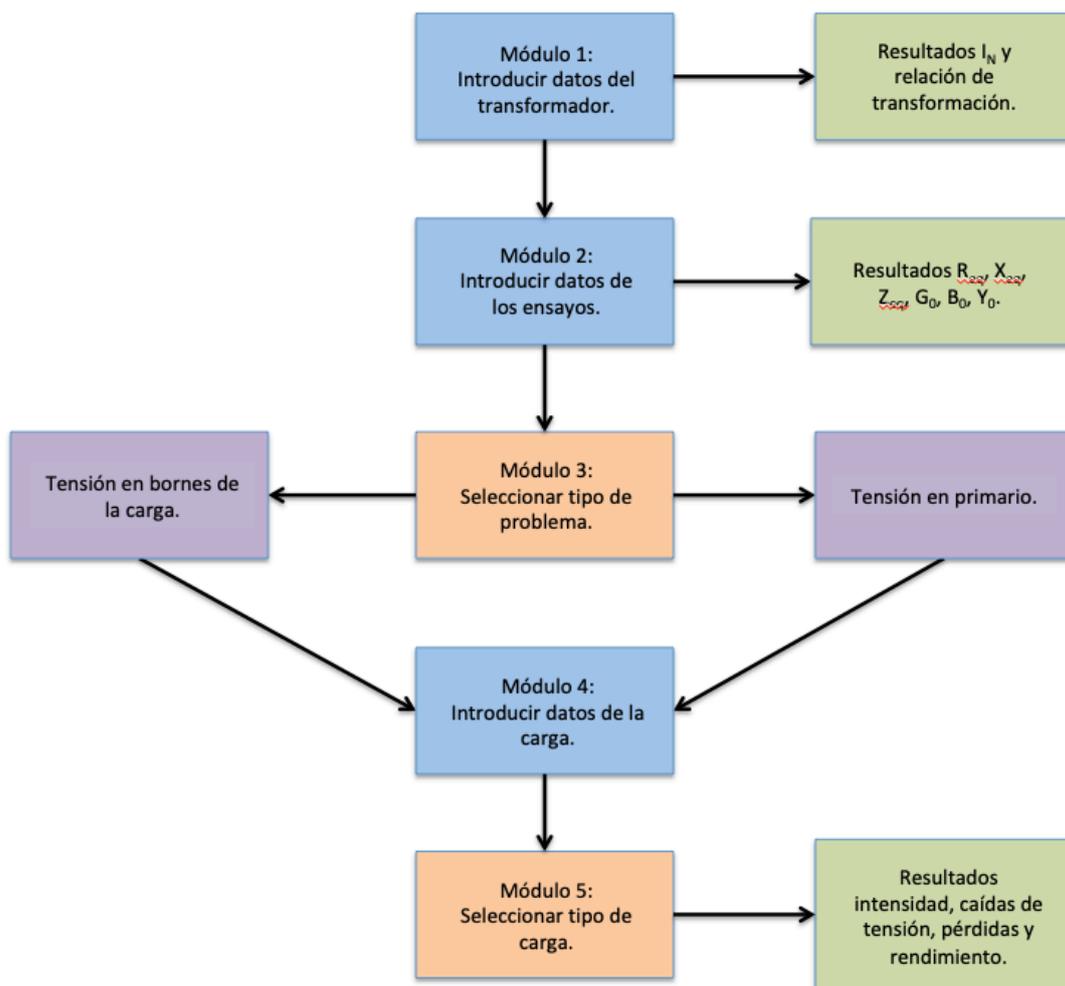


Figura 24. Diagrama de flujo.

7.2.1. Módulo 1: Datos del transformador.

En este módulo se selecciona el tipo de transformador (monofásico o trifásico) y se introducen los valores de las tensiones nominales de primario y secundario y la potencia aparente nominal en las unidades indicadas. La herramienta devolverá de forma automática el valor de la intensidad nominal y la relación de transformación del transformador.



Figura 25. Módulo 1.

En primer lugar se realiza la selección del tipo del transformador seleccionando el *radio button* que contenga la opción deseada:

```
A=get(hObject,'String');
switch A
    case 'Monofasico'
        set(handles.text4,'string','V1n');
        set(handles.text93,'string','V2n');
    case 'Trifasico'
        set(handles.text4,'string','U1n');
        set(handles.text93,'string','U2n');
end
```

La función *get* se encargan de recoger los datos introducidos en los *Edit Text* y la función *set* se encarga de escribir valores o enunciados en los llamados *Static Text*. Esta función tiene dos objetivos. El primero es el de guardar una variable que contenga la información del tipo de máquina que se está estudiando y el segundo es el de cambiar los enunciados de las tensiones nominales del transformador; si el transformador es monofásico se representará la tensión mediante la letra “V”, sin

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

embargo, si es trifásico se representará mediante la letra “U”, como puede apreciarse en la figura.

La recogida de datos de las tensiones nominales y de la potencia aparente nominal se realiza de la siguiente manera:

```
Vn=get(handles.Vn,'String');
Vn=str2double(Vn)*1000;
if A1==3
    Vn=Vn/sqrt(3);
end

V2n=get(handles.V2n,'String');
V2n=str2double(V2n)*1000;
if A1==3
    V2n=V2n/sqrt(3);
end

SnMono=get(handles.editSnMono,'String');
SnMono=str2double(SnMono)*1000;
if A1==3
    SnMono=SnMono/3;
end
```

No solo se almacenan las variables anotadas sino que también se realizan los cambios pertinentes de unidades. Además, en función de la variable del tipo de transformador (“A1”), se convierten los datos de los transformadores trifásicos a los monofásicos equivalentes si la opción de tipo de transformador trifásico se encuentra seleccionada.

A continuación se calculan y anotan los valores de intensidad nominal y relación de transformación.

```
rt=Vn/V2n;
In=SnMono/Vn;
set(handles.rt,'string',rt);
set(handles.text63,'string',In);
```

Siendo “rt” e “In” la relación de transformación y la intensidad nominal respectivamente. Es importante distinguir que si el transformador es trifásico habrá que aplicar las expresiones (20), (21) y (22) mencionados en el apartado 5.11.

7.2.2. Módulo 2: Ensayos de vacío y cortocircuito.

En este módulo se introducen los datos de los ensayos de cortocircuito y vacío realizados por la rama de primario. Se introducen datos de tensión, intensidad y pérdidas y se calculan los parámetros que se muestran en la figura de forma

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

automática. Para realizar los cálculos se emplean las ecuaciones (4), (5), (6), (7), (8) y (9) mostradas en los apartados 5.8 y 5.9.

The image shows a software interface titled "Ensayos*" with two main sections: "Cortocircuito" and "Vacio". Each section contains three input fields with units. Below these is a note "*Realizados por la rama primaria." and a "Parametros internos" section with six parameters and their units.

Cortocircuito	
Tensión cortocircuito:	<input type="text"/> kV
Intensidad cortocircuito:	<input type="text"/> A
Pérdidas cortocircuito:	<input type="text"/> kW

Vacio	
Tensión vacío:	<input type="text"/> kV
Intensidad vacío:	<input type="text"/> A
Pérdidas vacío:	<input type="text"/> kW

*Realizados por la rama primaria.

Parametros internos	
R eq:	Ohm
X eq:	Ohm
Z eq:	Ohm
G o:	S
B o:	S
Y o:	S

Figura 26. Módulo 2.

El código funciona de forma análoga al módulo anterior; se almacenan los datos introducidos y se realizan las conversiones de unidades necesarias, se transforman los datos a un circuito monofásico equivalente si fuera necesario y por último la herramienta realiza los cálculos y muestra los resultados sobre la interfaz. En este caso no se adjunta ningún código ya que como se ha dicho es análogo al anterior pero con diferentes variables, el código completo que se ha desarrollado para el desarrollo de la interfaz se muestra en el anexo.

7.2.3. Módulo 3: selección del tipo de problema.

En este módulo el usuario debe seleccionar el tipo de problema que desea resolver. Por un lado, si selecciona el tipo “Tensión en bornes”, el usuario introducirá los datos de la carga y la tensión de alimentación de la rama de primario para así conocer la tensión en bornes de la carga y la intensidad en secundario. Para resolver este tipo de ejercicio se utilizarán las fórmulas (11), (12), (13), (14), (15) y (16) o (17) según el tipo de carga, todas ellas explicadas en el apartado 5.9. Por otro lado, si selecciona el tipo “Tensión primario”, el usuario introducirá los datos de la carga y la tensión de alimentación de dicha carga en la rama de secundario para así conocer la tensión de alimentación y la intensidad en bornes de la rama primaria. Para resolver este segundo tipo de ejercicio se empleará la ecuación vectorial (10) del apartado 5.9.

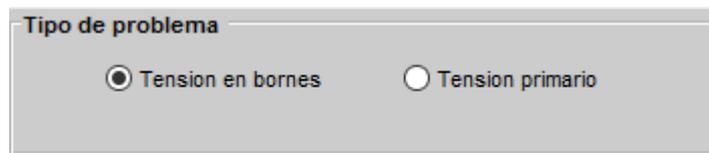


Figura 27. Módulo 3.

Seleccionando la opción deseada, el panel que contiene el tipo de problema a resolver queda habilitado, mientras que el panel que no ha sido seleccionado queda inhabilitado. Esta acción se realiza mediante el comando “*Enable*” en la programación de los *radio button* del panel mostrado.

```
tipo=get(hObject,'String');
switch tipo
    case 'Tension en bornes'
        set(handles.tensionbornescarga,'Enable','Off');
        set(handles.cargasec,'Enable','Off');
        set(handles.fdpsec,'Enable','Off');
        set(handles.TensionDato,'Enable','On');
        set(handles.CargaPrim,'Enable','On');
        set(handles.fdpPrim,'Enable','On');

        set(handles.radiobutton10,'Enable','Off');
        set(handles.radiobutton11,'Enable','Off');
        set(handles.radiobutton17,'Enable','Off');
        set(handles.radiobutton7,'Enable','On');
        set(handles.radiobutton9,'Enable','On');
        set(handles.radiobutton16,'Enable','On');

    case 'Tension primario'
        set(handles.tensionbornescarga,'Enable','On');
        set(handles.cargasec,'Enable','On');
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
set(handles.fdpsec, 'Enable', 'On');  
set(handles.TensionDato, 'Enable', 'Off');  
set(handles.CargaPrim, 'Enable', 'Off');  
set(handles.fdpPrim, 'Enable', 'Off');  
  
set(handles.radiobutton10, 'Enable', 'On');  
set(handles.radiobutton11, 'Enable', 'On');  
set(handles.radiobutton17, 'Enable', 'On');  
set(handles.radiobutton7, 'Enable', 'Off');  
set(handles.radiobutton9, 'Enable', 'Off');  
set(handles.radiobutton16, 'Enable', 'Off');
```

end

A continuación se muestra una figura en la que el panel se encuentra en estado habilitado y otra en la que se encuentra en estado inhabilitado.

