



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE
TELECOMUNICACIONES
DEPARTAMENTO DE
ELECTROMAGNETISMO Y RADIACIÓN



**DESARROLLO DE UN SOFTWARE EDUCATIVO DIDÁCTICO
INTERACTIVO PARA SISTEMAS DE ONDAS GUIADAS VERSIÓN 3**

ALVARADO ENRIQUE
VARGAS ANGEL

Bárbula, 2 de diciembre del 2016



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE
TELECOMUNICACIONES
DEPARTAMENTO DE
ELECTROMAGNETISMO Y RADIACIÓN



CERTIFICADO DE APROBACIÓN

Los abajo firmantes miembros del jurado asignado para evaluar el trabajo especial de grado titulado «DESARROLLO DE UN SOFTWARE EDUCATIVO DIDÁCTICO INTERACTIVO PARA SISTEMAS DE ONDAS GUIADAS VERSIÓN 3», realizado por los bachilleres ALVARADO ENRIQUE, cédula de identidad 20.727.387, VARGAS ANGEL, cédula de identidad 21.480.211, hemos decidido otorgar la máxima calificación y la mención honorífica al presente trabajo, con base a los siguientes motivos:

Firma

Prof. PAULINO DEL PINO
TUTOR

Firma

Prof. MARELYN SÁEZ
JURADO

Firma

Prof. CARLOS MEJÍAS
JURADO

Bárbula, 2 de diciembre del 2016

Índice general

Índice de Figuras	XI
Índice de Tablas	XV
Índice de Códigos	XVII
Acrónimos	XIX
Constantes Físicas	XXI
Símbolos	XXIII
Resumen	XXV
I. Introducción	1
1.1. Motivación	1
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	4
1.3. Alcance	4
II. Marco Conceptual	7
2.1. Principio y Ecuaciones Fundamentales de la Línea de Transmisión	7
2.1.1. Solución a las ecuaciones de los telegrafistas	9
2.1.1.1. Línea sin pérdidas ($r = g = 0$)	9
2.1.1.2. Caso Armónico	11
2.2. FDTD en una Dimensión	13
2.2.1. Características Espaciales	13
2.2.2. Características Temporales	14
2.3. Entorno Gráfico de Simulación	14
III. Procedimientos de la investigación	17
3.1. Exploración Bibliográfica	17
3.2. Selección de paquetes de Python apropiados	17

3.2.1.	Paquete de cálculos numéricos	18
3.2.2.	Paquete para realizar y mostrar gráficos	19
3.2.3.	Paquete para crear una interfaz gráfica de usuario (GUI)	19
3.3.	Investigación y estudio de python y paquetes seleccionados	20
3.4.	Estudio y visualización de los transitorios presente en un arreglo de líneas de transmisión mediante el uso de FDTD	21
3.5.	Diseño y estructura de la Aplicación	32
3.5.1.	Módulo para la representación de transitorios en un arreglo de líneas de transmisión	32
3.5.2.	Módulos para análisis del régimen permanente existente en la línea de transmisión	38
3.5.2.1.	Módulo de Relación de Onda Estacionaria (ROE)	38
3.5.2.2.	Módulo de adaptación de impedancia mediante una línea terminada en cortocircuito dispuesta en paralelo	40
3.5.2.3.	Módulo de adaptación de impedancia mediante dos líneas terminadas en cortocircuito dispuestas en paralelo	42
3.5.2.4.	Módulo de adaptación de impedancia mediante el transformador de cuarto de onda	45
3.5.2.5.	Módulo Carta de Smith	49
3.5.2.6.	Módulo de adaptación de impedancia mediante el transformador binomial	55
3.5.2.7.	Módulo de adaptación de impedancia mediante el transformador Chebyshev	56
3.5.2.8.	Módulo de adaptación de impedancia mediante el transformador de línea no uniforme (Tapered Line)	60
3.5.2.9.	Módulo de balun de constantes distribuidas	61
3.5.2.10.	Módulo de balun con elementos concentrados	62
3.5.2.11.	Módulo de parámetros de dispersión	63
3.5.3.	Módulo de interacción virtual con equipos y accesorios presentes en el laboratorio de Sistemas de Ondas Guiadas	63
3.6.	Construcción de SEDISOG V3.0 usando Python	64
IV.	Análisis, interpretación y presentación de los resultados	67
4.1.	Logo del programa.	67
4.2.	Interfaz gráfica de usuario	68
4.2.1.	Ventana principal	68
4.2.2.	Ventanas auxiliares	69
4.2.3.	Ventana para el módulo transitorio	73
4.2.4.	Ventana para el módulo relación de onda estacionaria (ROE)	78
4.2.5.	Ventana para el módulo transformador de cuarto de onda	79
4.2.6.	Ventana para el módulo transformador Binomial	81
4.2.7.	Ventana para el módulo transformador Chebyshev	83

4.2.8. Ventana para el módulo adaptación con un stub	84
4.2.9. Ventana para el módulo adaptación con dos stubs	86
4.2.10. Ventana para el módulo balun con elementos concentrados	88
4.2.11. Ventana para el módulo balun con elementos distribuidos	89
4.2.12. Ventana para el módulo Carta de Smith	90
4.2.13. Ventana para el módulo línea no uniforme	93
4.2.14. Ventana para el módulo parámetros de dispersión	94
4.2.15. Ventana para el laboratorio virtual de SOG	96
4.3. Requerimientos del sistema	99
4.3.1. Requerimientos de hardware	99
4.3.2. Requerimientos de software	99
V. Conclusiones y recomendaciones	101
5.1. Conclusiones	101
5.2. Recomendaciones	102
Referencias Bibliográficas	105

Índice de figuras

2.1. Modelo adoptado para la línea de transmisión de longitud L	7
2.2. Constitución de las celdas del modelo.	8
2.3. Solución a la Ecuación (2.5) en dos instantes t_1 y t_2 . (Fuente [2])	11
3.1. Disposición temporal y espacial del Voltaje y Corriente, siendo las X's rojas la tensión y los círculos negros la corriente. (Fuente [12])	22
3.2. Equivalente de Norton aplicado al generador y primera celda del modelo de la línea de transmisión. (Fuente [3])	24
3.3. Modelo usado para la ultima celda, conectando una carga Z_L en el extremo de la línea (Fuente [3])	26
3.4. Modelo equivalente para la conexión de tres líneas de transmisión en paralelo. (Fuente [14])	31
3.5. Arreglos de líneas de transmisión	33
3.6. Diagramas de Flujo para transitorios en arreglos lineales	34
3.7. Diagrama para el cálculo de V e I en tres líneas dispuestas en paralelo	35
3.8. Diagramas de Flujo de los bloques para calcular de V e I en cada línea	36
3.9. Diagrama de flujo para la creación del diagrama de reflexiones	37
3.10. Diagrama de flujo para el cálculo de los parámetros de salida del modulo ROE	39
3.11. Adaptación mediante 1 stub: Algoritmo de inicio y resultados principales.	41
3.12. Adaptación mediante 1 stub: Algoritmo de variaciones sobre el sistema.	42
3.13. Adaptación mediante 2 stub: Algoritmo de inicio y resultados principales.	44
3.14. Adaptación mediante 2 stubs: Algoritmo de variaciones sobre el sistema.	45
3.15. Adaptación mediante transformador de cuarto de onda: Algoritmo de inicio y resultados principales.	47
3.16. Adaptación mediante transformador de cuarto de onda: Algoritmo de variaciones sobre el sistema.	48
3.17. Diagrama de flujo para el asistente o ayuda de la Carta de Smith	50
3.18. Diagrama de flujo para calcular $Z(d)$	51

3.19. Diagrama de flujo para adaptar impedancia mediante un Stub	52
3.20. Diagrama de flujo para adaptar impedancia mediante dos Stubs	53
3.21. Diagrama de flujo para adaptar impedancia mediante elementos concentrados	54
3.22. Diagramas de flujo para los bloques de adaptar	55
3.23. Adaptación mediante transformador Binomial y Chebyshev: Algoritmo de inicio y resultados principales.	58
3.24. Adaptación mediante transformador Binomial y Chebyshev: Algoritmo de variaciones sobre el sistema.	59
4.1. Logo del programa	67
4.2. Ventana Principal de SEDISOG 3	68
4.3. Despliegue de la opción Archivo	70
4.4. Despliegue de la opción contenido	71
4.5. Contenido de la asignatura	72
4.6. Despliegue de la opción herramientas	73
4.7. Interfaz para el módulo Transitorio	74
4.8. Interfaz para visualizar transitorios en arreglos lineales	75
4.9. Arreglo de dos líneas	75
4.10. Interfaz para mostrar resultados al simular un arreglo de tres líneas dispuestas en paralelo	76
4.11. Menú para elegir vista	77
4.12. Diagrama de reflexiones	78
4.13. Interfaz para el módulo ROE	79
4.14. Interfaz para resultado del módulo ROE	79
4.15. Interfaz de introducción de datos del modulo adaptador de impedancias mediante el transformador de cuarto de onda	80
4.16. Interfaz de resultados del modulo adaptador de impedancias mediante el transformador de cuarto de onda	81
4.17. Interfaz de introducción de datos del modulo adaptador de impedancias mediante el transformador Binomial	82
4.18. Interfaz de resultados del modulo adaptador de impedancias mediante el transformador Binomial	82
4.19. Interfaz de introducción de datos del modulo adaptador de impedancias mediante el transformador Chebyshev	84
4.20. Interfaz de resultados del modulo adaptador de impedancias mediante el transformador Chebyshev	84
4.21. Interfaz de introducción de datos del modulo adaptador de impedancias mediante un stub	85
4.22. Interfaz de resultados del modulo adaptador de impedancias mediante un stub	86
4.23. Interfaz de introduccion de datos del modulo adaptador de impedancias mediante dos stub	87

4.24. Interfaz de resultados del modulo adaptador de impedancias mediante dos stub	87
4.25. Interfaz de introduccion de datos del modulo balun de elementos concentrados	88
4.26. Interfaz de resultados del modulo balun de elementos concentrados	89
4.27. Interfaz de introduccion de datos del modulo balun de elementos distribuidos	90
4.28. Interfaz de resultados del modulo balun de elementos distribuidos	90
4.29. Interfaz para el módulo Carta de Smith	91
4.30. Adaptar impedancia con elementos concentrados mediante el uso de la Carta de Smith	92
4.31. Asistente del módulo Carta de Smith	92
4.32. Interfaz de introducción de datos del modulo Línea No Uniforme	94
4.33. Interfaz de resultados del modulo Línea No Uniforme	94
4.34. Interfaz de introducción de datos del modulo Parámetros de Dispersión de una Línea de transmisión	95
4.35. Interfaz de resultados del modulo Parámetros de Dispersión de una Línea de transmisión	96
4.36. Menú LabSOG	97
4.37. Interfaz del Laboratorio Virtual	97
4.38. Montaje en el Laboratorio Virtual	98

Acrónimos

UC	Universidad de Carabobo
SOG	Sistemas de Ondas Guiadas
SEDISOG	Software Educativo Didáctico e Interactivo para Sistemas de Ondas Guiadas
FDTD	Finite Diference Time Domain
RF	Radio Frecuencia
GUI	Graphical User Interface
PC	Personal Computer

Constantes Físicas

$$\text{Velocidad de la luz } c = 2,997\,924\,58 \times 10^8 \text{ ms}^{-1}$$

Símbolos

V	Voltaje	V
I	Corriente	A
r	Resistencia por unidad de longitud	(Ω/m)
l	Inductancia por unidad de longitud	(H/m)
c	Capacitancia por unidad de longitud	(F/m)
g	Conductancia por unidad de longitud	(S/m)
γ	Constante de propagación	
α	Constante de atenuación	(N/m)
β	Constante de fase	(rad/m)
λ	Longitud de Onda	(m)
Z	Impedancia por unidad de longitud	$(\Omega/u.l.)$
Y	Admitancia por unidad de longitud	$(S/u.l.)$

DESARROLLO DE UN SOFTWARE EDUCATIVO DIDÁCTICO INTERACTIVO PARA SISTEMAS DE ONDAS GUIADAS VERSIÓN 3

por

ALVARADO ENRIQUE y VARGAS ANGEL

Presentado en el Departamento de Electromagnetismo y Radiación
de la Escuela de Ingeniería en Telecomunicaciones
el 2 de diciembre del 2016 para optar al Título de
Ingeniero de Telecomunicaciones

RESUMEN

En la Universidad de Carabobo, específicamente en la escuela de Ingeniería de Telecomunicaciones y la escuela de Ingeniería Eléctrica, se dicta la asignatura "Sistemas de Ondas Guiadas", en la cual se estudia el proceso de transmisión de energía electromagnética en RF, existente en la línea de Transmisión, dicho estudio consta del análisis de dos casos: régimen transitorio y régimen permanente, ambos requieren un nivel adecuado de abstracción, debido a su complejidad visual y matemática. De este hecho surge el objetivo principal del presente trabajo especial de grado que consiste en la creación de una herramienta computacional que sea capaz de ilustrar los fundamentos teóricos de la asignatura, de manera que facilite a los estudiantes comprender y visualizar los contenidos, de igual forma complementar los esfuerzos docentes al impartir el conocimiento.

Como antecedente al proyecto que se plantea actualmente, la asignatura cuenta con un programa creado mediante un trabajo especial de grado, titulado "Software Educativo Didáctico Interactivo para Sistemas de Ondas Guiadas Versión 2" (SE-DISOG V.2) el cual se pretende mejorar, creando con una nueva versión llamada SEDISOG V.3.

SEDISOG V.3 será desarrollada bajo el lenguaje Python, e incorporará herramientas que permitan realizar estudios más completos en el régimen transitorio cuando se tienen arreglos de líneas de transmisión, mientras que para el régimen permanente, se añadirán opciones que permitan estudiar parámetros importantes en la asignatura con más detalle; dichas mejoras contribuirán a desarrollar el aprendizaje de los estudiantes, ya que al tener un contacto más directo e interactivo con las gráficas del régimen permanente y transitorio se reforzarán los conocimientos y conceptos impartidos por el docente durante la clase. La nueva versión contará también con una interfaz gráfica que simule los equipos y accesorios existentes en el laboratorio de la asignatura, brindando así versatilidad y autonomía al usuario para hacer montajes con la posibilidad de analizarlos posteriormente, de manera que tenga una experiencia previa al momento de realizar la práctica con los equipos reales en el laboratorio. Sin duda con estas mejoras el usuario se trasladará a un ambiente autodidacta.

Palabras Claves: Líneas de Transmisión, Simulación, Didáctico, Autodidacta

Tutor: PAULINO DEL PINO

Profesor del Departamento de Electromagnetismo y Radiación

Escuela de Telecomunicaciones. Facultad de Ingeniería

Capítulo I

Introducción

1.1. Motivación

El proceso de estudiar y comprender asignaturas en el área de electromagnetismo requiere de una alta exigencia de análisis y razonamiento abstracto, debido a su complejidad visual y matemática. Para ello se han involucrado herramientas de tipo computacional capaces de brindar soporte a los conocimientos que se adquieren en la teoría, ofreciendo un entorno donde se pueda simular, analizar y visualizar la naturaleza de los fenómenos electromagnéticos, los cuales en su mayoría están descritos por expresiones matemáticas complejas, por ende el estudio de sistemas de ondas guiadas no escapa de esta realidad.

En la Facultad de Ingeniería de la Universidad de Carabobo específicamente en la Escuela de Ingeniería de Telecomunicaciones y la Escuela de Ingeniería Eléctrica es dictada la asignatura "Sistemas de Ondas Guiadas", la cual se enfoca al estudio del proceso de transmisión de energía electromagnética en RF, a través de la línea de transmisión.

El estudio del proceso de transmisión se realiza mediante el análisis del régimen transitorio y el análisis del régimen permanente.

En el régimen transitorio donde se observa como evoluciona la onda electromagnética dentro de la línea de transmisión en función del tiempo, la posición y las condiciones iniciales, se presenta una dificultad para los estudiantes, en la visualización del comportamiento de la tensión (V) y la Corriente (I) por la presencia de dichas dependencias. Mientras que para el análisis en régimen permanente no existe una dependencia temporal logrando una visualización más sencilla y de fácil comprensión, pero al desarrollar este estudio conduce a un conjunto de modelos matemáticos tediosos de resolver. Es posible encontrar en Internet programas o aplicaciones que realicen los cálculos de interés de forma rápida, sin embargo estos presentan un problema para los estudiantes, dado que no se ajustan a los requerimientos y contenido de la asignatura.

Es así como todas las dificultades y problemas para comprender los conceptos relacionados con la asignatura sistemas de ondas guiadas planteados en párrafos anteriores, evidencian la necesidad de una herramienta visual, didáctica e interactiva que sirva como recurso para el aprendizaje de los estudiantes y material de apoyo para el docente.

Atendiendo la dificultad de la asignatura y a la necesidad de herramientas adicionales para su mejor comprensión por parte de los estudiantes, se creó un software didáctico SEDISOG V.2 elaborado por un trabajo anterior [1], el cual brinda soporte en el cálculo de régimen transitorio, régimen estacionario y en el manejo de ecuaciones. Sin embargo, la tecnología avanza a pasos agigantados en la actualidad, el ser humano es bombardeado diariamente con nuevos avances e inventos que apuntan al desarrollo de la ciencia en múltiples ámbitos; y las universidades al ser la cuna del conocimiento y el progreso no escapan de esta realidad, no deben quedarse atrás. De este hecho surge la necesidad de mantener nuevas herramientas didácticas y recursos tecnológicos para ampliar las capacidades de comprensión y análisis de los conceptos impartidos en la cátedra de SOG.

En busca de satisfacer los nuevos desafíos de la tecnología, que imponen el hecho de tener un software actualizado y que abarque en su totalidad la asignatura, e igualmente sea capaz de ofrecer mayor libertad y versatilidad al usuario, con opciones nuevas y algunas optimizadas, tal que el enfoque sea aún más autodidacta, se

plantea realizar una revisión, del software ya existente que permita crear una nueva herramienta visual, didáctica e interactiva que incluya las ideas mencionadas y de esta manera darle la bienvenida a la nueva versión SEDISOG V.3.

Considerando la problemática y con el fin de beneficiar a personas que deseen incursionar en esta área del conocimiento, se plantea que esta nueva versión de SEDISOG, permita al usuario tener un material didáctico de fácil acceso y que facilite el proceso de aprendizaje, al ritmo deseado.

En esta versión se busca abarcar todo el contenido de la asignatura, con la finalidad de ofrecer apoyo a usuarios que con inquietud, desean complementar lo visto en clase.

En la herramienta se encontrará la teoría que sustenta la asignatura y además de simulaciones que permitan tener una mejor y confiable visualización del comportamiento de fenómenos presentes en las líneas de transmisión, que contraste la teoría con la práctica. El software dispondrá de una herramienta para el análisis del régimen transitorio que facilite observar los transitorios presente en la línea o arreglo de líneas de transmisión con su respectivo diagrama de reflexiones, en las herramientas de cálculos interactivas para el régimen permanente se incorporará la opción de observar el comportamiento de nuevos parámetros, los cuales proporcionaran un análisis mas detallado de voltaje y corriente existente en la línea de transmisión.

Igualmente se incorporará un laboratorio virtual que ofrezca libertad al usuario de disponer y manipular equipos además de accesorios que se encuentran en el laboratorio de la asignatura y realizar montajes de forma interactiva para luego observar los resultados del mismo y analizarlos, de manera que el usuario tenga una experiencia previa al realizar la práctica.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una versión mejorada y ampliada del "Software Educativo Didáctico Interactivo para Sistemas de Ondas Guiadas Versión 2" (SEDISOG V.2) que permita reforzar el proceso de enseñanza aprendizaje en la asignatura "Sistema de Ondas Guiadas".

1.2.2. Objetivos Específicos

- Representar los transitorios presentes en un arreglo de hasta tres líneas de transmisión dispuestas, en serie o tres en paralelo, con diferentes cargas y tipos de excitación haciendo uso de FDTD en una dimensión.
- Desarrollar los nuevos módulos para el análisis del régimen permanente y añadir nuevas opciones como, adaptar con elementos concentrados, transformador de línea no uniforme, además de añadir nuevas opciones a los ya existente.
- Incorporar un entorno virtual que permita realizar ensayos del laboratorio de Sistemas de Ondas Guiadas con un mayor grado de autonomía y versatilidad para el usuario, con respecto a la versión anterior. enfocado a la posibilidad de realizar prácticas libremente sin un patrón a seguir.

1.3. Alcance

Por medio del presente trabajo de grado se realizará el diseño y la elaboración de SEDISOG V.3 basado en la versión anterior (SEDISOG V.2). Esta actualización tendrá la teoría de cada capítulo contemplado en la asignatura "Sistemas de Ondas Guiadas", las herramientas dispuestas en el software pueden incluir animaciones, simulaciones, o un asistente que indique los pasos de un procedimiento específico, de acuerdo con el tema que este siendo tratado.

En el régimen transitorio, se dispondrá de una interfaz para visualizar la evolución temporal de tensión (V) y corriente (I) con el respectivo diagrama de reflexiones también llamado diagrama de Bewley, para cualquier tipo de arreglo permitido (un máximo de 3 líneas dispuestas en serie o tres líneas en paralelo). Para el caso de las líneas en serie se podrá colocar una discontinuidad resistiva en serie o paralelo con la línea o arreglo bajo estudio, en caso general el tipo de fuente es diverso (pulso Gaussiano, pulso trapesoidal, senoidal y escalón) y las cargas R, L y C con sus posibles combinaciones seguirán formando parte en esta versión. Debe señalarse que a al momento de simular el arreglo de tres líneas en paralelo, la interfaz de visualización de resultado contará con un menú donde el usuario escogerá la vista que se ajuste a sus requerimientos, para el régimen permanente en el módulo referente a la carta de Smith se añadirá la opción de adaptar con elementos concentrados, además para todas las opciones presente en este módulo, entrará en acción un asistente el cual indica un paso a paso del procedimiento necesario para realizar la tarea seleccionada.

El laboratorio virtual adquirirá una interfaz tal que ofrezca un mayor grado de autonomía y versatilidad al usuario para que este sea capaz de interactuar con elementos virtuales que simulan los existentes en el laboratorio, con un aspecto realista.

Capítulo II

Marco Conceptual

2.1. Principio y Ecuaciones Fundamentales de la Línea de Transmisión

Para estudiar el comportamiento del Voltaje (V) y la Corriente (I) postularemos una segmentación de la línea de transmisión en secciones o celdas de ancho Δx , como se muestra en la figura 2.1, considerando que el tamaño de estas debe ser mucho menor a la longitud de onda.

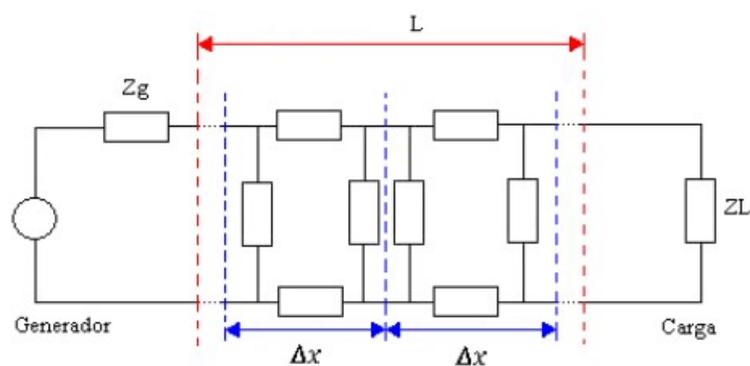


Figura 2.1: Modelo adoptado para la línea de transmisión de longitud L .

De acuerdo con lo postulado, en la figura 2.2 se muestran los elementos que componen cada una de las celdas, estos son:

- r resistencia por unidad de longitud [Ω/m]
- l inductancia por unidad de longitud [H/m]
- c capacidad por unidad de longitud [F/m]
- g conductancia por unidad de longitud [S/m]

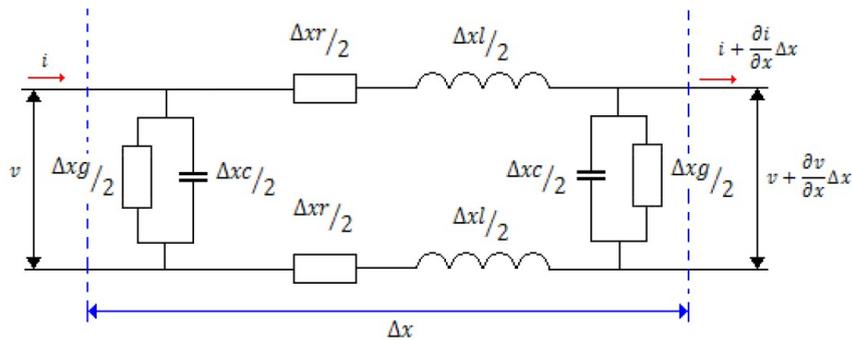


Figura 2.2: Constitución de las celdas del modelo.

Los parámetros r, l, c y g , involucrados en el modelo, se dice que son constantes distribuidas, puesto que a lo largo de la línea, desde el generador a la carga, r, l, c y g se encuentran distribuidos en ella. Además, v e i son funciones de la variable espacial x y del tiempo t .

Al aplicar las leyes de corriente y voltaje de Kirchhoff, y luego hacer tender a cero la dimensión de la celda, obtenemos las siguientes ecuaciones:

$$-\frac{\partial v}{\partial x} = ri + l \frac{\partial i}{\partial t} \quad (2.1)$$

$$-\frac{\partial i}{\partial x} = gv + c \frac{\partial v}{\partial t} \quad (2.2)$$

siendo las ecuaciones 2.1 y 2.2 las "ecuaciones de los telegrafistas"[2].

2.1.1. Solución a las ecuaciones de los telegrafistas

Derivando las ecuaciones 2.1 y 2.2 y efectuando sustituciones para tener ecuaciones solo en v e i , resulta:

$$-\frac{\partial^2 v}{\partial x^2} = lc \frac{\partial^2 v}{\partial t^2} + (rc + lg) \frac{\partial v}{\partial t} + rgv \quad (2.3)$$

$$-\frac{\partial^2 i}{\partial x^2} = lc \frac{\partial^2 i}{\partial t^2} + (rc + lg) \frac{\partial i}{\partial t} + rgi \quad (2.4)$$

Las ecuaciones 2.3 y 2.4 son del mismo tipo denominado "ecuación de onda en una dimensión", cuya solución en general no es sencilla. Hay varios casos en los que debido a la introducción de hipótesis simplificativas la solución se facilita notablemente, y de ello se destacan ya sea por su carácter ilustrativo y/o aplicativo los siguientes:

1. La línea sin perdida.
2. La línea no distorcionante
3. el caso armónico (en el que la excitación es sinusoidal)

2.1.1.1. Línea sin perdidas ($r = g = 0$)

La línea sin perdidas es aquella donde $r = g = 0$, por ende las ecuaciones 2.3 y 2.4 se reducen a:

$$-\frac{\partial^2 v}{\partial x^2} = lc \frac{\partial^2 v}{\partial t^2} \quad (2.5)$$

$$-\frac{\partial^2 i}{\partial x^2} = lc \frac{\partial^2 i}{\partial t^2} \quad (2.6)$$

Las ecuaciones 2.5 y 2.6 también se denominan ecuaciones de onda en una dimensión. Estas tienen soluciones simples respecto a las ecuaciones 2.3 y 2.4.

Para encontrar la solución a estas ecuaciones se propone:

$$v(x, t) = f(x \pm at) \quad (2.7)$$

Donde a es la velocidad de propagación y viene expresada por:

$$a = \frac{1}{\sqrt{lc}} \quad (2.8)$$

Al sustituir la ecuación 2.8 en la expresión 2.5 y operar matemáticamente se tiene la siguiente ecuación:

$$v(x, t) = v_1(x - at) + v_2(x + at) + v_0 \quad (2.9)$$

Donde v_1 y v_2 son funciones cualesquiera que posean segunda derivada y v_0 es una constante. Dado que v e i no son independientes entre sí, si no que están vinculadas por las ecuaciones 2.5 y 2.6, es sencillo demostrar que la corriente viene representada por:

$$i(x, t) = \frac{1}{\sqrt{lc}} i_1(x - at) - \frac{1}{\sqrt{lc}} v_2(x + at) + i_0 \quad (2.10)$$

donde i_0 es una constante que no esta relacionada con v_0 . Puede comprobarse que cada uno de los términos del segundo miembro de la ecuación 2.9, es solución de la ecuación de onda (2.5) y al considerar el comportamiento de $v_1(x - at)$, resulta que se trata de una forma de onda que se desplaza en sentido de las x positivas con velocidad a (ver figura)

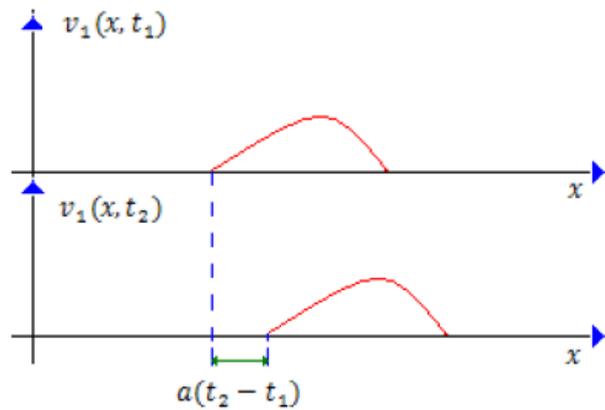


Figura 2.3: Solución a la Ecuación (2.5) en dos instantes t_1 y t_2 . (Fuente [2])

2.1.1.2. Caso Armónico

En este caso se parte del hecho en que la excitación es sinusoidal por lo que las tensiones y corrientes instantáneas a lo largo de la línea vendrán dadas por:

$$v(x, t) = |V(x)| \cos(\omega t + \phi_v) \quad (2.11)$$

$$i(x, t) = |I(x)| \cos(\omega t + \phi_i) \quad (2.12)$$

donde $|V(x)|$ y $|I(x)|$ son los valores pico de tensión y corriente respectivamente en el punto de abscisa x , mientras que ϕ_v y ϕ_i son las fases correspondientes. Las Ecuaciones 2.11 y 2.12 mediante el cálculo complejo pueden ser escritas así:

$$v(x, t) = |V(x)| e^{j\phi_v} \quad (2.13)$$

$$i(x, t) = |I(x)| e^{j\phi_i} \quad (2.14)$$

Las ecuaciones 2.13 y 2.14 se denominan fasores tensión corriente. Al sustituir dichas expresiones en las ecuaciones de los telegrafista se obtiene:

$$\frac{\partial V(x)}{\partial x} = -ZI(x) \quad (2.15)$$

$$\frac{\partial I(x)}{\partial x} = -YV(x) \quad (2.16)$$

donde

$$Z = Z(\omega) = r + j\omega l \text{ Impedancia por unidad de longitud}$$

$$Y = Y(\omega) = g + j\omega c \text{ Admitancia por unidad de longitud}$$

Las ecuaciones 2.15 y 2.16 se conocen como ecuaciones de los telegrafistas en el dominio complejo. Diferenciando ambos miembros de estas ecuaciones y reemplazando, nos queda:

$$\frac{\partial^2 V(x)}{\partial x^2} = -ZYV(x) \quad (2.17)$$

$$\frac{\partial^2 I(x)}{\partial x^2} = -ZYI(x) \quad (2.18)$$

Las ecuaciones 2.17 y 2.18 son típicas ecuaciones de onda unidimensionales cuya solución es simple:

$$V(x) = Ae^{-\gamma x} + Be^{\gamma x} \quad (2.19)$$

$$I(x) = Ce^{-\gamma x} + De^{\gamma x} \quad (2.20)$$

donde

$$\gamma = \sqrt{ZY} \quad (2.21)$$

Se denomina constante de propagación, además A,B,C y D dependen de las condiciones de borde.

De la ecuación 2.21 se obtiene lo siguiente:

$$\gamma = \sqrt{ZY} = \sqrt{(r + j\omega l)(g + j\omega c)} \quad (2.22)$$

separando en parte real e imaginaria:

$$\gamma = \sqrt{ZY} = \alpha + j\beta \quad (2.23)$$

siendo α la constante de atenuación que viene en unidades de [N/m] y β es la constante de fase en [rad/m].

2.2. FDTD en una Dimensión

Al momento de simular la propagación de energía electromagnética en RF dentro de la línea de transmisión es necesario aplicar el método de las diferencias finitas en el dominio del tiempo en una dimensión, dado que este permite resolver numéricamente, además, aproximado, las ecuaciones de los telegrafistas dentro de la línea, aplicando diferencia finitas centrales con segundo orden de precisión. A su vez es necesario conocer las siguientes características [3].

2.2.1. Características Espaciales

Al momento de aplicar este método la zona de estudio se debe discretizar, generando una grilla, para el caso de una dimensión se hace similitud a una línea

segmentada en pasos de Δx , a cada uno de estos segmentos se le llama celda espacial, en cada una de estas celdas se debe cumplir las ecuaciones que describen el comportamiento de la línea de transmisión.

2.2.2. Características Temporales

En este método la dimensión temporal es también discretizada en forma dependiente de la frecuencia, de la discretización espacial y de los materiales que componen la región de estudio. El dominio temporal es tal que el voltaje y la corriente no coinciden en el tiempo, para ello se calcula primero, los valores de voltaje en todos los puntos del dominio computacional reservado para V , luego se calculan los valores de corriente en todos los puntos del dominio computacional reservado para I , haciendo uso de los valores de voltaje calculados anteriormente, y de esta manera se progresa en el tiempo, actualizando los valores de voltaje y corriente para todo el dominio computacional.

2.3. Entorno Gráfico de Simulación

Para representar el contenido de la asignatura de forma didáctica e interactiva se presenta una solución enfocada al desarrollo de una interfaz donde se pueda simular, analizar y visualizar, la teoría impartida en la asignatura "Sistemas de Ondas Guiadas", este entorno se crea para ser usado en un computador, dado que brinda beneficios, tales como, comodidad al usuario al momento de interactuar con la interfaz, el PC se presta para realizar cálculos numéricos necesarios para el desarrollo matemático que presente la cátedra, además de una excelente visualización de los resultados obtenidos de esta herramienta.