Cálculo de la tensión de primario.

Tensión en bornes de la carga: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension primario: kV

Intensidad primario: A

Figura 28. Panel habilitado.

Cálculo de la tensión de primario.

Tensión en bornes de la carga: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension primario: kV

Intensidad primario: A

Figura 29. Panel inhabilitado.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

7.2.4. Módulos 4 y 5: condiciones de carga del transformador.

Independientemente del tipo de problema seleccionado la mecánica de funcionamiento de los módulos es la misma, la única diferencia radica en los datos de entrada y de salidas, y las operaciones que realiza la herramienta.

A continuación se detalla el funcionamiento de cada tipo de problema. Las fórmulas que se aplican son las mencionadas en el apartado anterior.

Para el cálculo de la tensión en bornes de la carga se introduce la tensión de alimentación por la rama de primario y los datos de la carga conectada en secundario.

```
TensionDato=get(handles.TensionDato,'String');
TensionDato=str2double(TensionDato)*1000;

if A1==3
    TensionDato=TensionDato/sqrt(3);
end
CargaDato=get(handles.CargaPrim,'String');
CargaDato=str2double(CargaDato)*1000;

if A1==3
    CargaDato=CargaDato/3;
end
fdp=get(handles.fdpPrim,'String');
fdp=str2double(fdp);
```

A continuación se selecciona el carácter de la carga en el panel llamado “Tipo de carga”. Para ello hay que programar el panel compuesto por los diferentes *radio button*. Este dato se almacena para cálculos posteriores.

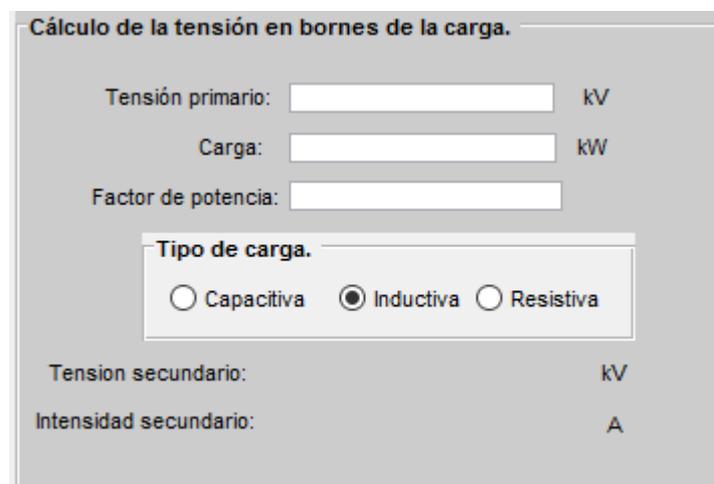
```
if get(handles.radiobutton16,'Value')==1
    B1=1;
elseif get(handles.radiobutton7,'Value')==1
    B1=2;
else
    B1=3;
end
```

En caso de que se seleccione la opción llamada “Resistiva”, el programa autocompletará el campo llamado “Factor de potencia” con el valor 1. Veamos el caso en el que se realiza esta acción con el valor del factor de potencia del primer tipo de problema.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
if B1==1
    fdp=1;
    set(handles.fdpPrim, 'String', fdp);
end
```

Por último faltan por rellenar los campos de tensión e intensidad del circuito secundario, estos cálculos se realizan y muestran en pantalla en funciones posteriores que se explicarán a continuación.



Cálculo de la tensión en bornes de la carga.

Tensión primario: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension secundario: kV

Intensidad secundario: A

Figura 30. Módulo problema tipo I.

El tipo de problema restante, que consiste en calcular la tensión a la que habrá que alimentar por la rama primaria al transformador siendo en este caso el dato de la tensión la que se tiene en bornes de la carga, se realiza de forma exactamente análoga al problema anterior, la única diferencia será una des las variables de entrada y las fórmulas que aplicará la herramienta de cálculo para resolver el problema.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Cálculo de la tensión de primario.

Tensión en bornes de la carga: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension primario: kV

Intensidad primario: A

Figura 31. Módulo problema tipo II.

7.2.5. Presentación de resultados.

Los siguientes paneles dan paso a la muestra en pantalla de los resultados del problema; la tensión en intensidad en la rama deseada, las caídas de tensión, el rendimiento en la situación de carga y máximo [(18), (19)] y las pérdidas (2).

Rendimiento		Caídas de tensión.	
	%	Ur:	%
Máximo:	%	Ux:	%
		Uz:	%

Pérdidas	
Wo:	W
Wcc:	W
Pérdidas totales:	W

Figura 32. Panel de resultados.

Calcular

Figura 33. Panel de *Push Button*.

Pulsando el botón “Calcular”, la herramienta realiza los cálculos y muestra en pantalla los resultados. A continuación se muestra el código para la resolución de el primer tipo de problema (cálculo de la tensión en bornes de la carga), el código completo podrá consultarse en el anexo adjunto. Finalmente se muestran los resultados.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Primero se realizan los cálculos:

```
if tipo1==1

int=CargaDato/(TensionDato*fdp);
In=SnMono/Vn;

req=Wcc/Icc^2;
zeq=Vcc/Icc;
xeq=sqrt(zeq^2-req^2);

fi=acos(fdp);
ifactcarga=int/In;

ux=(xeq*In)/Vn;
ur=(req*In)/Vn;
uz=(zeq*In)/Vn;

tensionnominalrt=TensionDato/rt;

if B1==3
u=ifactcarga*(ur*100*fdp+ux*100*sin(fi))+((ifactcarga^2)/200)*(u
x*100*fdp-ur*100*sin(fi))^2;
elseif B1==1
u=ifactcarga*(ur*100*fdp)+((ifactcarga^2)/200)*(ux*100*fdp)^2;
else
u=ifactcarga*(ur*100*fdp-
ux*100*sin(fi))+((ifactcarga^2)/200)*(ux*100*fdp+ur*100*sin(fi))^2;
end

tensioncalculada=tensionnominalrt*(1-(u/100));

if A1==3
tensioncalculada=tensioncalculada*sqrt(3);
end

rdto=100*((ifactcarga*SnMono*fdp)/(ifactcarga*SnMono*fdp+((Wo*(Te
nsionDato)^2)/(Vo)^2)+((ifactcarga^2)*(req*(In)^2))));
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
inmax=sqrt(((Wo*(Vn)^2)/(Vo)^2)/(req*(In)^2));

rdtomax=100*((inmax*SnMono*fdp)/(inmax*SnMono*fdp+((Wo*(Vn)^2)/(Vo)^2)+(inmax)^2*(req*(In)^2)));

go=Wo/(Vo)^2;
yo=Io/Vo;
bo=sqrt(yo^2-go^2);

perdidas=((Wo*(TensionDato)^2)/(Vo)^2)+req*(int)^2;
```

Finalmente se muestran los resultados:

```
set(handles.Ur,'string',ur*100);
set(handles.Ux,'string',ux*100);
set(handles.Uz,'string',uz*100);

set(handles.tensionsecundario,'string',tensioncalculada/1000);
set(handles.text66,'string',int/rt);

set(handles.text88,'string',req*(int)^2);
set(handles.text89,'string',((Wo*(TensionDato)^2)/(Vo)^2));
set(handles.perdidas,'string',perdidas);

set(handles.rt,'string',rt);
set(handles.text63,'string',In);

set(handles.Req,'string',req);
set(handles.Xeq,'string',xeq);
set(handles.Zeq,'string',zeq);

set(handles.go,'string',go);
set(handles.yo,'string',yo);
set(handles.bo,'string',bo);

set(handles.Rendimiento,'string',rdto);
set(handles.Maximo,'string',rdtomax);
```

Pulsando el botón “Reset” la interfaz retornará a su estado inicial, es decir, se borrarán los campos en los que se encuentren escritos tanto los datos para la realización del problema como los resultados obtenidos tras la realización de problema. El procedimiento a seguir a la hora de programar este botón es el siguiente; se muestra un ejemplo con una de las variables.

```
SnMono=char(' ');  
set(handles.editSnMono,'String',SnMono);
```

Para el resto de las variables el procedimiento es exactamente el mismo.

7.2.6. Mensajes de error.

Los mensajes de error son una herramienta añadida con el fin de avisar al usuario de la interfaz que alguno de los datos introducidos no ha sido introducido de forma correcta. Para ello, se utiliza la función *errordlg*.

Se han programado dos mensajes distintos de error; el primero advierte de que el dato introducido no es válido, por ejemplo, si se introduce una letra en lugar de un número. La figura que se muestra en pantalla y el código utilizado es el siguiente:

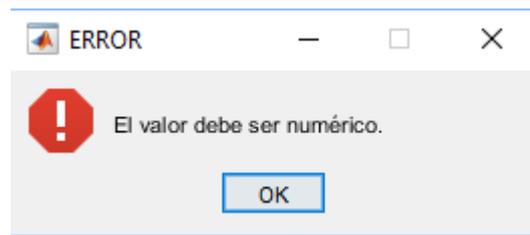


Figura 34. Error de tipo numérico.