Para recrear un entorno gráfico de simulación es necesario revisar y comprender, concepto y herramientas que ayudan a tomar la mejor decisión al momento de cumplir con esta meta. de acuerdo con lo mencionado anteriormente el concepto de una GUI es primordial dado que como su nombre lo indica, Es una aplicación informática la cual crea un medio de interacción entre el usuario y el sistema operativo

de un dispositivo u ordenador, mediante un conjunto de imágenes, textos, botones, caja de texto, herramientas gráficas, entre otros, para permitir al usuario introducir o recibir información. Para ello se necesita de un lenguaje de programación, en Python existen librerías para este fin como son Tkinter, wxPython, PyQt y PyGTK, la diferencia entre ellas radica en los detalles visuales que una pueda tener sobre otra, las herramientas que puedan ofrecer, el apoyo que la comunidad preste a cada una de ellas, esto es importante al instante de hacer un desarrollo de este tipo, debido a que si no se tiene suficiente conocimiento del tema, es necesario buscar soporte o ayuda y si la librería usada es aceptada por un gran número de desarrolladores, será sencillo compartir experiencias [4].

Luego de comprender que una GUI es lo que se quiere implementar para tener un entorno gráfico donde el usuario pueda introducir datos y obtener resultados, sin duda es necesario el uso de un lenguaje de programación donde se pueda crear los códigos que darán vida a la aplicación. En virtud de lo requerido es necesario, comprender que es Python.

Se dice que Python es un lenguaje de programación de alto nivel, interpretado en otras palabras no necesita ser compilado, es multiplataforma lo cual indica que puede ser ejecutado en diferentes sistemas operativos. La filosofía de este lenguaje es realizar un código bastante legible, fácil de manejar y lo mejor es que es un lenguaje orientado a objetos lo cual crea un paradigma de resolución de problemas bastante sofisticado e interactivo [5]. Python cuenta con una licencia llamada "Python Software Foundation License." PSFL, la cual es una licencia de software libre [6].

Del lenguaje de programación mencionado anteriormente se destaca que debido a su paradigma de lenguaje orientado a objetos es recurrente el uso de librerías, también llamado módulos o paquetes, los cuales son objetos que sirven como contenedores de acciones o funciones lo cual ayuda a dividir el trabajo, hacer más legible nuestro código y más eficiente al momento de corregir errores, dado que en Python todo son objetos, entonces un módulo puede tener en su interior otros paquetes [7]. Algunas de las librerías más usadas cuando se requiere desarrollar entornos gráficos de visualización en el área científica sin duda son las que se detallan a continuación:

Numpy

Este Modulo de Python es fundamental para realizar cálculos científicos, ya que brinda mayor soporte para el trabajo con vectores y matrices, lo cual crea un entorno matemático de alto nivel para lograr realizar las operaciones y representaciones matemáticas que se requieren en esta área del conocimiento. Ofreciendo, sofisticadas funciones de radiodifusión, herramientas para la integración de C/C++ y Fortran, transformada de Fourier [8].

Matplotlib

Es una librería que permite la generación de gráficas a partir de una lista o arreglos generados con la extensión Numpy de Python. las figuras resultantes de este paquete son de muy buena calidad lo que hace que sea ideal para el desarrollo de publicaciones en artículos relacionados al área de la ciencia. Mientras que para una GUI este módulo se acopla perfectamente y así crear un entorno en el cual se pueda visualizar resultados matemáticos con muy buena presencia [9].

PyQt

Es una biblioteca gráfica para Python, la cual contiene un grupo de herramientas útiles a la hora de desarrollar aplicaciones con interfaz gráfica, dado que ofrece un entorno, con los elementos necesarios para realizar un programa destinado a lo científico o académico, de una manera bastante interactiva. PyQt tiene doble licencia en todas las plataformas soportadas bajo la GNU GPL v3 y la Licencia Comercial Riverbank [10].

Capítulo III

Procedimientos de la investigación

3.1. Exploración Bibliográfica

En esta etapa del trabajo se recopiló la información acerca de sistemas de ondas guiadas y líneas de transmisión. Como fuente bibliográfica principal se tomó el libro de la asignatura Sistemas de Ondas Guiadas de Aldo Bianchi [2], mientras que para añadir los nuevos temas al programa se utilizaron otros textos siendo el texto de Ingeniería de Microondas desarrollado por David Pozar [11], una de las bibliografías de mayor referencia en el área, de igual forma el resto de la información fue revisada y seleccionada para añadir las nuevas herramientas al software.

De acuerdo con la bibliografía seleccionada, se realizó un análisis de este material relacionado con la asignatura, a fin de tener mayor conocimiento de la materia y realizar los módulos del programa que están dedicados a representar los tópicos vistos en clase de manera sencilla y ajustado a lo impartido por el profesor.

3.2. Selección de paquetes de Python apropiados

Posterior a la recopilación y estudio de la información necesaria para el desarrollo de la herramienta, además de establecido los contenidos de Sistemas de Ondas

Guiadas, fue indispensable documentarse y desarrollar habilidades de programación enfocada en Python debido a que la versión anterior de SEDISOG fue realizada en este lenguaje de programación y el cual hasta la actualidad presenta buen rendimiento, por ende cambiar de lenguaje no es necesario para adaptar las mejoras en esta nueva versión.

Debe señalarse que obligados a cumplir con los requerimientos impuestos en esta aplicación fue necesario buscar paquetes o librerías que cumplieran con todo lo previsto, tal como, resolución de arreglos matemáticos, una herramienta para visualizar gráficas, crear una interfaz de usuario actual y sencilla, entre otros requisitos. Cabe destacar que al momento de elegir un paquete este debe ser apoyado por una comunidad suficientemente grande para tener la posibilidad de retroalimentación con esta, y así obtener mejores resultados en menor tiempo.

A continuación se presenta el proceso de selección de paquetes dependiendo del requisito:

3.2.1. Paquete de cálculos numéricos

Python por defecto contiene una librería llamada math, la cual incorpora herramientas para el desarrollo matemático básico pero a la hora de incursionar en los arreglos de tipo matriciales este paquete deja de ser funcional, y dado al contenido matemático que describe a la asignatura Sistemas de Ondas Guiadas, es necesario apoyarse en este tipo de arreglos, para ello existe un módulo que no es nativo de python pero se puede instalar fácilmente a nuestro interprete, este se llama Numpy.

Numpy Ofrece poderosas herramientas para cálculo numérico aplicado a la ciencia, además ofrece una documentación bastante clara para observar que contiene este paquete y es soportada por una comunidad amplia la cual brinda constante ayuda con respecto al tema.

3.2.2. Paquete para realizar y mostrar gráficos

En esta aplicación el uso de gráficos es sumamente importante ya que gracias a estos es posible analizar y mostrar al usuario el comportamiento de un fenómeno, mediante una curva con sus respectivo sistema de coordenadas y detalles que identifiquen el origen de lo mostrado, y así mismo que las figuras generadas puedan ser almacenadas en diferentes formatos para su posterior estudio. Matplotlib es sin duda el generador y asistente de gráficos que ofrece al usuario una cantidad de opciones cuando es necesario representar contenido matemático en forma de gráficas, con facilidad y gran contenido en detalles para hacer de estas lo suficientemente entendible al observador final.

3.2.3. Paquete para crear una interfaz gráfica de usuario (GUI)

Al momento de crear una interfaz gráfica de usuario es necesario tener los objetivos y el aspecto de la aplicación bastante definido ya que con esa idea, se obtiene el criterio de elección del paquete para crear una GUI que cumpla con lo planeado, de lo contrario al surgir una nueva idea o mejora que ocurra después de comenzado el proyecto, y la librería no soporta el nuevo planteamiento puede llegar a ocurrir un conflicto entre decisiones aisladas.

Debido a lo expuesto anteriormente, antes de elegir que paquete seria el apropiado para los requerimientos del programa se fijaron los desafíos que este debía satisfacer, tal como, facilidad para combinarse con gráficas, manejo de arrastrar y soltar objetos, un aspecto visual mas actual, también, que el código del software sea de lectura sencilla para posteriores aportes y ademas que sea de fácil acceso a la informacion para disminuir el tiempo de culminación del programa.

Existen diversas librerías para la creación de este tipo de interfaces tal como:

- Tkinter
- WxPython

- PyQT
- Pygame
- PyGTK

De esta lista, Tkinter fue descartado debido a que a pesar de estar pre instalado en python para casi todas las plataformas este ofrece pocos elementos gráficos y presenta una apariencia poco habitual. Mientras que el paquete WxPython es mas complejo de aprender y para proyectos de este tipo suele ser inestable y complicado de depurar.

PyGTK ofrece un grupo gráfico bastante completo pero es relativamente lento para dibujar los elementos del GUI, lo que causa una apariencia incomoda al usuario, lo que hace que quede descartada de la lista. En nuestra lista nos quedo por detallar entre Pygame y PyQT, siendo la primera, una librería que dio estructura visual a la versión anterior de SEDISOG, lo que significa un apoyo significativo a la hora de iniciar mejoras, solo que esta no ofrecía herramientas que satisficieran los nuevos desafíos establecidos, por consiguiente se decidió emigrar al módulo PyQT, ya que este ofrece un gran contenido gráfico para recrear los objetivos planteados, ademas tiene una documentación abundante en información.

PyQt Ofrece la opción de utilizar una herramienta que permite crear ventanas con todos sus elementos (Botones, etiquetas, cajas de texto, entre otros), de una forma intuitiva con solo arrastrar y soltar los componentes de la interfaz gráfica, lo que permite realizar un trabajo con mejor presencia y en menor tiempo. Esta herramienta se llama Qt Designer es totalmente libre y existe suficiente documentación al respecto, para nuevos y mas experimentados desarrolladores en el área.

3.3. Investigación y estudio de python y paquetes seleccionados

En esta etapa de la investigación se recopilo la informacion necesaria acerca del lenguaje de programación como pilar fundamental al momento de comprender la

sintaxis y paradigma que enfoca python para abordar soluciones, para ello se uso textos y fuentes bibliográficas existentes en Internet, también la consulta de algunos foros dedicados al entorno.

Luego de seleccionado los paquetes necesarios se procedió al estudio de estos, mediante la lectura, de la documentación que ofrecen los creadores de estos módulos y ademas, del uso de tutoriales ubicados en la WEB.

3.4. Estudio y visualización de los transitorios presente en un arreglo de líneas de transmisión mediante el uso de FDTD

Luego de familiarizarse con el lenguaje de programación y los paquetes seleccionados para el desarrollo de la aplicación, se invirtió esfuerzo en buscar información referente al método de las "Diferencias Finitas en el Dominio del Tiempo" (FDTD), el cual es implementado para modelar y encontrar solución a problemas electromagnéticos en el dominio del tiempo tanto en una, dos y tres dimensiones, mediante un proceso de discretización de las derivadas parciales presente en las ecuaciones que describen el fenómeno.

Con la información recopilada, el siguiente paso sería estudiar y sintetizar el método de FDTD, la cual se centro en el estudio para una sola dimensión, dado que para representar los transitorios en la línea de transmisión se utiliza este procedimiento, ya que las ecuaciones de los telegrafistas, que describen el modelo matemático de la línea, contiene derivadas parciales que desarrollando la teoría de Yee, se pueden discretizar, quedando expresiones matemáticas que una herramienta de computo sí puede manejar, debido a que la cantidad de muestras necesarias para representar el proceso transitorio en la línea, pasó de ser infinita a finita.

Las ecuaciones ya discretizadas describen el comportamiento del voltaje y la corriente presente en la línea de transmisión, tomando en cuenta las diferentes condiciones de borde que pueden surgir, como una carga que puede ser netamente re-

sistiva (R), una carga (RL) serie o paralelo, (RC) serie o paralelo, la resistencia del generador (Rg) o el tipo de excitación, además de los parámetros intrínsecos de la línea (r , l , c y g) y algunas discontinuidad que pueda presente entre el generador y la carga como puede ser el caso de una resistencia en serie o paralelo, dos o más líneas de diferentes propiedades conectadas en paralelo, entre otros.

Con el objetivo de observar la evolución temporal de la tensión y corriente en la Línea utilizando este método, es necesario crear una estructura o malla como se muestra en la figura 3.1, la cual señala de forma intuitiva el resultado del proceso de discretización que se mencionó anteriormente, observando una discretización espacial y otra temporal, tanto para V e I , de esta manera, se puede notar la disposición temporal y espacial de V e I cuando apliquemos el algoritmo. El proceso para conocer la evolución V se obtiene resolviendo las ecuaciones de los telegrafistas en todas las muestras espaciales para cada instante de tiempo usando valores almacenados de corriente del tiempo anterior, un procedimiento similar se hace para el cálculo de la corriente solo que para esta se utilizan valores almacenados de voltaje del instante de tiempo anterior. Mediante recursividad que existe entre V e I , se presenta la evolución temporal de la tensión y corriente en una línea de transmisión, el proceso culmina cuando el espaciado temporal se termina.

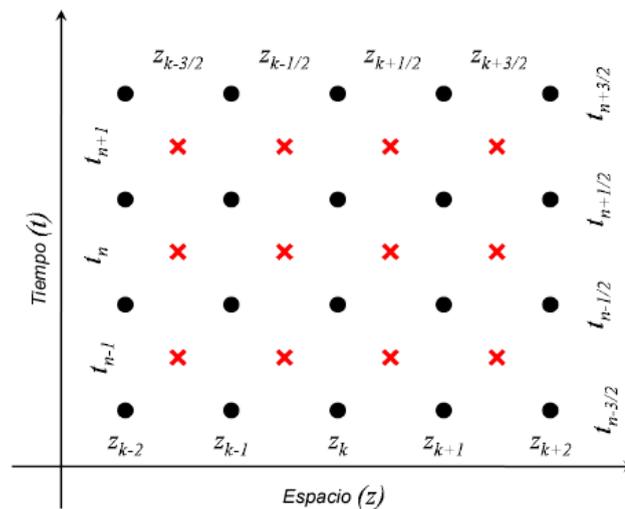


Figura 3.1: Disposición temporal y espacial del Voltaje y Corriente, siendo las X's rojas la tensión y los círculos negros la corriente. (Fuente [12])

Comprendido el procedimiento para el cálculo de la tensión y corriente a lo largo de la línea de transmisión, se implemento el modelo de parámetros distribuidos en una línea de transmisión, del cual se obtienen las ecuaciones que describen el comportamiento de V e I en cada punto de la línea al aplicar Ley de corrientes y tensiones de Kirchoff en cada celda del modelo, además el ancho Δz debe cumplir con, $\Delta z \ll \lambda$. en las expresiones 3.1 y 3.2 se muestra la relación tensión y corriente.

$$-\frac{\partial i(z, t)}{\partial z} = gv(z, t) + c \frac{\partial v(z, t)}{\partial t} \quad (3.1)$$

$$-\frac{\partial v(z, t)}{\partial z} = ri(z, t) + l \frac{\partial i(z, t)}{\partial t} \quad (3.2)$$

las expresiones 3.1 y 3.2 se deben discretizar, para ello se emplearon aproximaciones usadas en [13]. De este modo una derivada parcial puede aproximarse a:

$$\frac{\partial v(z, t)}{\partial z} = \frac{\partial v_k^n}{\partial z} \approx \frac{v_{k+1/2}^n - v_{k-1/2}^n}{\Delta z} \quad (3.3)$$

$$\frac{\partial i(z, t)}{\partial t} = \frac{\partial i_k^n}{\partial t} \approx \frac{i_k^{n+1/2} - i_k^{n-1/2}}{\Delta t} \quad (3.4)$$

En las ecuaciones 3.3 y 3.4 se refleja el resultado de la aproximación aplicada a las derivadas parciales respecto a la posición y el tiempo respectivamente. Aplicando estas aproximaciones a las ecuaciones de los telegrafistas nos queda:

$$\frac{-(i_{k+1/2}^n - i_{k-1/2}^n)}{\Delta z} = g \frac{(v_k^{n+1/2} + v_k^{n-1/2})}{2} + c \frac{(v_k^{n+1/2} - v_k^{n-1/2})}{\Delta t} \quad (3.5)$$

$$\frac{-(v_{k+1}^{n+1/2} - i_k^{n-1/2})}{\Delta z} = r \frac{(i_{k+1/2}^{n+1} + i_{k+1/2}^n)}{2} + l \frac{(i_{k+1/2}^{n+1} - v_{k+1/2}^n)}{\Delta t} \quad (3.6)$$

Al operar matemáticamente los términos de las expresiones 3.5 y 3.6, se obtiene las expresiones 3.7 y 3.8, las cuales representan las ecuaciones que actualizan los

valores de tensión y corriente a lo largo de la línea.

$$v_k^{n+1/2} = \frac{\left(\frac{c}{\Delta t} - \frac{g}{2}\right)}{\left(\frac{c}{\Delta t} + \frac{g}{2}\right)} v_k^{n-1/2} - \frac{1}{\left(\frac{c}{\Delta t} + \frac{g}{2}\right)} \frac{(i_{k+1/2}^{n+1} - i_{k-1/2}^n)}{\Delta z} \quad (3.7)$$

$$i_{k+1/2}^{n+1} = \frac{\left(\frac{l}{\Delta t} - \frac{r}{2}\right)}{\left(\frac{l}{\Delta t} + \frac{r}{2}\right)} i_{k+1/2}^n - \frac{1}{\left(\frac{l}{\Delta t} + \frac{r}{2}\right)} \frac{(v_{k+1}^{n+1/2} - v_k^{n+1/2})}{\Delta z} \quad (3.8)$$

Al momento de iniciar el proceso para calcular Voltaje y Corriente es necesario, establecer las condiciones de borde presente en el arreglo que se pretende simular, tal como, la condición de borde que ejerce el generador, la carga, cualquier discontinuidad que se encuentre en la línea de transmisión, además cualquier tipo de conexión entre 2 o mas líneas, entre otros. Para ello es necesario realizar las operaciones que siguen a continuación:

Condición de borde en el Extremo del Generador ($k=1$)

Para obtener la expresión de actualización de tensión en el extremo del generador, se debe aplicar el equivalente de Norton al generador y acoplado con el modelo de parámetros distribuidos para la línea de transmisión mostrado en la figura 2.2, el esquema resultante puede observarse en la figura 3.2.