```
Vn=get(handles.Vn,'String');  
if isnan(Vn)  
    errordlg('El valor debe ser numérico.','ERROR')  
    set(handles.Vn,'String',' ');  
    Vn=0;  
end
```

En este ejemplo se ha utilizado la variable de la tensión nominal de primario (Vn). El programa lee y almacena la variable, pero en el caso de que esta contenga un carácter que no sea un número el programa genera la ventana mostrada en la figura

anterior, vacía el campo en el que se ha introducido el dato y almacena la variable con el valor "0" hasta que se introduzca de nuevo el dato en el formato correcto.

El segundo mensaje de error se activa cuando el dato introducido en el campo llamado "Factor de potencia" no es un valor aceptable. Como es sabido, el factor de potencia ($\cos(\varphi)$) debe comprender valores entre 0 y 1. En caso de no cumplir este requisito la ventana que se despliega es la siguiente:

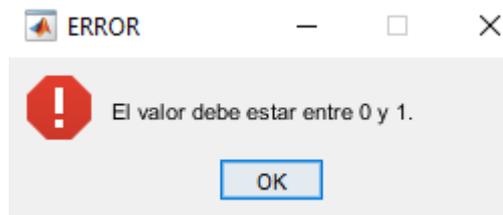


Figura 35. Error en el factor de potencia.

El código utilizado es el siguiente:

```
fdp=get(handles.fdpPrim,'String');
fdp=str2double(fdp);

if fdp<0
    errordlg('El valor debe estar entre 0 y 1.','ERROR')
    fdp=char(' ');
    set(handles.fdpPrim,'String',fdp);
end
if fdp>1
    errordlg('El valor debe estar entre 0 y 1.','ERROR')
    fdp=char(' ');
    set(handles.fdpPrim,'String',fdp);
end
```

En este ejemplo se ha utilizado la variable del factor de potencia (fdp) del primer tipo de problema, para el segundo tipo de problema el procedimiento es el mismo pero utilizando la variable correspondiente. El programa lee y almacena la variable, pero en el caso de que esta contenga un valor que no sea un número comprendido entre 0 y 1 el programa genera la ventana mostrada en la figura anterior y vacía el campo en el que se ha introducido el dato.

8. EJEMPLOS DE APLICACIÓN.

Para poner a prueba la interfaz gráfica desarrollada y demostrar su correcto funcionamiento se expondrán en este apartado dos ejemplos, uno por cada tipo de problema y transformador.

8.2. Ejemplo 1.

En el presente ejemplo se muestra la situación en la que un transformador monofásico se ensaya en vacío y cortocircuito para conocer sus parámetros internos y posteriormente se conecta una carga en bornes de secundario mientras se alimenta el transformador a su tensión nominal. El objetivo será determinar la tensión en bornes de la carga, el rendimiento y las pérdidas de potencia.

Un transformador monofásico con relación de transformación 13.200/230 V y una potencia nominal de 1.000 kVA se ensaya a vacío y cortocircuito por la rama de primario bajo las siguientes condiciones:

-Ensayo de cortocircuito:

-Tensión: 528 V.

-Intensidad: 75,7576 A.

-Pérdidas de cortocircuito: 4.000 W.

-Ensayo de vacío:

-Tensión: 13.200 V.

-Intensidad: 1,5151 A.

-Pérdidas de vacío: 2.500 W

Se conecta una carga en bornes de secundario que consume 800 kW con un factor de potencia 0,8 inductivo. Alimentando la rama primaria a tensión nominal, calcular la tensión en bornes de la carga, la intensidad de la rama de secundario, los coeficientes de caída de tensión, el rendimiento y las pérdidas de potencia.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Solución mediante interfaz gráfica:

En primer lugar se introducen los datos del transformador.



Datos del transformador

Tipo de transformador

Monofasico Trifasico

V1n: 13.2 kV

V2n: 0.23 kV

Sn: 1000 kVA

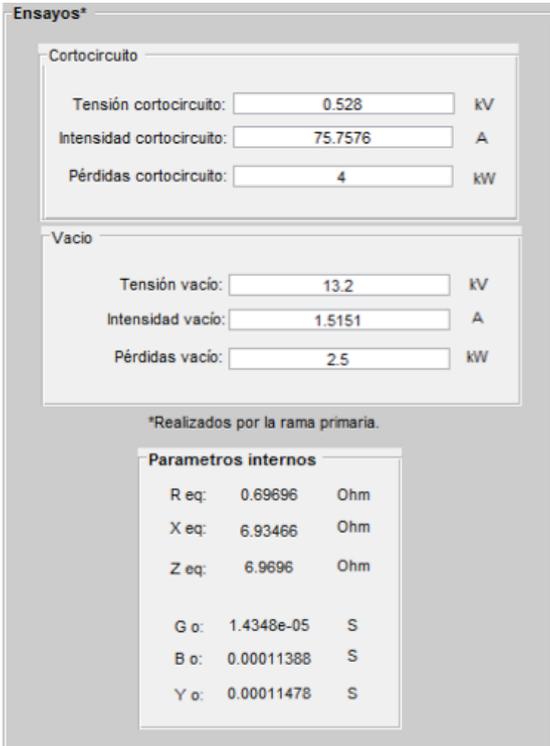
In: 75.7576 A

Relación de transformación:

57.3913

Figura 36. Datos del transformador I.

A continuación los datos de los ensayos realizados.



Ensayos*

Cortocircuito

Tensión cortocircuito: 0.528 kV

Intensidad cortocircuito: 75.7576 A

Pérdidas cortocircuito: 4 kW

Vacio

Tensión vacío: 13.2 kV

Intensidad vacío: 1.5151 A

Pérdidas vacío: 2.5 kW

*Realizados por la rama primaria.

Parametros internos

R eq:	0.69696	Ohm
X eq:	6.93466	Ohm
Z eq:	6.9696	Ohm
G o:	1.4348e-05	S
B o:	0.00011388	S
Y o:	0.00011478	S

Figura 37. Datos de los ensayos I.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Por último se selecciona el tipo de problema y se introducen los datos de las condiciones de carga. Pulsando el botón calcular se obtienen los resultados deseados.

Tipo de problema

Tension en bornes Tension primario

Cálculo de la tensión en bornes de la carga.

Tensión primario: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension secundario: 0.223672 kV

Intensidad secundario: 1.32002 A

Figura 38. Tipo de problema y condiciones de carga.

Rendimiento			Caídas de tensión.		
	99.194	%	Ur:	0.4	%
Máximo:	99.2156	%	Ux:	3.97995	%
			Uz:	4	%
Pérdidas					
Wo:	2500	W			
Wcc:	4000	W			
Pérdidas totales:	6500	W			

Figura 39. Resultados I.

Por tanto los resultados son los siguientes; la tensión en bornes de la carga es de 223,673 V y la intensidad que atraviesa dicha carga es de 1,32 A. Resultados

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

lógicos ya que al tratarse de una carga inductiva y estar alimentado por primario a tensión nominal, la tensión en bornes de la carga es menor que la nominal.

Los coeficientes de caída de tensión son:

$$-U_r = 0,4 \%$$

$$-U_x = 3,97995 \%$$

$$-U_z = 4 \%$$

De este modo funciona la máquina con un rendimiento de 99.194 % y generando unas pérdidas de potencia de 6,5 kW.

A continuación se muestra la imagen de la interfaz al completo y otras dos imágenes en las que se muestra el mismo problema pero alterando el carácter de la carga para mostrar el efecto que esto produce.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Datos del transformador

Tipo de transformador

Monofasico Trifasico

V1n: kV

V2n: kV

Sn: kVA

In: A

Relación de transformación:

Tipo de problema

Tension en bornes Tension primario

Cálculo de la tensión en bornes de la carga.

Tensión primario: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension secundario: kV

Intensidad secundario: A

Ensayos*

Cortocircuito

Tensión cortocircuito: kV

Intensidad cortocircuito: A

Pérdidas cortocircuito: kW

Vacio

Tensión vacío: kV

Intensidad vacío: A

Pérdidas vacío: kW

*Realizados por la rama primaria.

Parametros internos

R eq:	0.69696	Ohm
X eq:	6.93466	Ohm
Z eq:	6.9696	Ohm
G o:	1.4348e-05	S
B o:	0.00011388	S
Y o:	0.00011478	S

Cálculo de la tensión de primario.

Tensión en bornes de la carga: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension primario: kV

Intensidad primario: A

Rendimiento	Caídas de tensión.
99.194 %	Ur: 0.4 %
Máximo: 99.2156 %	Ux: 3.97995 %
	Uz: 4 %

Pérdidas

Wo:	2500	W
Wcc:	4000	W
Pérdidas totales:	6500	W

Calcular

Figura 40. Vista general I.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Datos del transformador

Tipo de transformador

Monofasico Trifasico

V1n: kV

V2n: kV

Sn: kVA

In: A

Relación de transformación:

Tipo de problema

Tension en bornes Tension primario

Cálculo de la tensión en bornes de la carga.

Tensión primario: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension secundario: kV

Intensidad secundario: A

Ensayos*

Cortocircuito

Tensión cortocircuito: kV

Intensidad cortocircuito: A

Pérdidas cortocircuito: kW

Vacio

Tensión vacío: kV

Intensidad vacío: A

Pérdidas vacío: kW

*Realizados por la rama primaria.

Parametros internos

R eq:	0.69696	Ohm
X eq:	6.93466	Ohm
Z eq:	6.9696	Ohm
G o:	1.4348e-05	S
B o:	0.00011388	S
Y o:	0.00011478	S

Cálculo de la tensión de primario.

Tensión en bornes de la carga: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension primario: kV

Intensidad primario: A

Rendimiento

%

Máximo: %

Caídas de tensión.

Ur:	<input type="text" value="0.4"/> %
Ux:	<input type="text" value="3.97995"/> %
Uz:	<input type="text" value="4"/> %

Pérdidas

Wo:	<input type="text" value="2500"/> W
Wcc:	<input type="text" value="4000"/> W
Pérdidas totales:	<input type="text" value="6500"/> W

Calcular

Figura 41. Vista general II.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Datos del transformador

Tipo de transformador

Monofasico Trifasico

V1n: kV

V2n: kV

Sn: kVA

In: A

Relación de transformación:

Tipo de problema

Tension en bornes Tension primario

Ensayos*

Cortocircuito

Tensión cortocircuito: kV

Intensidad cortocircuito: A

Pérdidas cortocircuito: kW

Vacio

Tensión vacío: kV

Intensidad vacío: A

Pérdidas vacío: kW

*Realizados por la rama primaria.

Parámetros internos

R eq:	0.69696	Ohm
X eq:	6.93466	Ohm
Z eq:	6.9696	Ohm
G o:	1.4348e-05	S
B o:	0.00011388	S
Y o:	0.00011478	S

Cálculo de la tensión en bornes de la carga.

Tensión primario: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension secundario: kV

Intensidad secundario: A

Cálculo de la tensión de primario.

Tensión en bornes de la carga: kV

Carga: kW

Factor de potencia:

Tipo de carga.

Capacitiva Inductiva Resistiva

Tension primario: kV

Intensidad primario: A

Rendimiento

%

Máximo: %

Caídas de tensión.

Ur:	<input type="text" value="0.4"/> %
Ux:	<input type="text" value="3.97995"/> %
Uz:	<input type="text" value="4"/> %

Pérdidas

Wo:	<input type="text" value="2500"/> W
Wcc:	<input type="text" value="2560"/> W
Pérdidas totales:	<input type="text" value="5060"/> W

Calcular

Figura 42. Vista general III.

8.2. Ejemplo 2.

En este caso se presenta un transformador trifásico que es ensayado del mismo modo que el anterior para conocer sus parámetros internos. La diferencia radica en que en este caso se impone la condición de la caída de tensión en bornes de la carga para así conocer cual debe ser la tensión a la que se debe alimentar la máquina. Del mismo modo se muestra el rendimiento y las pérdidas de potencia.

Un transformador trifásico con relación de transformación 220.000/20.000 V y una potencia nominal de 120.000 kVA se ensaya a vacío y cortocircuito por la rama de primario bajo las siguientes condiciones:

-Ensayo de cortocircuito:

-Tensión: 28.600 V.

-Intensidad: 314,918 A.

-Pérdidas de cortocircuito: 120 kW.

-Ensayo de vacío:

-Tensión: 220 kV.

-Intensidad: 6,29836 A.

-Pérdidas de vacío: 39.690 W

Se conecta una carga en bornes de secundario que consume 80.000 kW con un factor de potencia 0,8 inductivo. Alimentando la carga a tensión nominal, calcular la tensión en bornes de primario, la intensidad de la rama de primario, los coeficientes de caída de tensión, el rendimiento y las pérdidas de potencia.

Solución mediante interfaz:

En primer lugar, al igual que en el ejemplo anterior, se introducen los datos nominales del transformador y se selecciona la opción de tipo de transformador adecuada. La interfaz muestra automáticamente los resultados de intensidad nominal y relación de transformación.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

The image shows a software window titled "Datos del transformador". Inside, there is a sub-section titled "Tipo de transformador" with two radio buttons: "Monofasico" (unselected) and "Trifasico" (selected). Below this, there are four input fields with their respective units: "U1n" with value "220" and unit "kV", "U2n" with value "20" and unit "kV", "Sn:" with value "120000" and unit "KVA", and "In:" with value "314.918" and unit "A". At the bottom, the text "Relación de transformación:" is followed by the value "11".

Figura 43. Datos del transformador II.

A continuación se introducen los datos de los ensayos para obtener los parámetros internos del transformador.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Ensayos*

Cortocircuito

Tensión cortocircuito: kV

Intensidad cortocircuito: A

Pérdidas cortocircuito: kW

Vacio

Tensión vacío: kV

Intensidad vacío: A

Pérdidas vacío: kW

*Realizados por la rama primaria.

Parámetros internos

R eq:	0.403334	Ohm
X eq:	52.4318	Ohm
Z eq:	52.4334	Ohm
G o:	8.20041e-07	S
B o:	4.95799e-05	S
Y o:	4.95867e-05	S

Figura 44. Datos de los ensayos II.

Por último se selecciona el tipo de problema y se introducen las condiciones de carga. Pulsando el botón “Calcular” se obtienen los resultados.

Tipo de problema

Tension en bornes Tension primario

Figura 45. Seleccionar el tipo de problema.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

Cálculo de la tensión de primario.		
Tensión en bornes de la carga:	20	kV
Carga:	80000	kW
Factor de potencia:	0.8	
Tipo de carga.		
<input type="radio"/> Capacitiva	<input checked="" type="radio"/> Inductiva	<input type="radio"/> Resistiva
Tension primario:	235.211	kV
Intensidad primario:	262.432	A
Rendimiento		Caídas de tensión.
	99.8465 %	Ur: 0.1 %
Máximo:	99.8564 %	Ux: 12.9996 %
		Uz: 13 %
Pérdidas		
Wo:	13230	W
Wcc:	27777.8	W
Pérdidas totales:	41007.8	W

Figura 46. Condiciones de carga y resultados.

Por tanto los resultados son los siguientes; la tensión en bornes de primario ha de ser 235,211 kV y la intensidad que circula por dicha rama es de 262,432 A.

Los coeficientes de caída de tensión son:

-Ur= 0,1 %

-Ux= 12,9996 %

-Uz= 13 %

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

De este modo funciona la máquina con un rendimiento de 99.8465 % y generando unas pérdidas de potencia de 41,0078 kW.

A continuación se muestra la imagen general de la interfaz al completo.

The image shows a software interface for transformer calculations, divided into several sections:

- Datos del transformador:**
 - Tipo de transformador: Monofasico Trifasico
 - U1n: 220 kV
 - U2n: 20 kV
 - Sn: 120000 kVA
 - In: 314.918 A
 - Relación de transformación: 11
- Ensayos*:**
 - Cortocircuito:**
 - Tensión cortocircuito: 28.6 kV
 - Intensidad cortocircuito: 314.918 A
 - Pérdidas cortocircuito: 120 kW
 - Vacio:**
 - Tensión vacío: 220 kV
 - Intensidad vacío: 6.29836 A
 - Pérdidas vacío: 39.690 kW
- Parametros internos:**
 - R eq: 0.403334 Ohm
 - X eq: 52.4318 Ohm
 - Z eq: 52.4334 Ohm
 - G o: 8.20041e-07 S
 - B o: 4.95799e-05 S
 - Y o: 4.95867e-05 S
- Tipo de problema:**
 - Tension en bornes Tension primario
- Cálculo de la tensión en bornes de la carga:**
 - Tensión primario: [input] kV
 - Carga: [input] kW
 - Factor de potencia: [input] 1
 - Tipo de carga: Capacitiva Inductiva Resistiva
 - Tension secundario: [input] kV
 - Intensidad secundario: [input] A
- Cálculo de la tensión de primario:**
 - Tensión en bornes de la carga: 20 kV
 - Carga: 80000 kW
 - Factor de potencia: 0.8
 - Tipo de carga: Capacitiva Inductiva Resistiva
 - Tension primario: 235.211 kV
 - Intensidad primario: 262.432 A
- Rendimiento:**
 - 99.8465 %
 - Máximo: 99.8564 %
- Caídas de tensión:**
 - Ur: 0.1 %
 - Ux: 12.9996 %
 - Uz: 13 %
- Pérdidas:**
 - Wo: 13230 W
 - Wcc: 27777.8 W
 - Pérdidas totales: 41007.8 W
- Calcular:**
 - Buttons: **Calcular** (blue), **Reset** (yellow)

Figura 47. Vista general IV.

9. PLANIFICACIÓN Y DESCRIPCIÓN DE TAREAS.

En este apartado se muestran las tareas realizadas para llevar a cabo el proyecto. Se nombrarán y explicarán las tareas y el momento en el que fueron realizadas. Las personas que han trabajado en este TFG son un ingeniero superior como director del TFG y un alumno.

A continuación se realizará una breve explicación de las tareas llevadas a cabo:

-Análisis del software: esta tarea consiste en la primera toma de contacto con la interfaz gráfica "GUIDE" de "MATLAB", conocer su funcionamiento mediante la lectura de diferentes guías y mediante la práctica realizando sencillos ejercicios para poner a prueba los conocimientos.

-Recopilación de la información: consiste en la investigación de todos los fundamentos teóricos sobre los que se apoya la resolución de los ejercicios a resolver mediante el uso de la aplicación a programar. Para el desarrollo de la interfaz será necesario programar una serie de fórmulas matemáticas obtenidas mediante dichos conocimientos de la teoría acerca de los transformadores de potencia.

-Diseño de la interfaz: en esta tarea se realiza de forma esquemática el diseño de la interfaz gráfica. El fin de esta tarea es definir y organizar los diferentes módulos que componen la interfaz para su posterior programación.

-Desarrollo de la interfaz: en este punto comienza la programación de la interfaz gráfica, que es la principal tarea de este TFG. La programación se realiza en función a el diseño realizado en la tarea anterior. En dicha tarea se aplican tanto los conocimientos teóricos obtenidos como los del uso de la aplicación "GUIDE".

-Validación de la interfaz: tras el desarrollo de la interfaz y una reunión con el director de este proyecto se analiza la aplicación para conocer sus fallos y la implantación de posibles mejoras. Una vez realizado este análisis se llevan a cabo las correcciones necesarias sobre la interfaz.

-Elaboración del documento: se redacta el documento o informe a entregar junto a la aplicación en el que está recogido de principio a fin todos los aspectos que componen el TFG, como por ejemplo: su contexto, sus fundamentos teóricos, sus beneficios, etc. Tras el desarrollo del documento y una reunión con el director de este proyecto se analiza el informe para conocer sus fallos y la implantación de posibles mejoras. Una vez realizado este análisis se llevan a cabo las correcciones necesarias sobre el documento.

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

En la última reunión se da por concluido todo el TFG, y el trabajo estaría terminado.

A continuación se muestra una tabla y un diagrama “Gantt” en el que se incluyen todas las tareas y de forma detallada cuando fueron realizadas.

Referencia	Descripción	Fecha inicio	Fecha final
H.0	Inicio del proyecto	6/11/18	6/11/18
R.1	Primera reunión	6/11/18	6/11/18
T.1	Análisis del software	13/11/18	12/12/18
H.1	Fin del análisis	13/12/18	13/12/18
R.2	Segunda reunión	17/12/18	17/12/18
T.2	Recopilación de la información	18/12/18	22/1/19
H.2	Fin de la recopilación de información	23/1/19	23/1/19
R.3	Tercera reunión	24/1/19	24/1/19
T.3	Diseño de la interfaz	28/1/19	7/2/19
R.4	Cuarta reunión	8/2/19	8/2/19
T.4	Desarrollo de la interfaz	11/2/19	28/3/19
R.5	Quinta reunión	29/3/19	29/3/19
T.5	Validación de la interfaz	1/4/19	25/4/19
H.3	Fin de la interfaz	29/4/19	29/4/19
R.6	Sexta reunión	29/4/19	29/4/19
T.6	Elaboración del documento	28/1/19	17/6/19
R.7	Última reunión	25/6/19	25/6/19
H.4	Fin del proyecto	25/6/19	25/6/19

Tabla 3. Tareas que componen el proyecto.

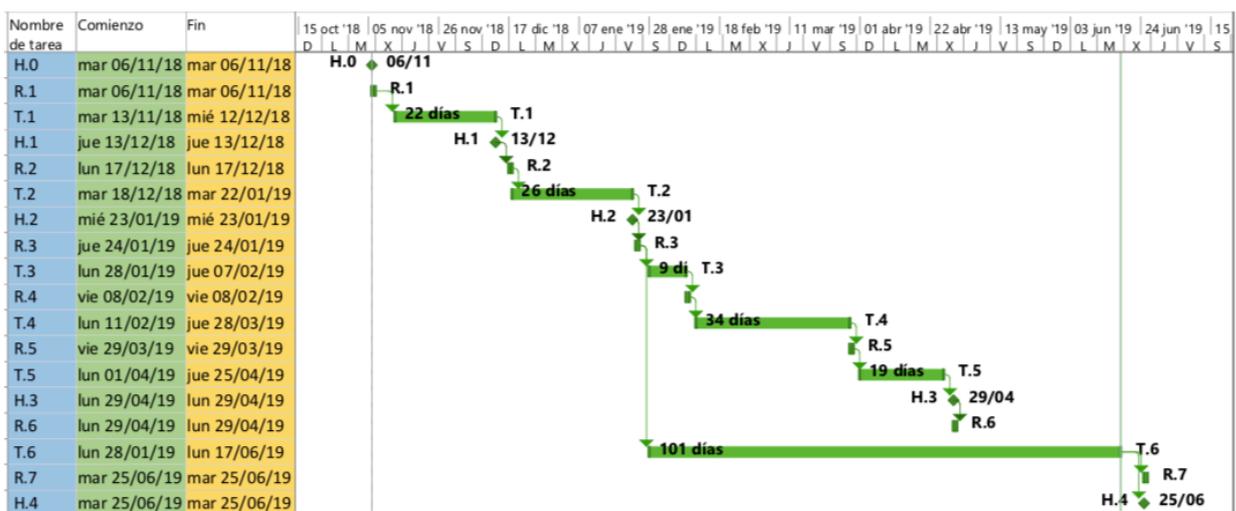


Figura 48. Diagrama Gantt.

10. PRESUPUESTO.

Este presupuesto se engloba en tres partidas distintas; la primera son las horas internas, donde se gestiona el presupuesto referente a los recursos humanos empleados para realizar el proyecto. La segunda partida la componen las amortizaciones, donde se recoge el presupuesto de los equipos y licencias utilizadas. Por último están anotados los gastos en materiales empleados en el TFG.

HORAS INTERNAS			
Concepto	Nº de horas	€/hora	Total
Ingeniero superior	45h	60€/hora	2.700 €
Ingeniero graduado	180h	25€/hora	4.500 €
SUBTOTAL			7.200 €

Tabla 4. Horas internas.

AMORTIZACIONES				
Concepto	Precio producto	Vida útil (horas)	Utilización (horas)	Total
Licencia MATLAB	70 €	1500h	70h	3,27 €
Licencia Office	100 €	2500h	60h	2,40 €
Ordenador	1.150 €	30000h	150h	5,75 €
SUBTOTAL				11,42 €

Tabla 5. Amortizaciones.

GASTOS	
Concepto	Total
Material	40 €
SUBTOTAL	40 €

Tabla 6. Gastos.

RESUMEN	
Horas internas	7.200 €
Amortizaciones	11,42 €
Gastos	40 €
SUBTOTAL	7.251,42 €
Costes indirectos (10%)	725,15 €
TOTAL	7976.57 €

Tabla 7. Resumen.

Por último se recogen todos los gastos en una tabla resumen y se tiene en cuenta un 10% del subtotal de los gastos para costes indirectos.

11. CONCLUSIONES.

Conocer el funcionamiento de los transformadores es fundamental para el correcto funcionamiento de un sistema eléctrico, por lo tanto será necesario conocer sus parámetros fundamentales como las caídas de tensión, las pérdidas de potencia y su rendimiento. El funcionamiento correcto de los transformadores asegura un suministro de energía eléctrica de calidad y fiable.

Los dispositivos o cargas conectadas a la red están diseñadas para trabajar bajo unas condiciones determinadas; no garantizar estas condiciones puede poner en riesgo a los dispositivos conectados y provocar averías. Por todo esto, es necesario que los transformadores seleccionados para abastecer las distintas áreas de la red sean los adecuados. Para ello debemos manejar las herramientas necesarias que nos proporcionen los datos anteriormente mencionados.

Disponer de una herramienta que permita calcular las caídas de tensión, pérdidas de potencia y rendimiento de un transformador sin la necesidad de realizar ensayos de campo o modelos a escala proporciona una considerable ventaja para todas las personas implicadas en trabajar con estas máquinas eléctricas. Las ventajas no solo se limitan a un ahorro económico sino que también conllevan un ahorro de tiempo y de recursos humanos.

En este Trabajo Fin de Grado se ha desarrollado una herramienta software cuya función principal es calcular como ya se ha dicho las caídas de tensión, pérdidas de potencia y rendimiento de un transformador, tanto monofásico como trifásico. También aporta una serie de parámetros internos de la máquina que son de interés para el estudio de la misma.

12. REFERENCIAS.

- [1] José Félix Miñambres, y otros. Guía de Autoaprendizaje de Máquinas Eléctricas. Madrid : Pearson Educación S.A, 2008. 978-84-8322-490-8.
- [2] A.L. Orille, "Centrales Eléctricas I, II y III", Ediciones UPC, 1993.
- [3] Máquinas eléctricas. Jesús Fraile Mora. Mc Graw Hill.
- [4] Feito, Javier Sanz. Máquinas Eléctricas. Madrid : Pearson Educación S.A, 2002. 84-205-3391.
- [5] Fitzgerald, A. E., Jr, Charles Kingsley y Umans, Stephen D. Máquinas Eléctricas. México D.F. : McGraw-Hill/ Interamericana Editores, S.A, 2004. 970-10-4052-X.
- [6] Navarro, Rafael Sanjurjo. Máquinas Eléctricas. Madrid : García-Maroto Editores S.L, 2011. 978-84-15214-14-4.
- [7] Chapman, Stephen J. Máquinas Eléctricas. Bogotá : McGraw-Hill Interamericana S.A, 2000. 958-41-0056-4.
- [8] Wildi, Theodore. Electrical Machines, Drives, And Power Systems. Upper Saddle River : Pearson Education International, 2006. 0-13-196918-8.
- [9] Iribarnegaray, Luis Serrano y Román, Javier Martínez. Máquinas Eléctricas. s.l. : Editorial Universitat Politècnica de València, 2013. 978-84-9048-165-3.
- [10] Ras, Enrique. Transformadores de Potencia, de Medida y de Protección. Barcelona : Marcombo Boixareu Editores, 1994. 84-267-0690-8.
- [11] Belver, Ma Inmaculada Zamora, y otros. Simulación de Sistemas Eléctricos. Madrid : Pearson Educación S.A, 2005. 84-205-4808-1.

13. ANEXO. CÓDIGO DE LA INTERFAZ.

A continuación se adjunta el código utilizado en la interfaz.