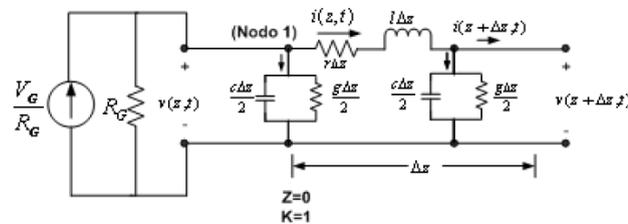


Figura 3.2: Equivalente de Norton aplicado al generador y primera celda del modelo de la línea de transmisión. (Fuente [3])

Al aplicar la ley de corrientes de Kirchoff en el nodo 1 ($k=1$) mostrado en la figura 3.2, se obtiene la siguiente expresión.

$$\frac{v_g}{r_g} = i_1^n + \left(\frac{1}{r_g} + \frac{g\Delta x}{2}\right)v_1^n + \frac{c\Delta x}{2} \frac{dv_1^n}{dt} \quad (3.9)$$

Al discretizar la ecuación, se obtiene:

$$\frac{(v_g^{n+1/2} + v_g^{n-1/2})}{2r_g} = i_1^n + \left(\frac{1}{r_g} + \frac{g\Delta x}{2}\right)(v_1^{n+1/2} + v_1^{n-1/2}) + \frac{c\Delta x}{2} \frac{(v_1^{n+1/2} - v_1^{n-1/2})}{\Delta t} \quad (3.10)$$

Reordenando los términos de la ecuación 3.10, se encuentra una expresión para la actualización de voltaje en la celda del generador ($k=1$), viene dada por:

$$v_1^{n+1/2} = \frac{\left[r_g \frac{c\Delta x}{2\Delta t} - \frac{1}{2} \left(1 + r_g \frac{g\Delta x}{2}\right)\right]}{\left[r_g \frac{c\Delta x}{2\Delta t} + \frac{1}{2} \left(1 + r_g \frac{g\Delta x}{2}\right)\right]} v_1^{n-1/2} + \frac{1}{\left[r_g \frac{c\Delta x}{2\Delta t} + \frac{1}{2} \left(1 + r_g \frac{g\Delta x}{2}\right)\right]} \left[\frac{(v_g^{n+1/2} + v_g^{n-1/2})}{2} - r_g i_1^n \right] \quad (3.11)$$

Condición de borde en el Extremo de la Carga ($k=N$)

Las condiciones de borde en este caso, dependerá del tipo de carga que conectemos por lo que el valor V e I en el extremo de la carga sera diferente para cada caso. En la figura 3.3 se muestra el esquemático utilizado para desarrollar las expresiones de actualización para voltaje o corriente en la ultima celda del modelo discretizado de la línea.

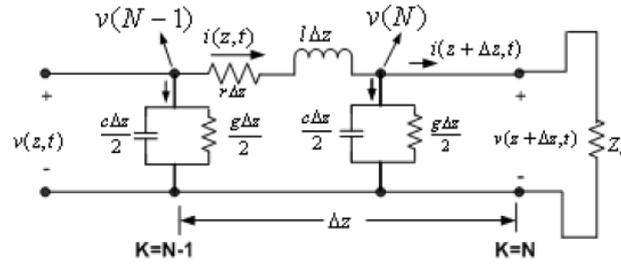


Figura 3.3: Modelo usado para la ultima celda, conectando una carga Z_L en el extremo de la línea (Fuente [3])

Carga Resistiva Pura (R_L)

Usando el modelo mostrado en la figura 3.3 y aplicando la ley de corriente de Kirchhoff en el nodo N obtenemos:

$$i_{N-1}^{n+1/2} = \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) v_N^n + \frac{c\Delta x}{2} \frac{dv_N^n}{dt} \quad (3.12)$$

al discretizar la expresión anterior:

$$i_{N-1}^{n+1/2} = \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \frac{(v_N^{n+1/2} + v_N^{n-1/2})}{2} + \frac{c\Delta x}{2} \frac{(v_N^{n+1/2} - v_N^{n-1/2})}{\Delta t} \quad (3.13)$$

agrupando y reordenando los términos de la expresión anterior, se tiene:

$$v_N^{n+1/2} = \left[\frac{\frac{c\Delta x}{2\Delta t} - \frac{1}{2} \left(\frac{1}{R_L} + \frac{g\Delta x}{2} \right)}{\frac{c\Delta x}{2\Delta t} + \frac{1}{2} \left(\frac{1}{R_L} + \frac{g\Delta x}{2} \right)} \right] v_N^{n-1/2} + \left[\frac{1}{\frac{c\Delta x}{2\Delta t} + \frac{1}{2} \left(\frac{1}{R_L} + \frac{g\Delta x}{2} \right)} \right] i_{N-1}^{n+1/2} \quad (3.14)$$

La ecuación 3.14 permite calcular el voltaje en la Última celda, donde se encuentra la carga resistiva pura.

Carga $R_L \parallel L_L$

Al utilizar el modelo que se observa en la figura 3.3 para el cual se hace la sumatoria de corriente en el nodo N, y se obtiene la siguiente expresión:

$$i_{N-1}^{n+1/2} = \frac{g\Delta x}{2} v_N^{n+1/2} + \frac{c\Delta x}{2} \frac{dv_N^{n+1/2}}{dt} + \frac{1}{R_L} v_N^{n+1/2} + i_l \quad (3.15)$$

al discretizar la expresión anterior, se obtiene:

$$i_{N-1}^{n+1/2} = \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \left[\frac{v_N^{n+1/2} + v_N^{n-1/2}}{2} \right] + \frac{c\Delta x}{2} \left[\frac{v_N^{n+1/2} - v_N^{n-1/2}}{\Delta t} \right] + i_l \quad (3.16)$$

agrupando y reordenando términos de la expresión anterior, se obtiene:

$$v_N^{n+1/2} = \frac{\left[\frac{c\Delta x}{2} - \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \right]}{\left[\frac{c\Delta x}{2} + \frac{g\Delta x}{2} + \frac{1}{R_L} \right]} v_N^{n-1/2} + \frac{1}{\left[\frac{c\Delta x}{2} + \frac{g\Delta x}{2} + \frac{1}{R_L} \right]} \left(i_{N-1}^{n+1/2} - i_l \right) \quad (3.17)$$

de la expresión anterior se desconoce el valor de i_l , pero se sabe que:

$$i_l = \frac{1}{l_L} \int_{-\infty}^{\infty} v_N^{n+1/2} dt \quad (3.18)$$

al discretizar se tiene:

$$i_l = \frac{1}{l_L} \sum_{-\infty}^{n-1} v_N^{n+1/2} \Delta t \quad (3.19)$$

la ecuación anterior representa la corriente en el inductor que es proporcional al voltaje acumulado entre los terminales del inductor durante todo el evento.

Por medio de la ecuación 3.17 y con apoyo de la ecuación 3.19, se calcula el valor de voltaje presente en la ultima celda cuando la carga es una resistencia con un inductor dispuestos en paralelo.

Carga $R_L - L_L$

Para este caso de igual forma se utiliza el modelo del caso anterior, pero se aplica la ley de voltajes de Kirchhoff en la ultima celda, así obtener la siguiente expresión:

$$v_{N-1}^{n+1/2} = (r\Delta x) i_{N-1}^{n+1/2} + l\Delta x \frac{di_{N-1}^{n+1/2}}{dt} + R_L i_N^{n+1/2} + L_L \frac{di_N^{n+1/2}}{dt} \quad (3.20)$$

al discretizar la expresión anterior, se tiene:

$$v_{N-1}^{n+1/2} = r\Delta x \frac{i_{N-1}^{n+1/2} + i_{N-1}^{n-1/2}}{2} + l\Delta x \frac{i_{N-1}^{n+1/2} - i_{N-1}^{n-1/2}}{\Delta t} + R_L \frac{i_N^{n+1/2} + i_N^{n-1/2}}{2} + L_L \frac{i_N^{n+1/2} - i_N^{n-1/2}}{\Delta t} \quad (3.21)$$

reordenando y agrupando términos de la ecuación 3.21, se tiene:

$$i_N^{n+1/2} = \frac{1}{\left[\frac{L_L}{\Delta t} + \frac{R_L}{2}\right]} v_{N-1}^{n+1/2} + \frac{\left[\frac{l\Delta t}{\Delta t} - \frac{r\Delta t}{2}\right]}{\left[\frac{L_L}{\Delta t} + \frac{R_L}{2}\right]} i_{N-1}^{n-1/2} + \frac{\left[\frac{L_L}{\Delta t} - \frac{R_L}{2}\right]}{\left[\frac{L_L}{\Delta t} + \frac{R_L}{2}\right]} i_N^{n-1/2} - \frac{\left[\frac{l\Delta t}{\Delta t} + \frac{r\Delta t}{2}\right]}{\left[\frac{L_L}{\Delta t} + \frac{R_L}{2}\right]} i_{N-1}^{n+1/2} \quad (3.22)$$

La ecuación 3.22 permite calcular el valor de corriente existente en la ultima celda cuando la carga es un capacitor en serie con una resistencia.

Carga $R_L - C_L$

Basado en el esquemático que se viene trabajando y aplicando una sumatoria de corrientes en el nodo N, se tiene:

$$i_{N-1}^{n+1/2} = \frac{g\Delta x}{2} v_N^{n+1/2} + \frac{c\Delta x}{2} \frac{dv_N^{n+1/2}}{dt} + i_N^{n+1/2} \quad (3.23)$$

al discretizar la expresión anterior, queda:

$$i_{N-1}^{n+1/2} = \frac{g\Delta x}{2} \frac{(v_N^{n+1/2} + v_N^{n-1/2})}{2} + \frac{c\Delta x}{2} \left[\frac{v_N^{n+1/2} - v_N^{n-1/2}}{\Delta t} \right] + i_N^{n+1/2} \quad (3.24)$$

ordenando y reagrupando términos de la expresión anterior, se obtiene:

$$v_N^{n+1/2} = \frac{\left[\frac{c\Delta x}{\Delta t} - \frac{g\Delta x}{2} \right]}{\left[\frac{c\Delta x}{\Delta t} + \frac{g\Delta x}{2} \right]} v_N^{n-1/2} + \frac{1}{\left[\frac{c\Delta x}{\Delta t} + \frac{g\Delta x}{2} \right]} \left[i_{N-1}^{n+1/2} - i_N^{n+1/2} \right] \quad (3.25)$$

De la ecuación anterior se desconoce el valor de la corriente en la carga para ellos se aplica lo siguiente:

$$i_N^{n+1/2} = i_{c_L}^{n+1/2} = C_L \frac{dv_{c_L}^{n+1/2}}{dt} \quad (3.26)$$

al discretizar la ecuación 3.26, se obtiene:

$$i_N^{n+1/2} = i_{c_L}^{n+1/2} = c_L \frac{v_{c_L}^{n+1/2} - v_{c_L}^{n-1/2}}{\Delta t} \quad (3.27)$$

El voltaje del capacitor necesario en la ecuación 3.27, se obtiene al aplicarse lo siguiente:

$$v_{c_L}^{n+1/2} = \frac{1}{c_L} \int_{-\infty}^{\infty} i_{c_L}^{n+1/2} dt \quad (3.28)$$

al discretizar la expresión anterior, se tiene:

$$v_{c_L} = \frac{1}{c_L} \sum_0^{n-1} i_{c_L}^{n+1/2} \cdot \Delta t \quad (3.29)$$

con la expresión anterior se obtiene el valor de voltaje acumulado en el capacitor para cada instante de tiempo, el cual sera necesario para conocer la corriente del mismo, que a su vez es la corriente presente en la carga, como se observa en la ecuación 3.27.

La corriente en la ultima celda es importante conocerla porque se usa en la ecuación 3.25 que permite calcular el valor del voltaje en esta celda para todo tiempo cuando se tiene como carga un resistor con capacitor en serie.

Carga $R_L || C_L$

Para este caso se aplica la ley de voltaje de Kirchhoff al modelo mostrado en la figura 3.3 para obtener la siguiente ecuación:

$$i_{N-1}^{n+1/2} = \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) v_N^{n+1/2} + \left(\frac{c\Delta x}{2} + C_L \right) \frac{dv_N^{n+1/2}}{dt} \quad (3.30)$$

al discretizar la expresión anterior, se obtiene:

$$i_{N-1}^{n+1/2} = \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \frac{(v_N^{n+1/2} + v_N^{n-1/2})}{2} + \left(\frac{c\Delta x}{2} + C_L \right) \frac{v_N^{n+1/2} - v_N^{n-1/2}}{\Delta t} \quad (3.31)$$

ordenando y reagrupando terminos de la expresión anterior, se tiene:

$$v_N^{n+1/2} = \frac{\left[\frac{c\Delta x + 2C_L}{2\Delta t} - \frac{1}{2} \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \right]}{\left[\frac{c\Delta x + 2C_L}{2\Delta t} + \frac{1}{2} \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \right]} v_N^{n-1/2} + \frac{1}{\left[\frac{c\Delta x + 2C_L}{2\Delta t} + \frac{1}{2} \left(\frac{g\Delta x}{2} + \frac{1}{R_L} \right) \right]} i_{N-1}^{n+1/2} \quad (3.32)$$

La ecuación 3.38 indica el comportamiento del voltaje en la ultima celda, donde se encuentra la carga cuando esta este conformada por un resistor y capacitor dispuestos en paralelo.

Condición de borde debido a conexión de dos o más líneas de transmisión en paralelo

Cuando se conectan varias líneas en paralelo estas establecen una discontinuidad en el plano de conexión como se muestra en la figura 3.4, en este caso se dispone de tres líneas en paralelo sin pérdidas dado que no existe resistencia y conductancia por unidad de longitud, de igual forma se puede obtener una expresión que represente la actualización de voltaje y corriente en el plano de conexión.

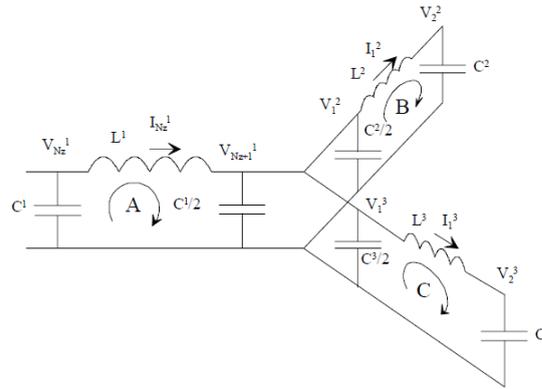


Figura 3.4: Modelo equivalente para la conexión de tres líneas de transmisión en paralelo. (Fuente [14])

Para ello se aplica la ley de voltaje de Kirchhoff en los lazos A, B y C que se muestran en la figura 3.4, para obtener lo siguiente:

$$\left(i_N^{n+1/2}\right)_1 = \left(i_N^{n-1/2}\right)_1 - \frac{\Delta t}{L_1 \Delta x} \left[\left(v_2^{n+1/2}\right)_1 - \left(v_1^{n+1/2}\right)_1 \right] \quad (3.33)$$

$$\left(i_1^{n+1/2}\right)_i = \left(i_1^{n-1/2}\right)_i - \frac{\Delta t}{L_i \Delta x} \left[\left(v_2^{n+1/2}\right)_i - \left(v_1^{n+1/2}\right)_i \right], \quad i = 2, 3 \quad (3.34)$$

La expresión 3.33 representa la corriente en la última celda para la línea 1, para ellos se recorre el lazo A, Mientras que la ecuación 3.34 indica la corriente en la línea 2 y 3, al recorrer los lazos B y C respectivamente se obtienen dichas expresiones.