```
function varargout = GUIDE(varargin)
% GUIDE MATLAB code for GUIDE.fig
%     GUIDE, by itself, creates a new GUIDE or raises the
existing
%     singleton*.
%
%     H = GUIDE returns the handle to a new GUIDE or the handle
to
%     the existing singleton*.
%
%     GUIDE('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in GUIDE.M with the given input
arguments.
%
%     GUIDE('Property','Value',...) creates a new GUIDE or
raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before GUIDE_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to GUIDE_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI
allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUIDE

% Last Modified by GUIDE v2.5 17-Jun-2019 13:12:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @GUIDE_OpeningFcn, ...
                  'gui_OutputFcn',  @GUIDE_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% End initialization code - DO NOT EDIT

% --- Executes just before GUIDE is made visible.
function GUIDE_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUIDE (see VARARGIN)

% Choose default command line output for GUIDE
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUIDE wait for user response (see UIRESUME)
% uiwait(handles.figure1);

tipo=get(handles.radiobutton14,'String');
switch tipo
    case 'Tension en bornes'
        set(handles.tensionbornescarga,'Enable','Off');
        set(handles.cargasec,'Enable','Off');
        set(handles.fdpsec,'Enable','Off');
        set(handles.TensionDato,'Enable','On');
        set(handles.CargaPrim,'Enable','On');
        set(handles.fdpPrim,'Enable','On');

        set(handles.radiobutton10,'Enable','Off');
        set(handles.radiobutton11,'Enable','Off');
        set(handles.radiobutton17,'Enable','Off');
        set(handles.radiobutton7,'Enable','On');
        set(handles.radiobutton9,'Enable','On');
        set(handles.radiobutton16,'Enable','On');

    case 'Tension primario'
        set(handles.tensionbornescarga,'Enable','On');
        set(handles.cargasec,'Enable','On');
        set(handles.fdpsec,'Enable','On');
        set(handles.TensionDato,'Enable','Off');
        set(handles.CargaPrim,'Enable','Off');
        set(handles.fdpPrim,'Enable','Off');

        set(handles.radiobutton10,'Enable','On');
        set(handles.radiobutton11,'Enable','On');
        set(handles.radiobutton17,'Enable','On');
        set(handles.radiobutton7,'Enable','Off');
        set(handles.radiobutton9,'Enable','Off');
        set(handles.radiobutton16,'Enable','Off');

end
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Outputs from this function are returned to the command
line.
function varargout = GUIDE_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see
VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject      handle to checkbox1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox1

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
% hObject      handle to checkbox2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox2

function Vn_Callback(hObject, eventdata, handles)
% hObject      handle to Vn (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Vn as text
%        str2double(get(hObject,'String')) returns contents of Vn
as a double
Vn=get(hObject,'String');
Vn=str2double(Vn);
if isnan(Vn)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Vn,'String',' ');
    Vn=0;
end
if inf==2
    Vn=Vn/sqrt(3);
end
handles.edit=Vn;
guidata(hObject,handles);
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Executes during object creation, after setting all
properties.
function Vn_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Vn (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function editSnMono_Callback(hObject, eventdata, handles)
% hObject    handle to editSnMono (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editSnMono as
text
%         str2double(get(hObject,'String')) returns contents of
editSnMono as a double
SnMono=get(hObject,'String');
SnMono=str2double(SnMono);
if isnan(SnMono)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',' ');
    SnMono=0;
end
tension=str2double(get(handles.Vn,'String'));
if get(handles.radiobutton3,'Value')==1.0
    in=SnMono/tension;
    set(handles.text63,'String',in);
else
    tension=tension/sqrt(3);
    SnMono=SnMono/3;
    in=SnMono/tension;
    set(handles.text63,'String',in);
end

% --- Executes during object creation, after setting all
properties.
function editSnMono_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editSnMono (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function SnTri_Callback(hObject, eventdata, handles)
% hObject    handle to SnTri (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SnTri as text
%         str2double(get(hObject,'String')) returns contents of
SnTri as a double
SnTri=get(hObject,'String');
SnTri=str2double(SnTri);
if isnan(SnTri)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    SnTri=0;
end
SnTri=SnTri/sqrt(3);
handles.edit=SnTri;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function SnTri_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SnTri (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Un_Callback(hObject, eventdata, handles)
% hObject    handle to Un (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Un as text
%         str2double(get(hObject,'String')) returns contents of Un
as a double
Un=get(hObject,'String');
Un=str2double(Un);
if isnan(Un)
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```

        errordlg('El valor debe ser numérico.','ERROR')
        set(handles.editSnMono,'String',0);
        Un=0;
    end
    Un=Un/sqrt(3);
    handles.edit=Un;
    guidata (hObject,handles);

    % --- Executes during object creation, after setting all
    properties.
    function Un_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to Un (see GCBO)
    % eventdata  reserved - to be defined in a future version of
    MATLAB
    % handles    empty - handles not created until after all
    CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %         See ISPC and COMPUTER.
    if ispc      &&      isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

    function Vcc_Callback(hObject, eventdata, handles)
    % hObject    handle to Vcc (see GCBO)
    % eventdata  reserved - to be defined in a future version of
    MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of Vcc as text
    %         str2double(get(hObject,'String')) returns contents of
    Vcc as a double
    Vcc=get(hObject,'String');
    Vcc=str2double(Vcc);
    if isnan(Vcc)
        errordlg('El valor debe ser numérico.','ERROR')
        set(handles.editSnMono,'String',0);
        Vcc=0;
    end
    if inf==2
        Vcc=Vcc/sqrt(3);
    end
    handles.edit=Vcc;
    guidata (hObject,handles);

    % --- Executes during object creation, after setting all
    properties.
    function Vcc_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to Vcc (see GCBO)
    % eventdata  reserved - to be defined in a future version of
    MATLAB
    % handles    empty - handles not created until after all
    CreateFcns called

    % Hint: edit controls usually have a white background on Windows.

```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```

%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Icc_Callback(hObject, eventdata, handles)
% hObject      handle to Icc (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Icc as text
%          str2double(get(hObject,'String')) returns contents of
Icc as a double
Icc=get(hObject,'String');
Icc=str2double(Icc);
if isnan(Icc)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    Icc=0;
end
handles.edit=Icc;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function Icc_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Icc (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Wcc_Callback(hObject, eventdata, handles)
% hObject      handle to Wcc (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Wcc as text
%          str2double(get(hObject,'String')) returns contents of
Wcc as a double
Wcc=get(hObject,'String');
Wcc=str2double(Wcc);
if isnan(Wcc)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    Wcc=0;
end

```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
end
if inf==2
    Wcc=Wcc/3;
end
handles.edit=Wcc;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function Wcc_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Wcc (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Vo_Callback(hObject, eventdata, handles)
% hObject    handle to Vo (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Vo as text
%       str2double(get(hObject,'String')) returns contents of Vo
as a double
Vo=get(hObject,'String');
Vo=str2double(Vo);
if isnan(Vo)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Vo,'String',' ');
    Vo=0;
end
if inf==2
    Vo=Vo/sqrt(3);
end
handles.edit=Vo;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function Vo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Vo (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
        set(hObject, 'BackgroundColor', 'white');
    end

    function Io_Callback(hObject, eventdata, handles)
    % hObject    handle to Io (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: get(hObject, 'String') returns contents of Io as text
    %          str2double(get(hObject, 'String')) returns contents of Io
as a double
    Io=get(hObject, 'String');
    Io=str2double(Io);
    if isnan(Io)
        errordlg('El valor debe ser numérico.', 'ERROR')
        set(handles.Io, 'String', ' ');
        Io=0;
    end
    handles.edit=Io;
    guidata (hObject,handles);

    % --- Executes during object creation, after setting all
properties.
    function Io_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to Io (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    empty - handles not created until after all
CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %          See ISPC and COMPUTER.
    if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end

    function Wo_Callback(hObject, eventdata, handles)
    % hObject    handle to Wo (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: get(hObject, 'String') returns contents of Wo as text
    %          str2double(get(hObject, 'String')) returns contents of Wo
as a double
    if get(handles.radiobutton3, 'Value')==1.0
        A1=1;
    else
        A1=3;
    end
    Vo=get(handles.Vo, 'String');
    Vo=str2double(Vo)*1000;
    if isnan(Vo)
        errordlg('El valor debe ser numérico.', 'ERROR')
        set(handles.Vo, 'String', ' ');
        Vo=0;
    end
end
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```

end
if A1==3
    Vo=Vo/sqrt(3);
end

Io=get(handles.Io,'String');
Io=str2double(Io);
if isnan(Io)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Io,'String',' ');
    Io=0;
end

Wo=get(hObject,'String');
Wo=str2double(Wo)*1000;
if isnan(Wo)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Wo,'String',' ');
    Wo=0;
end
if A1==3
    Wo=Wo/3;
end

go=(Wo/(Vo)^2);
yo=Io/Vo;
bo=sqrt(yo^2-go^2);

set(handles.go,'string',go);

set(handles.yo,'string',yo);

set(handles.bo,'string',bo);

% --- Executes during object creation, after setting all
properties.
function Wo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Wo (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in checkbox3.
function checkbox3_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% Hint: get(hObject,'Value') returns toggle state of checkbox3

% --- Executes on button press in checkbox4.
function checkbox4_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox4

function fdpPrim_Callback(hObject, eventdata, handles)
% hObject    handle to fdpPrim (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fdpPrim as
text
%          str2double(get(hObject,'String')) returns contents of
fdpPrim as a double
fdp=get(hObject,'String');
fdp=str2double(fdp);
if isnan(fdp)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    fdp=0;
end

handles.edit=fdp;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function fdpPrim_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fdpPrim (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function CargaPrim_Callback(hObject, eventdata, handles)
% hObject    handle to CargaPrim (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% Hints: get(hObject,'String') returns contents of CargaPrim as
text
%           str2double(get(hObject,'String')) returns contents of
CargaPrim as a double
CargaDato=get(hObject,'String');
CargaDato=str2double(CargaDato);
if isnan(CargaDato)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    CargaDato=0;
end
if inf==2
    CargaDato=CargaDato/3;
end
handles.edit=CargaDato;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function CargaPrim_CreateFcn(hObject, eventdata, handles)
% hObject    handle to CargaPrim (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc      &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function TensionDato_Callback(hObject, eventdata, handles)
% hObject    handle to TensionDato (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TensionDato as
text
%           str2double(get(hObject,'String')) returns contents of
TensionDato as a double
TensionDato=get(hObject,'String');
TensionDato=str2double(TensionDato);
if isnan(TensionDato)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    TensionDato=0;
end
if inf==2
    TensionDato=TensionDato/sqrt(3);
end
handles.edit=TensionDato;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
function TensionDato_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TensionDato (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fdpSec_Callback(hObject, eventdata, handles)
% hObject    handle to fdpSec (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fdpSec as text
%         str2double(get(hObject,'String')) returns contents of
fdpSec as a double
fdp=get(hObject,'String');
fdp=str2double(fdp);
if isnan(fdp)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    fdp=0;
end
handles.edit=fdp;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function fdpSec_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fdpSec (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function CargaSec_Callback(hObject, eventdata, handles)
% hObject    handle to CargaSec (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of CargaSec as
text
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
%          str2double(get(hObject,'String')) returns contents of
CargaSec as a double

%--- Executes during object creation, after setting all
properties.
function CargaSec_CreateFcn(hObject, eventdata, handles)
% hObject    handle to CargaSec (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%         str2double(get(hObject,'String')) returns contents of
edit16 as a double

% --- Executes during object creation, after setting all
properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%         str2double(get(hObject,'String')) returns contents of
edit17 as a double
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Executes during object creation, after setting all
properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%         str2double(get(hObject,'String')) returns contents of
edit19 as a double

% --- Executes during object creation, after setting all
properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
%         str2double(get(hObject,'String')) returns contents of
edit20 as a double

% --- Executes during object creation, after setting all
properties.
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit21_Callback(hObject, eventdata, handles)
% hObject    handle to edit21 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit21 as text
%         str2double(get(hObject,'String')) returns contents of
edit21 as a double

% --- Executes during object creation, after setting all
properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit21 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit22_Callback(hObject, eventdata, handles)
% hObject    handle to edit22 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit22 as text
%         str2double(get(hObject,'String')) returns contents of
edit22 as a double

% --- Executes during object creation, after setting all
properties.
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
function edit22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit22 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit23_Callback(hObject, eventdata, handles)
% hObject    handle to edit23 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit23 as text
%         str2double(get(hObject,'String')) returns contents of
edit23 as a double

% --- Executes during object creation, after setting all
properties.
function edit23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit23 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit24_Callback(hObject, eventdata, handles)
% hObject    handle to edit24 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit24 as text
%         str2double(get(hObject,'String')) returns contents of
edit24 as a double
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Executes during object creation, after setting all
properties.
function edit24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit24 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Calcular.
function Calcular_Callback(hObject, eventdata, handles)
% hObject    handle to Calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