Al aplicar la ley de corriente de Kirchhoff en el plano de encuentro, además de ordenar y agrupar términos se obtiene la siguiente expresión:

$$\begin{aligned} \left(v_1^{n+1/2}\right)_{2,3} = \left(v_N^{n+1/2}\right)_1 = \left(v_N^{n-1/2}\right)_1 - \\ - \frac{2\Delta t}{(C_1 + C_2 + C_3) \Delta x} \left[\left(i_1^{n+1/2}\right)_2 + \left(i_1^{n+1/2}\right)_3 - \left(i_N^{n+1/2}\right)_1 \right] \end{aligned} \quad (3.35)$$

La expresión anterior indica el valor del voltaje existente en el plano de discontinuidad para las líneas involucradas, como se puede observar es el mismo para cada una de ellas.

Con las expresiones de actualización, de tensión y corriente en cada segmento de línea y además en el plano de discontinuidad, es posible conocer como es la propagación de energía en un arreglo de líneas de transmisión para cada instante de tiempo.

3.5. Diseño y estructura de la Aplicación

Este programa está estructurado por varios módulos los cuales se detallan a continuación:

3.5.1. Módulo para la representación de transitorios en un arreglo de líneas de transmisión

Al tener todas las ecuaciones de actualización de voltaje (V) y corriente (I), para las distintas condiciones de borde existentes a lo largo de la línea de transmisión, fue necesario traducir cada expresión obtenida, al lenguaje de programación Python.

Para representar los transitorios presentes en un arreglo de líneas de transmisión se dividió en dos modelos de estudio, arreglos lineales y el arreglo de tres líneas dispuestas en paralelo, como se muestra en la figura 3.5, para luego crear los algoritmos necesarios en cada caso.

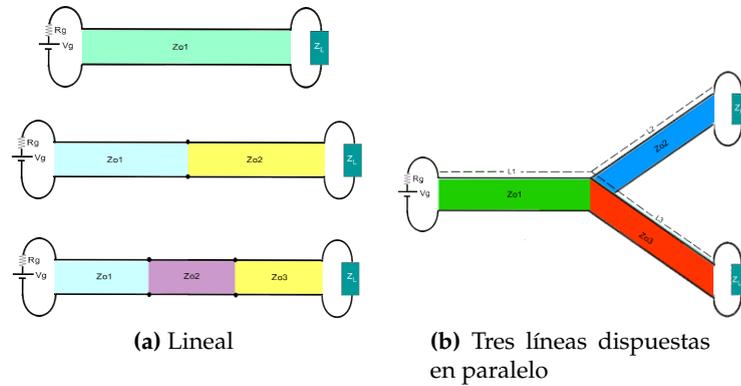


Figura 3.5: Arreglos de líneas de transmisión

El algoritmo realizado para simular y visualizar los transitorios en arreglos lineales se muestra en la figura 3.6, mediante un diagrama de flujo que permite obtener los valores de (V) e (I) para cualquier tiempo y posición de la línea.

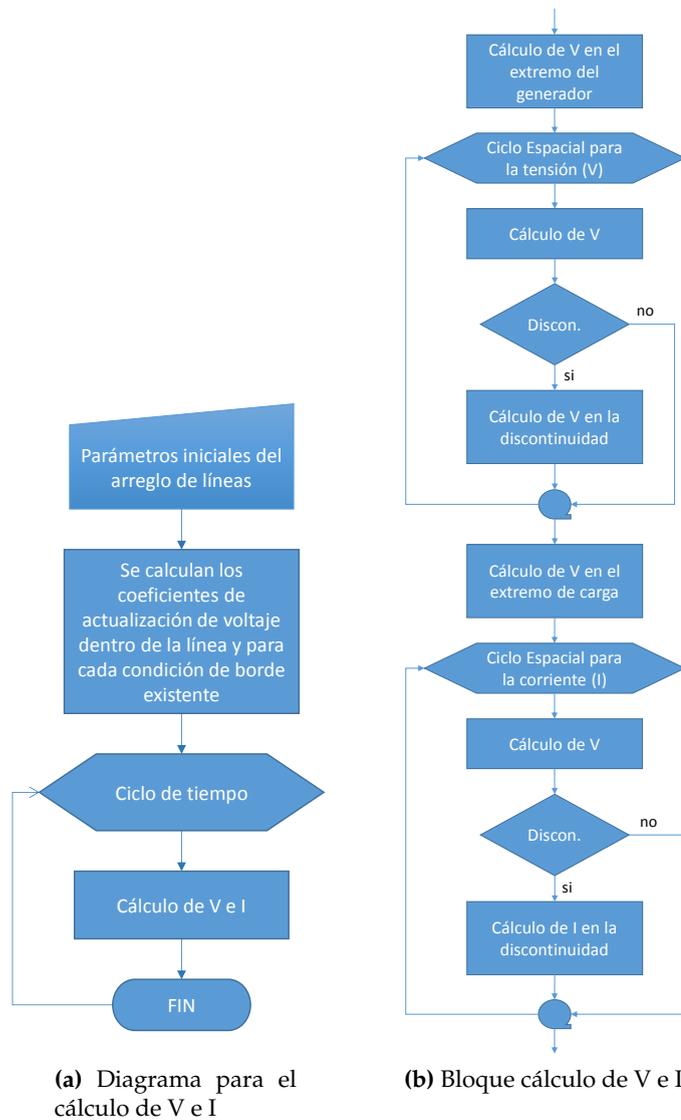


Figura 3.6: Diagramas de Flujo para transitorios en arreglos lineales

En la figura 3.6a se muestra el procedimiento para calcular el voltaje y corriente en un arreglo lineal de líneas de transmisión. Mientras que la figura 3.6b indica el proceso que realiza el bloque "Cálculo de V e I" de la figura 3.6a.

Asimismo para simular el caso del arreglo de tres líneas dispuestas en paralelo se generó un algoritmo que se muestra en forma de diagrama de flujo, en la figura 3.7.

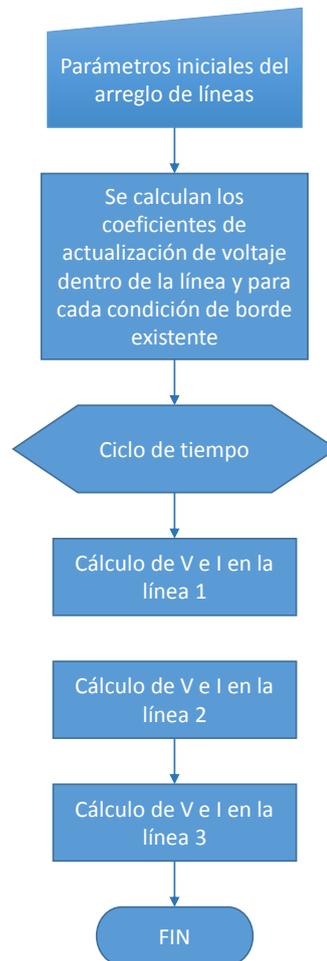
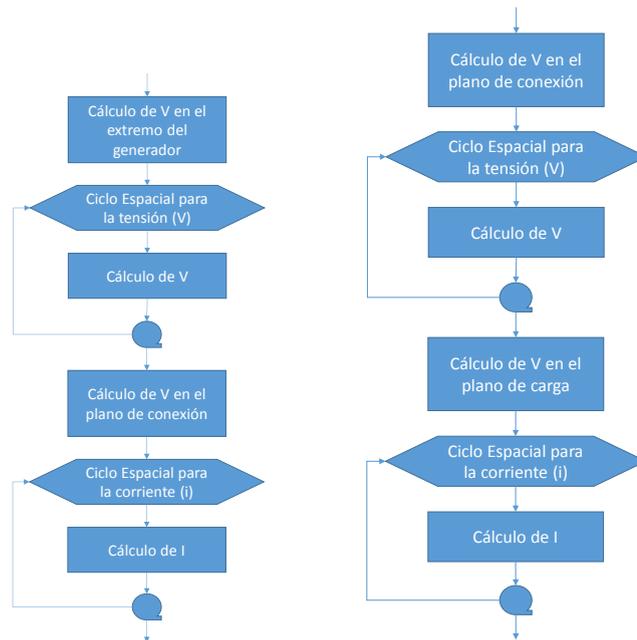


Figura 3.7: Diagrama para el cálculo de V e I en tres líneas dispuestas en paralelo

De la figura anterior se debe observar el procedimiento que realizan los bloques para el cálculo de tensión y corriente en cada línea, para ello se mostraron en la figura 3.8, además se debe destacar que la línea uno es la que se encuentra a la izquierda y las líneas 2 y 3 a la derecha, esto se puede observar en la figura 3.5b



(a) Bloque cálculo de V e I para la línea 1

(b) Bloque cálculo de V e I para las líneas 2 y 3

Figura 3.8: Diagramas de Flujo de los bloques para calcular de V e I en cada línea

Luego de generar los algoritmos para el proceso de simulación y visualización de los transitorios en un arreglo de líneas de transmisión, se generó la interfaz que permite dicha visualización, para ello se realizó el siguiente procedimiento:

- Identificación de los parámetros de entrada.
- Identificación del tipo de arreglo.
- Identificación de los parámetros de salida.
- Identificación de las ecuaciones que rigen el comportamiento de la línea, tomando en cuenta las diferentes condiciones de borde, expuestas anteriormente.
- Identificación de las condiciones de borde para la creación del diagrama de reflexiones.

- En el lenguaje Python:
 - Se generaron los campos de entrada para que el usuario introduzca o modifique parámetros.
 - Se generaron los campos para seleccionar el tipo de arreglo a simular, las discontinuidades a ubicar y seleccionar el tipo de fuente.
 - Se generaron los botones para navegar dentro de la herramienta, ir o retornar a algún menú del programa.
 - Se empleó el método de la FDTD en una dimensión para obtener los valores de tensión y corriente en cualquier punto de la línea y en cualquier instante de tiempo.
 - Se generó un algoritmo para realizar el diagrama de reflexiones
 - se generó un algoritmo que permite visualizar el proceso de evolución de la tensión y corriente en el arreglo, de forma animada.

El algoritmo que realiza el diagrama de reflexiones se muestra mediante el siguiente diagrama de flujo:

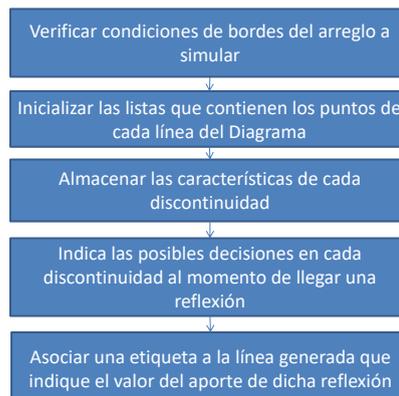


Figura 3.9: Diagrama de flujo para la creación del diagrama de reflexiones

3.5.2. Módulos para análisis del régimen permanente existente en la línea de transmisión

Para el estudio del régimen permanente se añadieron mejoras a los ya existentes y se crearon nuevos módulos para mejorar el proceso de aprendizaje en el comportamiento de la línea de transmisión en régimen permanente.

Los módulos se detallan a continuación:

3.5.2.1. Módulo de Relación de Onda Estacionaria (ROE)

Para crear este modulo se realizó el siguiente procedimiento:

- Generar una zona de gráficas donde se mostrará el comportamiento de los parámetros de salida.
- Identificar las ecuaciones que deben ser usadas para obtener los parámetros de salida.
- Identificar los parámetros de entrada.
- Identificar los parámetros de salida.
- En el lenguaje Python:
 - Se Generó un menú con campos de entrada donde el usuario podrá introducir o modificar parámetros.
 - Se creó los botones de navegación que permite al usuario ir a los diferentes menús.
 - Se generó un algoritmo que permite la animación de los parámetros de salida.
 - Se generó un algoritmo para ubicar el valor de los parámetros de salida en un punto de la línea.
 - Se generó un algoritmo que permite animar los parámetros de salida en función de la frecuencia.

- Se generó etiquetas para mostrar los resultados.

En la figura 3.10 se muestra el algoritmo que permite obtener los parámetros de salida: V, I, P, Z , y SWR , tomando como parámetros de entrada: Z_0, Z_L , la longitud (L) y la atenuación total en la línea (At).

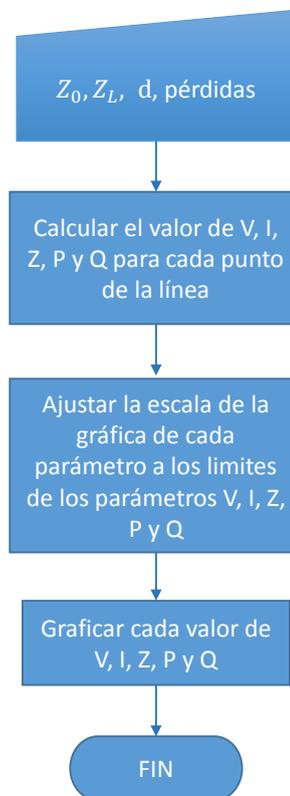


Figura 3.10: Diagrama de flujo para el cálculo de los parámetros de salida del módulo ROE

3.5.2.2. Módulo de adaptación de impedancia mediante una línea terminada en cortocircuito dispuesta en paralelo

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
 - Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
 - Se creó el botón para mostrar por pantalla los resultados.
 - Se creó una pareja de botones de opción para escoger que longitudes manejara el sistema si las de la solución 1 o la solución 2.
 - Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
 - Se generaron los algoritmos que realizan los cálculos para construir las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión.
 - Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

En las figuras 3.11 y 3.12 se muestra el algoritmo seguido para calcular las longitudes características del sistema, así como para dibujar las curvas patrón de onda estacionaria y Respuesta en frecuencia del módulo del coeficiente de reflexión. Los datos de entrada son: impedancia de carga (Z_L), frecuencia de diseño (f), ancho de banda (BW), impedancia característica de la línea (Z_0), impedancia característica del stub (Z_{01}), velocidad de propagación en la línea (V_f) y en el stub (V_{fs}).

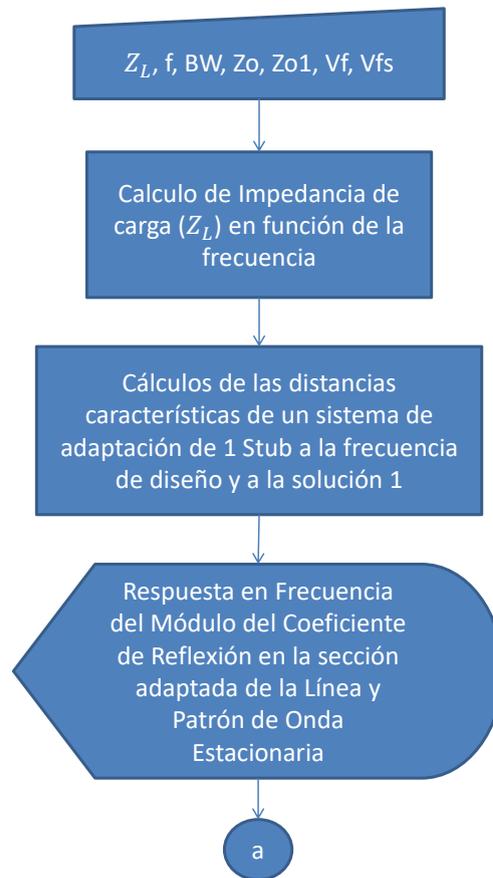


Figura 3.11: Adaptación mediante 1 stub: Algoritmo de inicio y resultados principales.

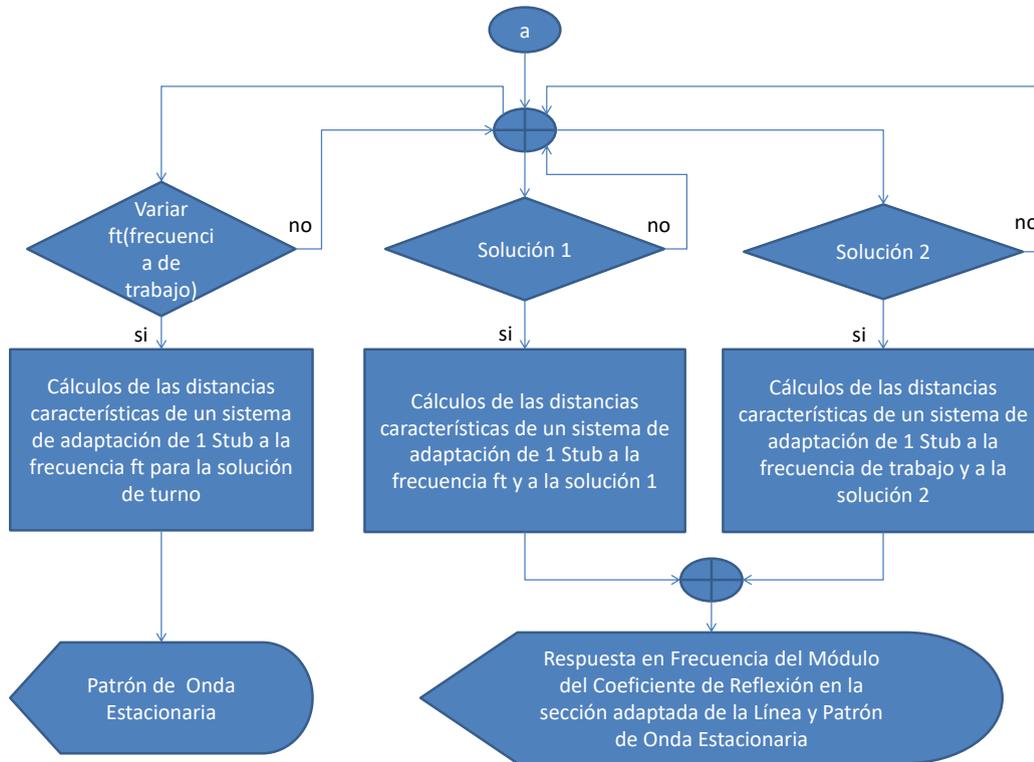


Figura 3.12: Adaptación mediante 1 stub: Algoritmo de variaciones sobre el sistema.