%A=get(handles.tipotrafo,'String');
%B=get(handles.tipocargaB,'String');
%C=get(handles.tipocargaC,'String');
%tipo=get(handles.uibuttongroup9,'String');

if get(handles.radiobutton3,'Value')==1
    A1=1;
else
    A1=3;
end

if get(handles.radiobutton14,'Value')==1
    tipol=1;
else
    tipol=2;
end

if get(handles.radiobutton16,'Value')==1
    B1=1;
elseif get(handles.radiobutton7,'Value')==1
    B1=2;
else
    B1=3;
end

if get(handles.radiobutton17,'Value')==1
    C1=1;
elseif get(handles.radiobutton10,'Value')==1
    C1=2;
else
    C1=3;
end
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
Vn=get(handles.Vn,'String');
Vn=str2double(Vn)*1000;
if isnan(Vn)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Vn,'String',' ');
    Vn=0;
end
if A1==3
    Vn=Vn/sqrt(3);
end

V2n=get(handles.V2n,'String');
V2n=str2double(V2n)*1000;
if isnan(V2n)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.V2n,'String',' ');
    V2n=0;
end
if A1==3
    V2n=V2n/sqrt(3);
end

SnMono=get(handles.editSnMono,'String');
SnMono=str2double(SnMono)*1000;
if isnan(SnMono)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',' ');
    SnMono=0;
end
if A1==3
    SnMono=SnMono/3;
end

Vcc=get(handles.edit30,'String');
Vcc=str2double(Vcc)*1000;
if isnan(Vcc)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Vcc,'String',' ');
    Vcc=0;
end
if A1==3
    Vcc=Vcc/sqrt(3);
end

Icc=get(handles.edit31,'String');
Icc=str2double(Icc);
if isnan(Icc)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Icc,'String',' ');
    Icc=0;
end
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
Wcc=get(handles.edit32,'String');
Wcc=str2double(Wcc)*1000;
if isnan(Wcc)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Wcc,'String',' ');
    Wcc=0;
end
if A1==3
    Wcc=Wcc/3;
end

Vo=get(handles.Vo,'String');
Vo=str2double(Vo)*1000;
if isnan(Vo)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Vo,'String',' ');
    Vo=0;
end
if A1==3
    Vo=Vo/sqrt(3);
end

Io=get(handles.Io,'String');
Io=str2double(Io);
if isnan(Io)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Io,'String',' ');
    Io=0;
end

Wo=get(handles.Wo,'String');
Wo=str2double(Wo)*1000;
if isnan(Wo)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.Wo,'String',' ');
    Wo=0;
end
if A1==3
    Wo=Wo/3;
end

%fdp=get(handles.fdpPrim,'String');
%fdp=str2double(fdp);

CargaDato=get(handles.CargaPrim,'String');
CargaDato=str2double(CargaDato)*1000;

if A1==3
    CargaDato=CargaDato/3;
end

TensionDato=get(handles.TensionDato,'String');
TensionDato=str2double(TensionDato)*1000;
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
if A1==3
    TensionDato=TensionDato/sqrt(3);
end

tensionbornescarga=get(handles.tensionbornescarga,'String');
tensionbornescarga=str2double(tensionbornescarga)*1000;

if A1==3
    tensionbornescarga=tensionbornescarga/sqrt(3);
end

cargasec=get(handles.cargasec,'String');
cargasec=str2double(cargasec)*1000;

if A1==3
    cargasec=cargasec/3;
end

%fdpsec=get(handles.fdpsec,'String');
%fdpsec=str2double(fdpsec);

rt=Vn/V2n;

if B1==1
    fdp=1;
    set(handles.fdpPrim,'String',fdp);
else
    fdp=get(handles.fdpPrim,'String');
    fdp=str2double(fdp);
end

if fdp<0
    errordlg('El valor debe estar entre 0 y 1.','ERROR')
    fdp=char(' ');
    set(handles.fdpPrim,'String',fdp);
end
if fdp>1
    errordlg('El valor debe estar entre 0 y 1.','ERROR')
    fdp=char(' ');
    set(handles.fdpPrim,'String',fdp);
end

if C1==1
    fdpsec=1;
    set(handles.fdpsec,'String',fdpsec);
else
    fdpsec=get(handles.fdpsec,'String');
    fdpsec=str2double(fdpsec);
end

if fdpsec<0
    errordlg('El valor debe estar entre 0 y 1.','ERROR')
    fdpsec=char(' ');
    set(handles.fdpsec,'String',fdpsec);
end
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```

if fdp>1
    errordlg('El valor debe estar entre 0 y 1.', 'ERROR')
    fdpsec=char(' ');
    set(handles.fdpsec, 'String', fdpsec);
end

if tipol==1

    int=CargaDato/(TensionDato*fdp);
    In=SnMono/Vn;
    req=Wcc/Icc^2;
    zeq=Vcc/Icc;
    xeq=sqrt(zeq^2-req^2);
    fi=acos(fdp);
    ifactcarga=int/In;
    ux=(xeq*In)/Vn;
    ur=(req*In)/Vn;
    uz=(zeq*In)/Vn;
    tensionnominalrt=TensionDato/rt;
    if B1==3

u=ifactcarga*(ur*100*fdp+ux*100*sin(fi))+((ifactcarga^2)/200)*(ux*100*
fdp-ur*100*sin(fi))^2;
        elseif B1==1

u=ifactcarga*(ur*100*fdp)+((ifactcarga^2)/200)*(ux*100*fdp)^2;
        else

            u=ifactcarga*(ur*100*fdp-
ux*100*sin(fi))+((ifactcarga^2)/200)*(ux*100*fdp+ur*100*sin(fi))^2;
        end

        tensioncalculada=tensionnominalrt*(1-(u/100));

        if A1==3
            tensioncalculada=tensioncalculada*sqrt(3);
        end
        rdto=100*((ifactcarga*SnMono*fdp)/(ifactcarga*SnMono*fdp+(Wo*(Te
nsionDato)^2)/(Vo)^2)+((ifactcarga)^2)*(req*(In)^2));
        inmax=sqrt(((Wo*(Vn)^2)/(Vo)^2)/(req*(In)^2));
        rdtomax=100*((inmax*SnMono*fdp)/(inmax*SnMono*fdp+(Wo*(Vn)^2)/(V
o)^2)+(inmax)^2*(req*(In)^2));

        go=Wo/(Vo)^2;
        yo=Io/Vo;
        bo=sqrt(yo^2-go^2);

        perdidas=((Wo*(TensionDato)^2)/(Vo)^2)+req*(int)^2;

        set(handles.Ur, 'string', ur*100);
        set(handles.Ux, 'string', ux*100);
        set(handles.Uz, 'string', uz*100);
        set(handles.tensionsecundario, 'string', tensioncalculada/1000);
    
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
set(handles.text66,'string',int/rt);
set(handles.text88,'string',req*(int)^2);
set(handles.text89,'string',((Wo*(TensionDato)^2)/(Vo)^2));

end

if tipol==2

%tensionnominalrt=TensionDato/rt;

In=SnMono/Vn;
intsecun=cargasec/(tensionbornescarga*fdpsec);
ifactcarga=intsecun/(In*rt);
fisecun=acos(fdpsec);
req=Wcc/(Icc^2);
zeq=Vcc/Icc;
xeq=sqrt(zeq^2-req^2);
aux=(intsecun)*sin(fisecun);
if C1==3
ivectorial=intsecun*cos(fisecun)-i*(aux);
tensionsecun=tensionbornescarga*rt+(ivectorial/rt)*(req+i*(xeq));
elseif C1==1
ivectorial=intsecun;

tensionsecun=tensionbornescarga*rt+(ivectorial/rt)*(req+i*(xeq));
else
ivectorial=intsecun*cos(fisecun)+i*(aux);
tensionsecun=tensionbornescarga*rt+(ivectorial/rt)*(req+i*(xeq));
end
tensionsecun=abs(tensionsecun);
if A1==3
tensionsecun=tensionsecun*sqrt(3);
end

uxsecun=(xeq*(In/rt))/V2n;
ursecun=(req*(In/rt))/V2n;
uzsecun=(zeq*(In/rt))/V2n;

go=Wo/(Vo)^2;
yo=Io/Vo;
bo=sqrt(yo^2-go^2);

perdidas=((Wo*(tensionbornescarga*rt)^2)/((Vo)^2))+ (req*(intsecun/rt)^2);

rdto=100*((ifactcarga*SnMono*fdpsec)/(ifactcarga*SnMono*fdpsec+((Wo*(tensionbornescarga*rt)^2)/((Vo)^2))+ (ifactcarga)^2*(req*(In)^2)));
inmax=sqrt(((Wo*(Vn)^2)/((Vo)^2))/(req*(In)^2));
rdtomax=100*((inmax*SnMono*fdpsec)/(inmax*SnMono*fdpsec+((Wo*(Vn)^2)/((Vo)^2))+ (inmax)^2*(req*(In)^2)));

set(handles.Ur,'string',ursecun*100);

set(handles.Ux,'string',uxsecun*100);