3.5.2.3. Módulo de adaptación de impedancia mediante dos líneas terminadas en cortocircuito dispuestas en paralelo

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.

- Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
- Se creó el botón para mostrar por pantalla los resultados.
- Se creó una pareja de botones de opción para escoger que longitudes manejara el sistema si las de la solución 1 o la solución 2.
- Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
- Se generaron los algoritmos que realizan los cálculos para construir las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión.
- Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

En las figuras 3.13 y 3.14 se muestra el algoritmo seguido para calcular las longitudes características del sistema, así como para dibujar las curvas patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión. Los datos de entrada son: impedancia de carga (Z_L), distancia entre la carga y el primer stub (d_s), distancia entre los stubs (d), frecuencia de diseño (f), ancho de banda (BW), impedancia característica de la línea (Z_0), impedancia característica del stub 1 (Z_{o1}), impedancia característica del stub 2 (Z_{o2}), velocidad de propagación en la línea (V_f), en el stub 1 (V_{f1}) y en el stub 2 (V_{f2}).

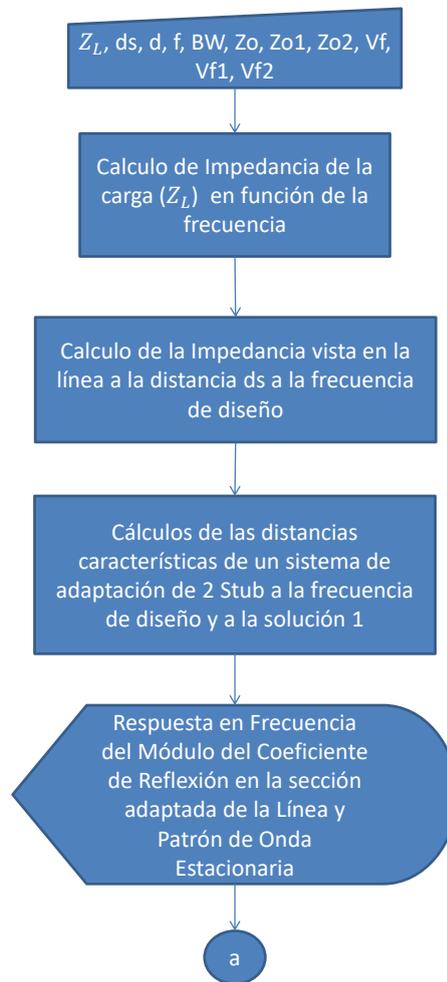


Figura 3.13: Adaptación mediante 2 stub: Algoritmo de inicio y resultados principales.

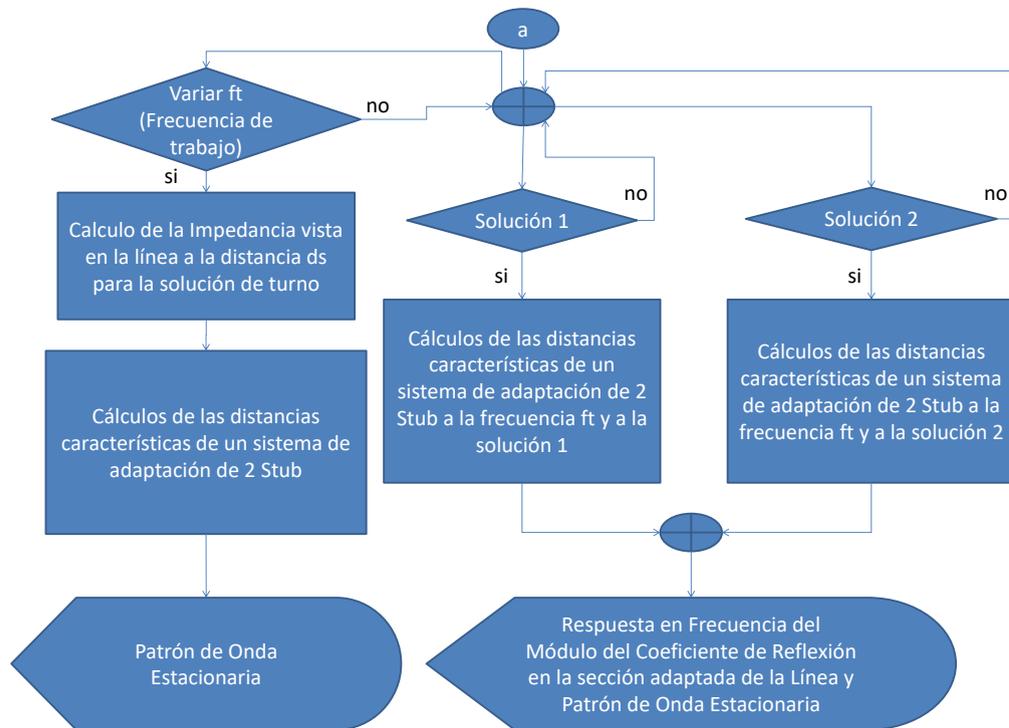


Figura 3.14: Adaptación mediante 2 stubs: Algoritmo de variaciones sobre el sistema.

3.5.2.4. Módulo de adaptación de impedancia mediante el transformador de cuarto de onda

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.

- Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
- Se creó el botón para mostrar por pantalla los resultados.
- Se creó una pareja de botones de opción para escoger que tipo de carga estará caracterizando a el adaptador, real y máxima, o real y mínima.
- Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
- Se generaron los algoritmos que realizan los cálculos para construir las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión.
- Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

En las figuras 3.15 y 3.16 se muestra el algoritmo seguido para calcular el ancho de banda ofrecido por el sistema de adaptación, la impedancia característica del adaptador a la frecuencia de diseño, las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión. Los datos de entrada son: impedancia de carga (Z_r), impedancia característica de la línea (Z_0), frecuencia de diseño (f), velocidad de propagación en la línea (V_f) y en el adaptador (V_{fa}), coeficiente de reflexión máximo en la sección adaptada de la línea (Γ_m).

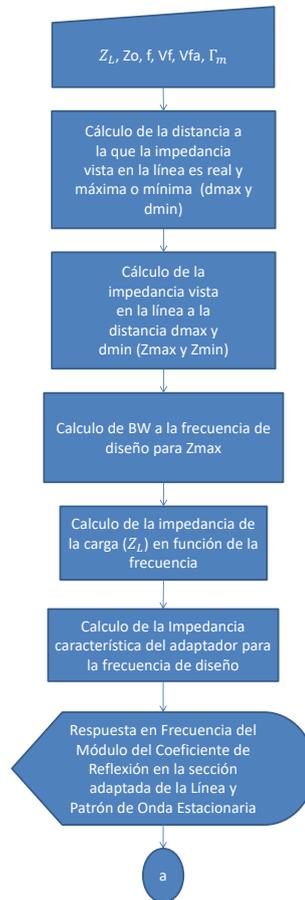


Figura 3.15: Adaptación mediante transformador de cuarto de onda: Algoritmo de inicio y resultados principales.

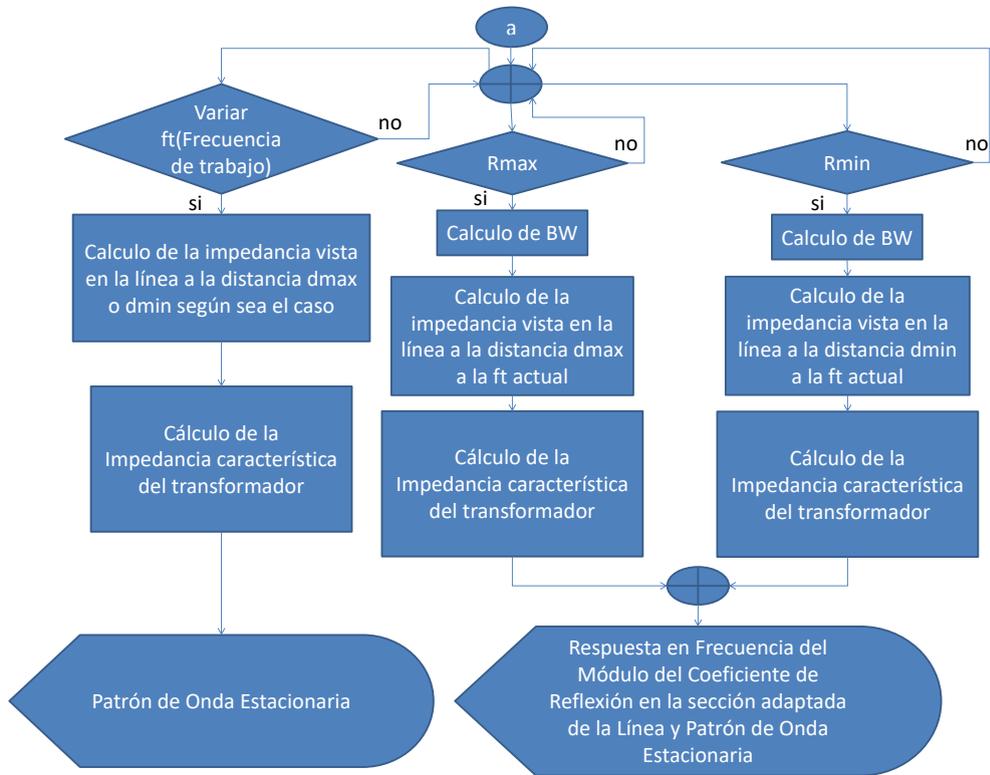


Figura 3.16: Adaptación mediante transformador de cuarto de onda: Algoritmo de variaciones sobre el sistema.

3.5.2.5. Módulo Carta de Smith

Este módulo fue diseñado para resolver problemas en régimen permanente usando la Carta de Smith, para ello es necesario recrear una de esta de forma virtual, de esta problemática surge la necesidad de usar una librería gráfica llamada pySmithPlot la cual crea una carta de Smith en forma de vector, tal que acepta como parámetros el valor de impedancia o admitancia normalizada, logrando tener una carta redimensionable y ofrece mayor facilidad para generar gráficos sobre ella.

Para la construcción de este módulo se realizó el siguiente procediendo:

- colocar un entorno gráfico donde se observara la carta de Smith.
- Identificar las ecuaciones que deben ser usadas en cada opción para obtener los parámetros de salida.
- Identificar los parámetros de entrada para cada opción que ofrece el módulo.
- Identificar los parámetros de salida para cada opción que ofrece el módulo.
- En lenguaje Python:
 - Se generó los menús que contiene los campos de entrada para introducir o modificar los parámetros.
 - se generó un entorno donde se ubica la Carta de Smith digitalizada.
 - Se generó los botones de navegación que permiten ir de un menú a otro dentro del módulo.
 - Se generó los botones que permiten al usuario animar el procedimiento usado o simplemente observar el resultado final.
 - Se generó el algoritmo que permite la animación de los diferentes opciones que ofrece este módulo.
 - Se generó un algoritmo que indica paso a paso el procedimiento que se usó para obtener el resultado final de las diferentes opciones que ofrece el módulo.

- Se generó las etiquetas para mostrar resultados en el menú correspondiente.

El algoritmo que indica paso a paso el procedimiento que usa este módulo para dar solución a las diferentes opciones, se muestra en el siguiente diagrama de flujo:

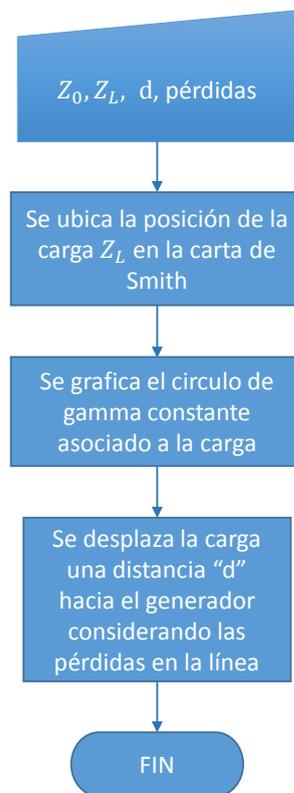


Figura 3.17: Diagrama de flujo para el asistente o ayuda de la Carta de Smith

Este módulo atiende los siguientes requerimientos:

Cálculo de $Z(d)$

En la Figura 3.18 se muestra el algoritmo para calcular una impedancia en la línea a una distancia (d) y ubicarla en la Carta de Smith.

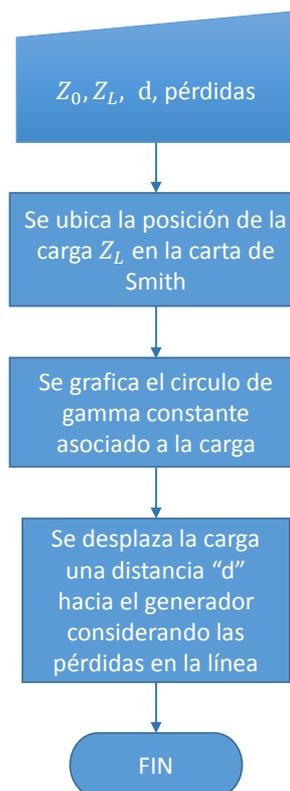


Figura 3.18: Diagrama de flujo para calcular $Z(d)$

Adaptar impedancia con un Stub

En la figura 3.19 se muestra el algoritmo para adaptar con una línea terminada en corto circuito dispuesta en paralelo con la línea, utilizando la Carta de Smith, por lo que el procedimiento es gráfico, además en una lista se indica el procedimiento.

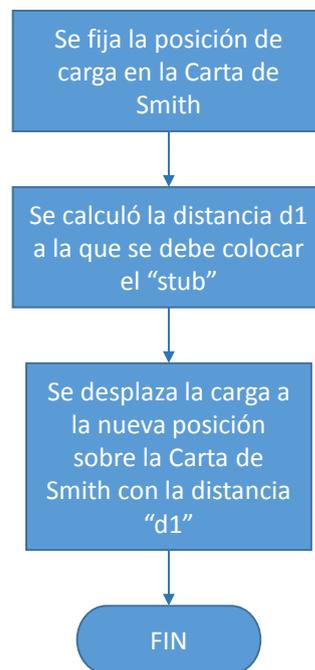


Figura 3.19: Diagrama de flujo para adaptar impedancia mediante un Stub

Adaptar impedancia con dos Stubs

En la figura 3.20 se muestra el algoritmo para adaptar con 2 líneas terminadas en corto circuito, mediante la Carta de Smith, lo que hace que el proceso sea gráfico, además en una lista se indica el procedimiento.

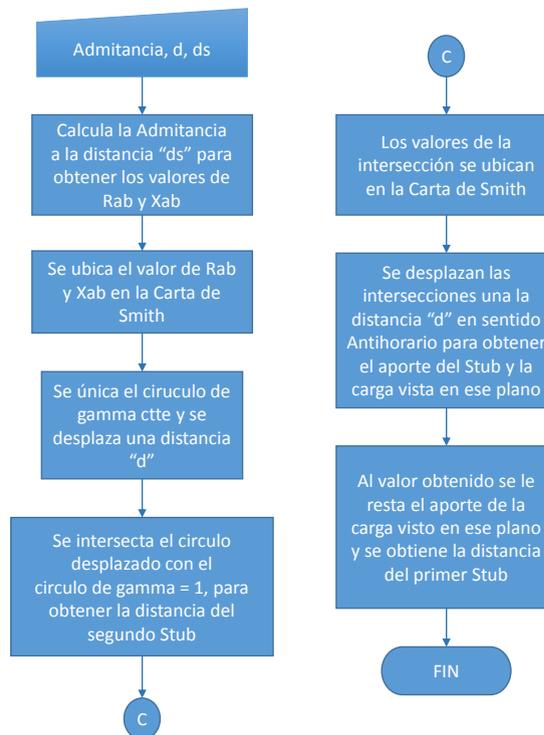


Figura 3.20: Diagrama de flujo para adaptar impedancia mediante dos Stubs

Adaptar impedancia con elementos concentrados

En la figura 3.21 se muestra el algoritmo para adaptar mediante elementos concentrados usando la Carta de Smith, por ende el proceso es gráfico, además en una lista se indica el procedimiento.