set(handles.Uz,'string',uzsecun*100);
```


Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Executes on button press in Reset.
function Reset_Callback(hObject, eventdata, handles)
% hObject    handle to Reset (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
SnMono=char(' ');
set(handles.editSnMono, 'String', SnMono);
Vn=char(' ');
set(handles.Vn, 'String', Vn);
rt=char(' ');
set(handles.rt, 'String', rt);
Vcc=char(' ');
set(handles.edit30, 'String', Vcc);
Icc=char(' ');
set(handles.edit31, 'String', Icc);
Wcc=char(' ');
set(handles.edit32, 'String', Wcc);
Vo=char(' ');
set(handles.Vo, 'String', Vo);
Io=char(' ');
set(handles.Io, 'String', Io);
Wo=char(' ');
set(handles.Wo, 'String', Wo);
TensionDato=char(' ');
set(handles.TensionDato, 'String', TensionDato);
CargaPrim=char(' ');
set(handles.CargaPrim, 'String', CargaPrim);
fdpPrim=char(' ');
set(handles.fdpPrim, 'String', fdpPrim);
set(handles.text63, 'String', ' ');
set(handles.text66, 'String', ' ');
set(handles.Req, 'String', ' ');
set(handles.Xeq, 'String', ' ');
set(handles.Zeq, 'String', ' ');
set(handles.Uz, 'String', ' ');
set(handles.tensionsecundario, 'String', ' ');
set(handles.Rendimiento, 'String', ' ');
set(handles.Maximo, 'String', ' ');
set(handles.perdidas, 'String', ' ');
set(handles.V2n, 'String', ' ');
set(handles.Ur, 'String', ' ');
set(handles.Ux, 'String', ' ');
set(handles.tensionsecun, 'String', ' ');
set(handles.intsecun, 'String', ' ');
set(handles.fdpsec, 'String', ' ');
set(handles.cargasec, 'String', ' ');
set(handles.tensionbornescarga, 'String', ' ');
set(handles.go, 'String', ' ');
set(handles.bo, 'String', ' ');
set(handles.yo, 'String', ' ');
set(handles.text88, 'String', ' ');
set(handles.text89, 'String', ' ');

% --- Executes on selection change in MonooTri.
function MonooTri_Callback(hObject, eventdata, handles)
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% hObject      handle to MonooTri (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
MonooTri contents as cell array
%             contents{get(hObject,'Value')} returns selected item
from MonooTri

% --- Executes during object creation, after setting all
properties.
function MonooTri_CreateFcn(hObject, eventdata, handles)
% hObject      handle to MonooTri (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: listbox controls usually have a white background on
Windows.
%             See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in primosec.
function primosec_Callback(hObject, eventdata, handles)
% hObject      handle to primosec (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
primosec contents as cell array
%             contents{get(hObject,'Value')} returns selected item
from primosec

% --- Executes during object creation, after setting all
properties.
function primosec_CreateFcn(hObject, eventdata, handles)
% hObject      handle to primosec (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: listbox controls usually have a white background on
Windows.
%             See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
end

function rt_Callback(hObject, eventdata, handles)
% hObject    handle to rt (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rt as text
%        str2double(get(hObject,'String')) returns contents of rt
as a double
rt=get(hObject,'String');
rt=str2double(rt);
if isnan(rt)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.editSnMono,'String',0);
    rt=0;
end
handles.edit=rt;
guidata (hObject,handles);

% --- Executes during object creation, after setting all
properties.
function rt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rt (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Tri.
function Tri_Callback(hObject, eventdata, handles)
% hObject    handle to Tri (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Tri

% --- Executes on button press in Mono.
function Mono_Callback(hObject, eventdata, handles)
% hObject    handle to Mono (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% Hint: get(hObject,'Value') returns toggle state of Mono

function V2n_Callback(hObject, eventdata, handles)
% hObject    handle to V2n (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of V2n as text
%         str2double(get(hObject,'String')) returns contents of
V2n as a double
set(handles.rt,'String',(str2double(get(handles.Vn,'String'))/str
2double(get(hObject,'String'))));

V2n=get(hObject,'String');
V2n=str2double(V2n);
if isnan(V2n)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.V2n,'String',' ');
    V2n=0;
end

% --- Executes during object creation, after setting all
properties.
function V2n_CreateFcn(hObject, eventdata, handles)
% hObject    handle to V2n (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fdpsec_Callback(hObject, eventdata, handles)
% hObject    handle to fdpsec (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fdpsec as text
%         str2double(get(hObject,'String')) returns contents of
fdpsec as a double

% --- Executes during object creation, after setting all
properties.
function fdpsec_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fdpsec (see GCBO)
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function cargasec_Callback(hObject, eventdata, handles)
% hObject handle to cargasec (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cargasec as
text
% str2double(get(hObject,'String')) returns contents of
cargasec as a double

% --- Executes during object creation, after setting all
properties.
function cargasec_CreateFcn(hObject, eventdata, handles)
% hObject handle to cargasec (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tensionbornescarga_Callback(hObject, eventdata, handles)
% hObject handle to tensionbornescarga (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
tensionbornescarga as text
% str2double(get(hObject,'String')) returns contents of
tensionbornescarga as a double
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Executes during object creation, after setting all
properties.
function tensionbornescarga_CreateFcn(hObject, eventdata,
handles)
% hObject handle to tensionbornescarga (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in checkbox8.
function checkbox8_Callback(hObject, eventdata, handles)
% hObject handle to checkbox8 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox8

% --- Executes on button press in checkbox9.
function checkbox9_Callback(hObject, eventdata, handles)
% hObject handle to checkbox9 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox9

% --- Executes on button press in checkbox10.
function checkbox10_Callback(hObject, eventdata, handles)
% hObject handle to checkbox10 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox10

% --- Executes on button press in checkbox11.
function checkbox11_Callback(hObject, eventdata, handles)
% hObject handle to checkbox11 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox11
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% --- Executes when selected object is changed in tipotrafo.
function tipotrafo_SelectionChangedFcn(hObject, eventdata,
handles)
% hObject handle to the selected object in tipotrafo
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
A=get(hObject,'String');
switch A
    case 'Monofasico'
        set(handles.text4,'string','V1n');
        set(handles.text93,'string','V2n');
    case 'Trifasico'
        set(handles.text4,'string','U1n');
        set(handles.text93,'string','U2n');
end

% --- Executes when selected object is changed in tipocargaB.
function tipocargaB_SelectionChangedFcn(hObject, eventdata,
handles)
% hObject handle to the selected object in tipocargaB
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes when selected object is changed in tipocargaC.
function tipocargaC_SelectionChangedFcn(hObject, eventdata,
handles)
% hObject handle to the selected object in tipocargaC
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes when uipanel6 is resized.
function uipanel6_SizeChangedFcn(hObject, eventdata, handles)
% hObject handle to uipanel6 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes when selected object is changed in uibuttongroup9.
function uibuttongroup9_SelectionChangedFcn(hObject, eventdata,
handles)
% hObject handle to the selected object in uibuttongroup9
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
tipo=get(hObject,'String');
switch tipo
    case 'Tension en bornes'
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
set(handles.tensionbornescarga, 'Enable', 'Off');
set(handles.cargasec, 'Enable', 'Off');
set(handles.fdpsec, 'Enable', 'Off');
set(handles.TensionDato, 'Enable', 'On');
set(handles.CargaPrim, 'Enable', 'On');
set(handles.fdpPrim, 'Enable', 'On');

set(handles.radiobutton10, 'Enable', 'Off');
set(handles.radiobutton11, 'Enable', 'Off');
set(handles.radiobutton17, 'Enable', 'Off');
set(handles.radiobutton7, 'Enable', 'On');
set(handles.radiobutton9, 'Enable', 'On');
set(handles.radiobutton16, 'Enable', 'On');

case 'Tension primario'
    set(handles.tensionbornescarga, 'Enable', 'On');
    set(handles.cargasec, 'Enable', 'On');
    set(handles.fdpsec, 'Enable', 'On');
    set(handles.TensionDato, 'Enable', 'Off');
    set(handles.CargaPrim, 'Enable', 'Off');
    set(handles.fdpPrim, 'Enable', 'Off');

    set(handles.radiobutton10, 'Enable', 'On');
    set(handles.radiobutton11, 'Enable', 'On');
    set(handles.radiobutton17, 'Enable', 'On');
    set(handles.radiobutton7, 'Enable', 'Off');
    set(handles.radiobutton9, 'Enable', 'Off');
    set(handles.radiobutton16, 'Enable', 'Off');

end

function edit30_Callback(hObject, eventdata, handles)
% hObject      handle to edit30 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit30 as text
%          str2double(get(hObject,'String')) returns contents of
edit30 as a double
edit30=get(hObject,'String');
edit30=str2double(Vn);
if isnan(edit30)
    errordlg('El valor debe ser numérico.', 'ERROR')
    set(handles.edit30, 'String', ' ');
    edit30=0;
end

% --- Executes during object creation, after setting all
properties.
function edit30_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit30 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit31_Callback(hObject, eventdata, handles)
% hObject      handle to edit31 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit31 as text
% str2double(get(hObject,'String')) returns contents of
edit31 as a double
edit31=get(hObject,'String');
edit31=str2double(Vn);
if isnan(edit31)
    errordlg('El valor debe ser numérico.','ERROR')
    set(handles.edit31,'String',' ');
    edit31=0;
end

% --- Executes during object creation, after setting all
properties.
function edit31_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit31 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit32_Callback(hObject, eventdata, handles)
% hObject      handle to edit32 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit32 as text
% str2double(get(hObject,'String')) returns contents of
edit32 as a double
if get(handles.radiobutton3,'Value')==1.0
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
        A1=1;
    else
        A1=3;
    end
    Vcc=get(handles.edit30,'String');
    Vcc=str2double(Vcc)*1000;
    if isnan(Vcc)
        errordlg('El valor debe ser numérico.','ERROR')
        set(handles.Vcc,'String',' ');
        Vcc=0;
    end
    if A1==3
        Vcc=Vcc/sqrt(3);
    end

    Icc=get(handles.edit31,'String');
    Icc=str2double(Icc);
    if isnan(Icc)
        errordlg('El valor debe ser numérico.','ERROR')
        set(handles.Icc,'String',' ');
        Icc=0;
    end

    Wcc=get(hObject,'String');
    Wcc=str2double(Wcc)*1000;
    if isnan(Wcc)
        errordlg('El valor debe ser numérico.','ERROR')
        set(handles.Wcc,'String',' ');
        Wcc=0;
    end
    if A1==3
        Wcc=Wcc/3;
    end

    req=(Wcc/Icc^2);
    zeq=(Vcc/Icc);
    xeq=sqrt(zeq^2-req^2);

    set(handles.Req,'string',req);

    set(handles.Xeq,'string',xeq);

    set(handles.Zeq,'string',zeq);

    % --- Executes during object creation, after setting all
    properties.
    function edit32_CreateFcn(hObject, eventdata, handles)
        % hObject    handle to edit32 (see GCBO)
        % eventdata  reserved - to be defined in a future version of
MATLAB
        % handles    empty - handles not created until after all
CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
        %         See ISPC and COMPUTER.
        if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
```

Trabajo Fin de Grado: Herramienta de cálculo de transformadores de potencia. Caídas de tensión, pérdidas y rendimiento.

```
end

% --- Executes on button press in radiobutton16.
function radiobutton16_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton16 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
radiobutton16

set(handles.fdpPrim, 'String',1);

% --- Executes on button press in radiobutton17.
function radiobutton17_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton17 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
radiobutton17
set(handles.fdpsec, 'String',1);
```