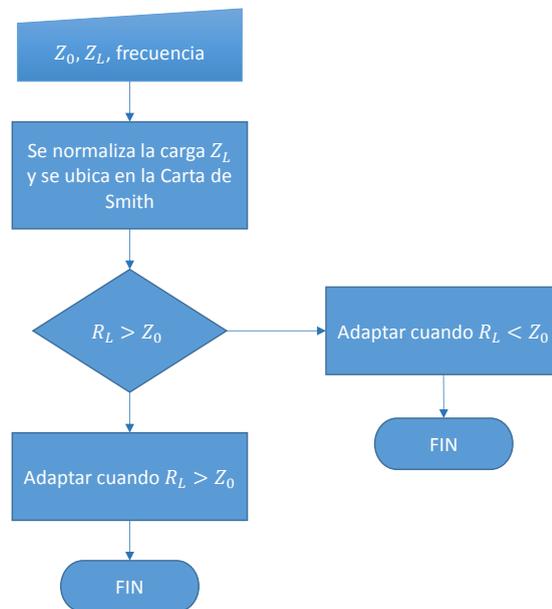


Figura 3.21: Diagrama de flujo para adaptar impedancia mediante elementos concentrados

En la figura se detallan los bloques que realizan el procedimiento de adaptar cuando $R_L > Z_o$ o $R_L < Z_o$.

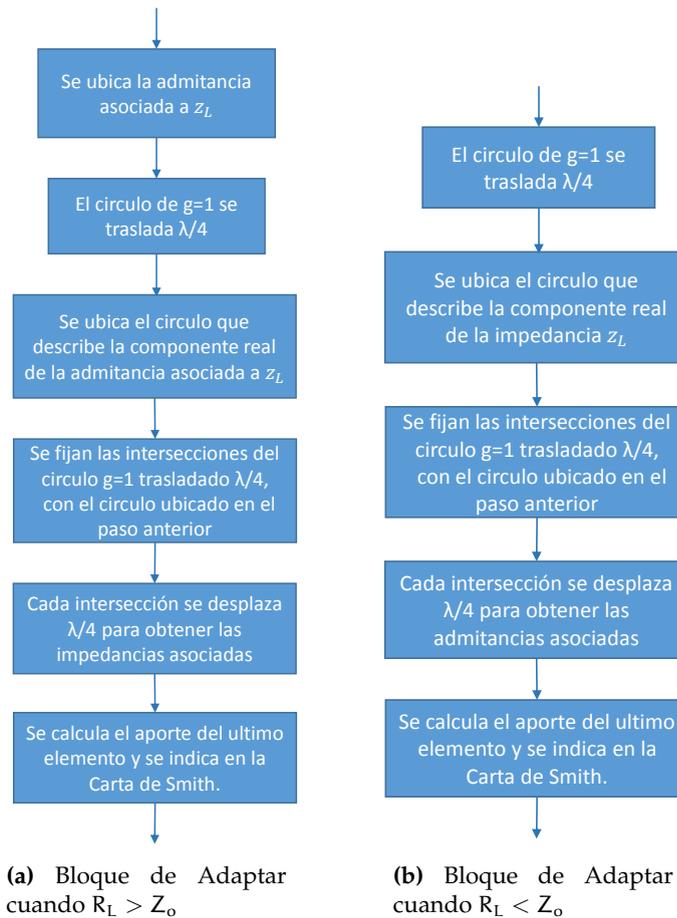


Figura 3.22: Diagramas de flujo para los bloques de adaptar

3.5.2.6. Módulo de adaptación de impedancia mediante el transformador binomial

Para desarrollar el módulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:

- Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
- Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
- Se creó el botón para mostrar por pantalla los resultados.
- Se creó una pareja de botones de opción para escoger que tipo de carga estará caracterizando a el adaptador, real y máxima, o real y mínima.
- Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
- Se generaron los algoritmos que realizan los cálculos para construir las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión.
- Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

En las figuras 3.23 y 3.24 se muestra el algoritmo seguido para calcular el ancho de banda ofrecido por el sistema de adaptación, las impedancias características de cada sección del adaptador a la frecuencia de diseño, las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión. Los datos de entrada son: impedancia de carga (Z_L), impedancia característica de la línea (Z_0), frecuencia de diseño (f), velocidad de propagación en la línea (V_f) y en el adaptador (V_{fa}), número de secciones (N), coeficiente de reflexión máximo en la sección adaptada de la línea (Γ_m).

3.5.2.7. Módulo de adaptación de impedancia mediante el transformador Chebyshev

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.

- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
 - Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
 - Se creó el botón para mostrar por pantalla los resultados.
 - Se creó una pareja de botones de opción para escoger que tipo de carga estará caracterizando a el adaptador, real y máxima, o real y mínima.
 - Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
 - Se generaron los algoritmos que realizan los cálculos para construir las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión.
 - Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

En las figuras 3.23 y 3.24 se muestra el algoritmo seguido para calcular el ancho de banda ofrecido por el sistema de adaptación, las impedancias características de cada sección del adaptador a la frecuencia de diseño, las curvas del patrón de onda estacionaria y respuesta en frecuencia del módulo del coeficiente de reflexión. Los datos de entrada son: impedancia de carga (Z_L), impedancia característica de la línea (Z_0), frecuencia de diseño (f), velocidad de propagación en la línea (V_f) y en el adaptador (V_{fa}), número de secciones (N), coeficiente de reflexión máximo en la sección adaptada de la línea (Γ_m).

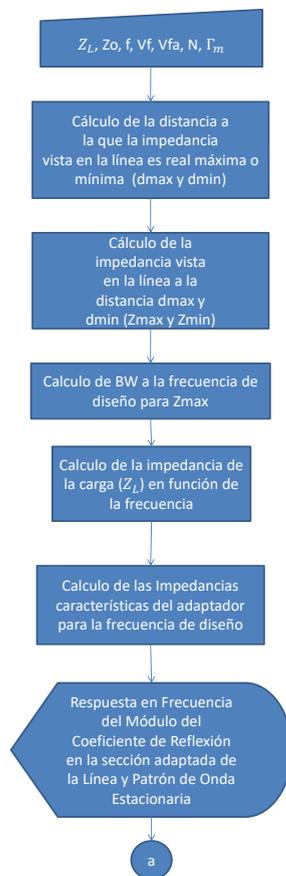


Figura 3.23: Adaptación mediante transformador Binomial y Chebyshev: Algoritmo de inicio y resultados principales.

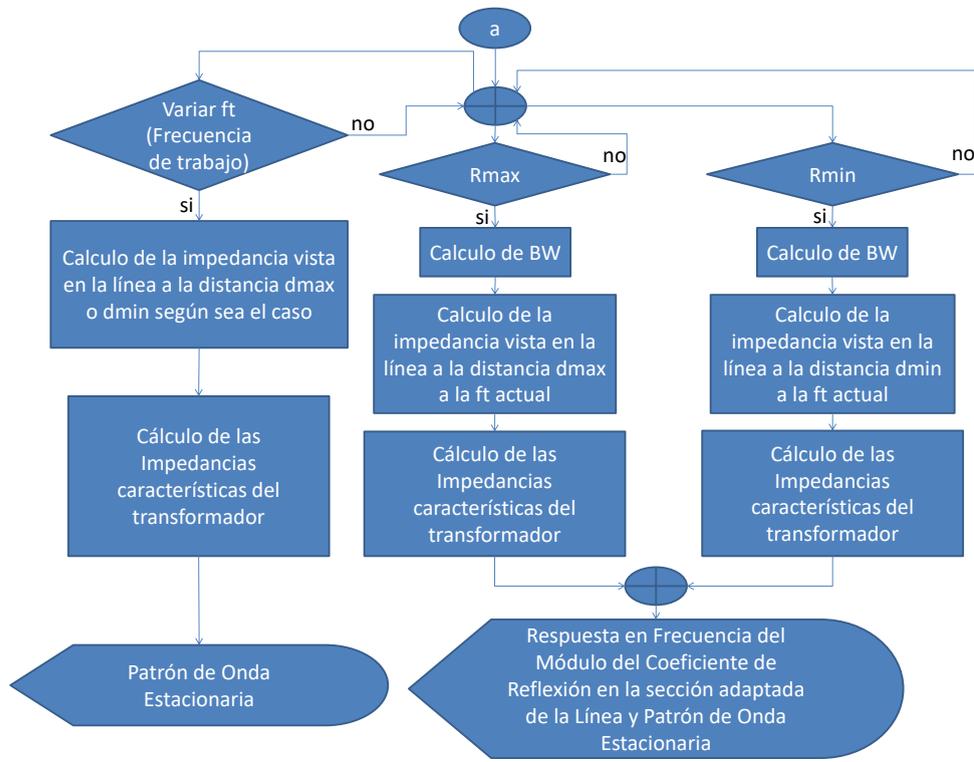


Figura 3.24: Adaptación mediante transformador Binomial y Chebyshev: Algoritmo de variaciones sobre el sistema.

Notese que las figuras 3.23 y 3.24 son el algoritmo para transformador binomial y chebyshev, sin embargo el algoritmo en Python de cada modulo se vio diferenciado debido a que las funciones con las cuales se determinan los coeficientes de reflexión en cada caso son diferentes siendo respectivamente la función binomial y la chebyshev.

3.5.2.8. Módulo de adaptación de impedancia mediante el transformador de línea no uniforme (Tapered Line)

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
 - Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
 - Se creó el botón para mostrar por pantalla los resultados.
 - Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
 - Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

El algoritmo de cálculo considera los datos de entrada y si la carga esta en la línea de menor o de mayor impedancia característica, hecho esto calcula la resistencia real mínima o la resistencia real máxima respectivamente para cargar a la línea no uniforme, posteriormente se genera el valor de la constante “a” que caracteriza a la línea no uniforme, se construyen las curvas de impedancia y de respuesta en frecuencia del módulo del coeficiente de reflexión, se exhiben los resultados en

pantalla. Los datos de entrada son: impedancia de carga (Z_L), frecuencia de diseño (f), longitud de la línea (L), impedancias características de las líneas conectada a la línea no uniforme (Z_{o1} y Z_{o2}), velocidad de propagación en las líneas conectada a la línea no uniforme (V_{f1} y V_{f2}).

3.5.2.9. Módulo de balun de constantes distribuidas

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
 - Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
 - Se creó el botón para mostrar por pantalla los resultados.
 - Se creó un campo para introducir variaciones en la frecuencia de operación del sistema.
 - Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

El algoritmo de cálculo considera si la carga esta en la sección balanceada o desbalanceada, a partir de esto configura los parámetros del sistema y procede a calcular las longitudes de cada línea de transmisión, posterior a esto construye las curvas del patron de ondas estacionarias y respuesta en frecuencia del módulo del coeficiente de reflexión que luego mostrara en la misma interfaz de exhibición de resultados. Los datos de entrada son: impedancia de carga (Z_L), frecuencia de diseño (f), impedancias características de las líneas que conforman el balun (Z_{o1} y Z_{o2}), velocidad de propagación en las líneas que conforman el balun (V_{f1} y V_{f2}).

3.5.2.10. Módulo de balun con elementos concentrados

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
 - Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
 - Se creo el botón para mostrar por pantalla los resultados.
 - Se creo un botón para volver a la interfaz que contiene los campos de introducción de datos.

El algoritmo de cálculo consiste en tomar los valores en los campos para usarlos en las siguientes ecuaciones.

$$X = \frac{Z_b}{\sqrt{Z_b Z_d}} \quad (3.36)$$

$$C = \frac{1}{2\pi f X} \quad (3.37)$$

$$L = \frac{X}{2\pi f} \quad (3.38)$$

De las ecuaciones anteriores se deduce que los datos de entrada son: impedancia de la línea desbalanceada (Z_d), impedancia de la línea balanceada (Z_b), frecuencia de trabajo (f_0).

Con C y L el usuario posee el conocimiento para diseñar un balun concentrado útil para el sistema que esta manejando.

3.5.2.11. Módulo de parámetros de dispersión

Para desarrollar el modulo se siguieron los siguientes pasos:

- Se determinó que ecuaciones usar.
- Se establecieron los parámetros de entrada y salida.
- En python:
 - Se crearon los campos mediante los cuales se inserta al programa los datos del sistema.
 - Se generaron los algoritmos que realizan los cálculos para los parámetros de salida.
 - Se creó el botón para mostrar por pantalla los resultados.
 - Se crearon 4 botones de opción para escoger que parámetros se quiere conocer del sistema.
 - Se creó un botón para volver a la interfaz de introducción de datos en caso de querer variar los datos del sistema.

El algoritmo de calculo se fundamento en tomar los datos de los campos de entrada para introducirlos en la ecuaciones que permiten conocer los parámetros de dispersión (parámetros S) para sistemas de dos puertos y a partir de estos parámetros obtener, los parámetros, Z , Y , o $ABCD$, según el usuario lo escoja a través de los 4 botones de opción dispuestos en el menú de entrega de datos. Los datos de entrada son: impedancia característica de la línea (Z_d), longitud de la línea (L), velocidad de propagación en la línea (V_p), frecuencia de trabajo (f_0).

3.5.3. Módulo de interacción virtual con equipos y accesorios presentes en el laboratorio de Sistemas de Ondas Guiadas

Antes de crear este módulo fue necesario revisar las prácticas que se desarrollan en el laboratorio de la asignatura, además de comprender el funcionamiento de

cada uno de los instrumentos y accesorios que se usan para llevar a cabo las prácticas, con el fin de virtualizar cada uno de estos elementos y hacer un entorno virtual donde el usuario interactue de forma similar a como lo hace en el laboratorio real.

Para generar este entorno virtual se realizó lo siguiente:

- Identificar el comportamiento de cada instrumento del laboratorio.
- Identificar los algoritmos para representar el funcionamiento de los instrumentos existentes en el laboratorio.
- Generar la capacidad de arrastrar y soltar cada elemento del entorno virtual.
- Identificar el montaje realizado por el usuario.
- Identificar la salida generada por el montaje creado.
- En lenguaje Python:
 - Se generó el algoritmo que representa el funcionamiento de cada instrumento.
 - Se generó la interfaz gráfica de cada instrumento.
 - Se generó una interfaz gráfica dividida en dos zonas: donde se encuentran los instrumentos y accesorios, también se tiene la zona de trabajo, donde existe la conexión entre los elementos.
 - Se generó los botones que permiten la edición del montaje implementado.

3.6. Construcción de SEDISOG V3.0 usando Python

Generado todos los módulos nombrados en secciones anteriores, se generó una interfaz gráfica principal donde se integraran cada uno de ellos, permitiendo al usuario navegar entre cada una de las opciones que presenta el programa sin problemas.

Para ello, en lenguaje Python se realizó lo siguiente:

- Se generó una interfaz gráfica que contuviera todos los módulos anteriormente descritos
- Se generó una barra de herramientas.
- Se generó en la barra de herramientas, botones desplegable tales como:
 - **Contenido:** Se encuentran los temas que son estudiados en la asignatura SOG.
 - **Herramientas:** Se encuentran los módulos que describen el régimen transitorio y permanente en la línea de transmisión.
 - **Laboratorio Virtual:** Se obtiene acceso al entorno virtual, donde se podrá interactuar con las herramientas existentes en el laboratorio de la asignatura, de manera virtual.

Capítulo IV

Análisis, interpretación y presentación de los resultados

El objetivo de este capítulo es presentar una descripción de la estructura del programa y las diferentes funciones que posee, además de explicar los resultados obtenidos con respecto a los objetivos planteados.

4.1. Logo del programa.

En la figura 4.1 se muestra el logo de la nueva versión del programa (SEDISOG 3).



Figura 4.1: Logo del programa

4.2. Interfaz gráfica de usuario

A continuación se muestran los resultados del diseño, de las ventanas que conforman el programa:

4.2.1. Ventana principal

El programa presenta una ventana inicial de tamaño variable, en esta se encuentran las opciones principales de navegación que permiten al usuario ir a las siguientes ventanas. Para ello véase la figura 4.2



Figura 4.2: Ventana Principal de SEDISOG 3

Esta ventana contiene:

- Barra de Título: Indica el nombre de la aplicación
- Barra de Herramientas: Contiene una lista de menus que se despliegan y presentan opciones, para ser activados a través del mouse.

- Botón Minimizar: Sirve para ocultar momentáneamente la aplicación que se muestra en pantalla sin cerrarla.
- Botón Maximizar: Sirve para ampliar la ventana y así ocupar toda la pantalla del monitor, con el fin de observar mejor.
- Botón Cerrar: sirve para salir de la aplicación.
- Logo del programa

4.2.2. Ventanas auxiliares

las ventanas auxiliares son aquellas que dan soporte a las siguientes opciones ubicadas en la barra de herramienta:

- Archivo: Al seleccionar esta opción se despliega un menú en el que se encuentra la opción de Home, Como se muestra en la figura [4.3](#).
- Contenido: Al seleccionar esta opción se despliega un menú en el que se encuentra el índice general, como se muestra en la figura [4.4](#).
- Índice General: en esta ventana se encuentre el contenido de la asignatura, como se muestra en la figura [4.5](#).

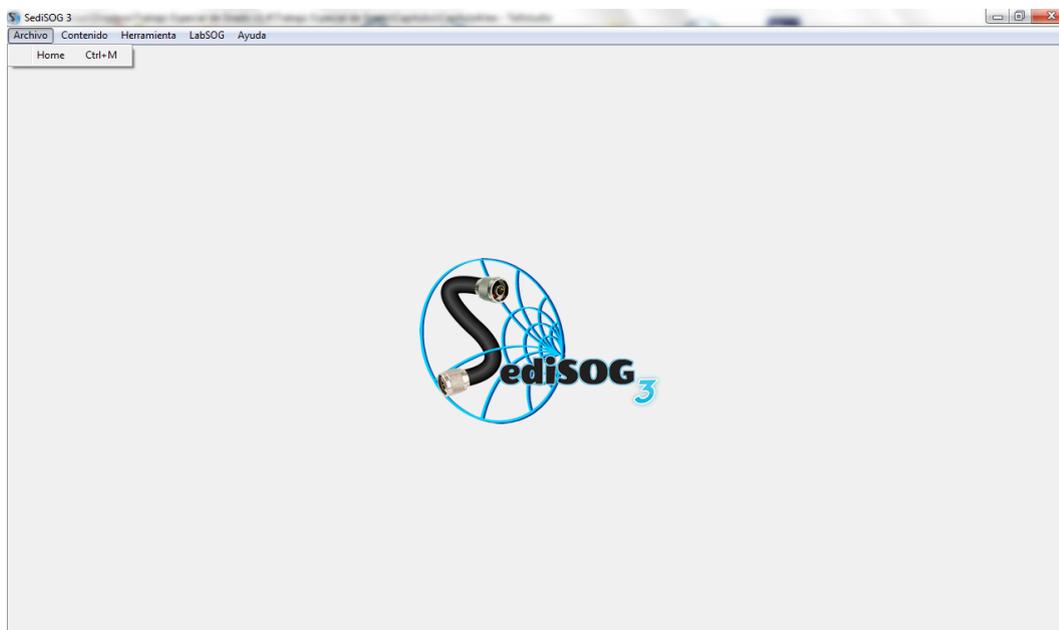


Figura 4.3: Despliegue de la opción Archivo

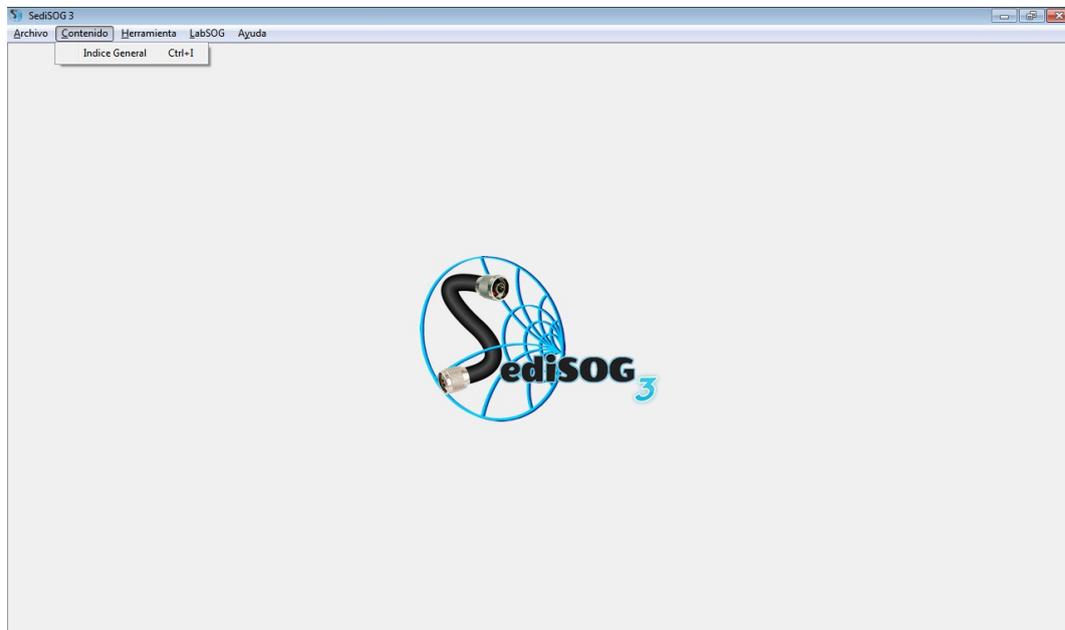


Figura 4.4: Despliegue de la opción contenido

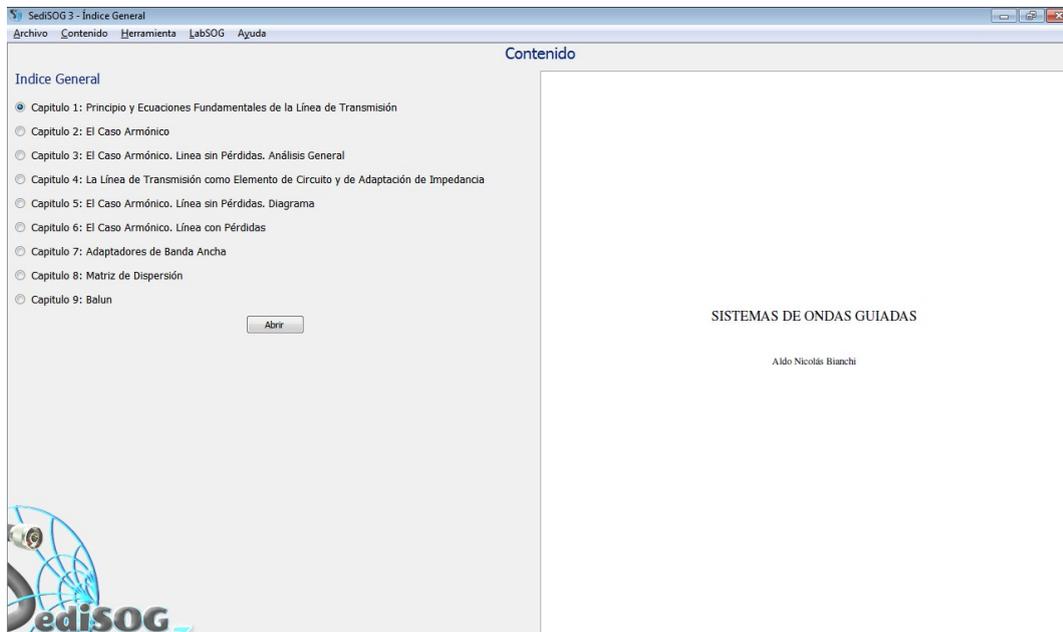


Figura 4.5: Contenido de la asignatura

En la barra de herramientas se encuentra la opción "Herramientas" que contiene un menú desplegable con los diferentes módulos relacionados con la asignatura, como se muestra en la figura 4.6.

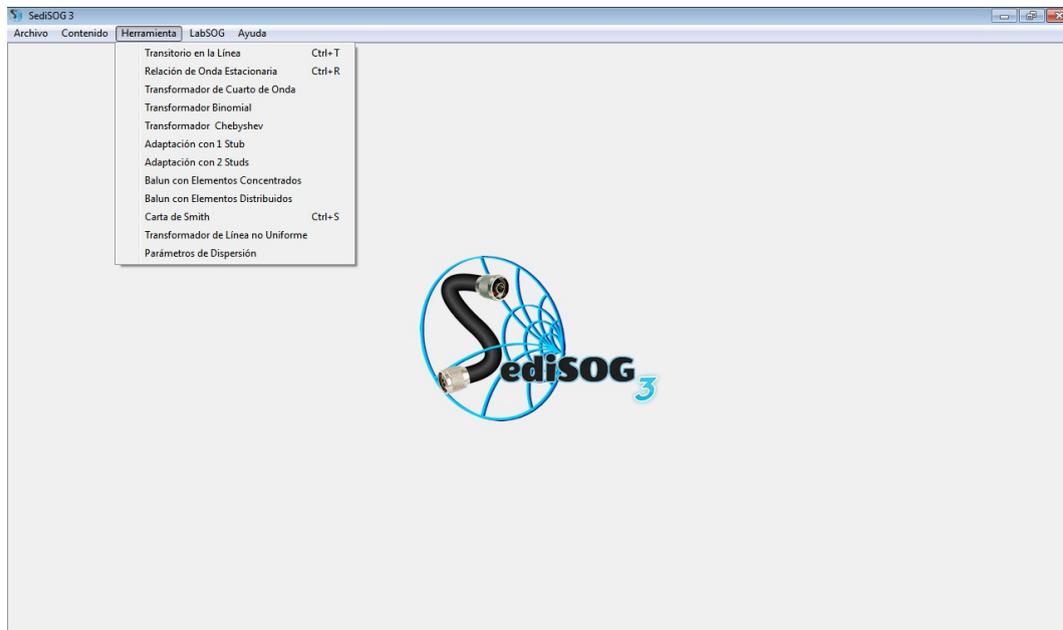


Figura 4.6: Despliegue de la opción herramientas

4.2.3. Ventana para el módulo transitorio

Esta ventana permite representar los transitorios en un arreglo de líneas de transmisión con diferentes tipos de cargas y discontinuidades. En la figura 4.7 se muestra la interfaz que contiene los campos de entrada donde el usuario podrá introducir o modificar parámetros del arreglo seleccionado.

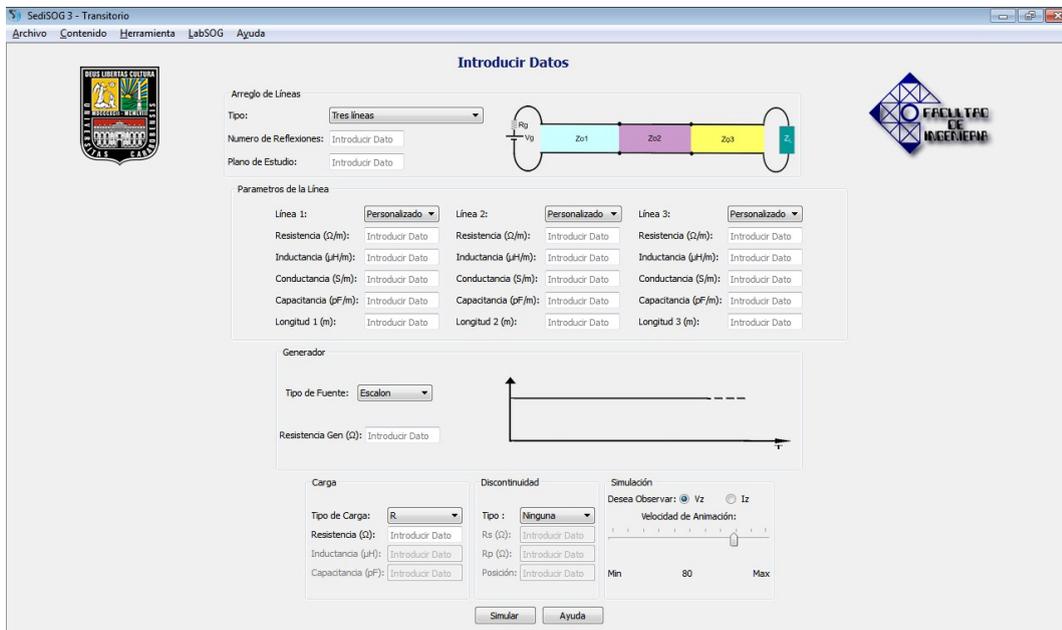


Figura 4.7: Interfaz para el módulo Transitorio

La cantidad de parámetros depende del tipo de arreglo que se desea simular, en la figura 4.8 se muestra la interfaz donde se ilustraran los resultados de la simulación para arreglos lineales de líneas de transmisión.

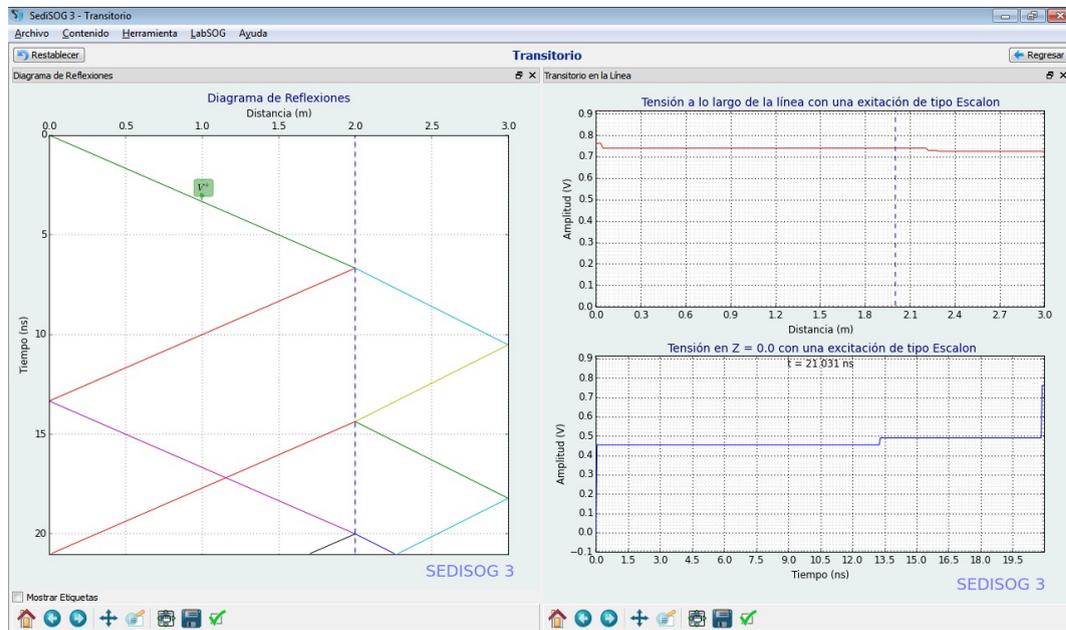


Figura 4.8: Interfaz para visualizar transitorios en arreglos lineales

En la figura anterior se muestra el resultado de simular un arreglo similar al de la figura 4.9, además se puede observar que la interfaz cuenta con dos ventanas, una para ilustrar el diagramas de reflexiones asociado y otra muestra un plot para observar la evolución temporal y espacial del voltaje (V) o corriente (I) en todo el arreglo y cuenta con otro plot en el que se grafica la reflectometría en el dominio temporal (TDR) de tensión o corriente respectivamente, en el plano de observación elegido.

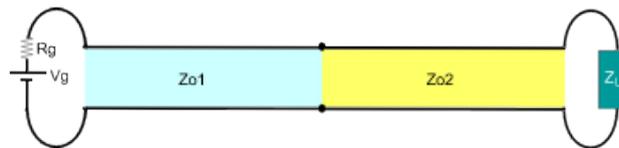


Figura 4.9: Arreglo de dos líneas

Para mostrar la simulación de un arreglo conformado por tres líneas de transmisión dispuestas en paralelo se generó un entorno gráfico como se ilustra en la figura 4.10, el cual cuenta con dos ventanas, una para el diagramas de reflexiones, y la segunda donde se muestra el comportamiento temporal y espacial del voltaje

(V) y a corriente (I) a lo largo de cada línea de transmisión.

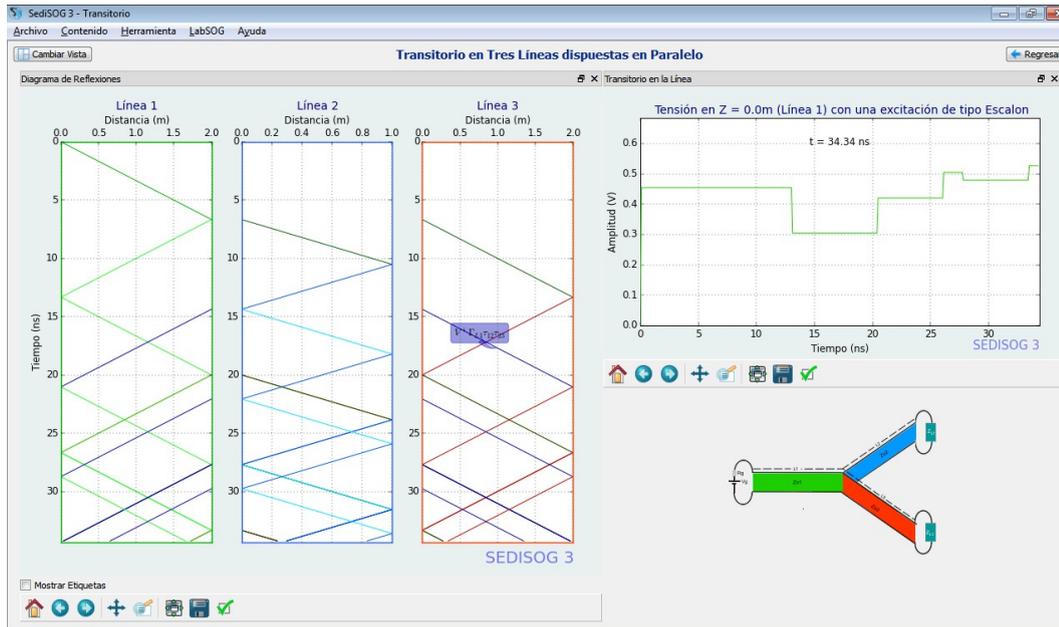


Figura 4.10: Interfaz para mostrar resultados al simular un arreglo de tres líneas dispuestas en paralelo

La interfaz cuenta con dos botones:

- Regresa: Este botón permite regresar a la interfaz del módulo transitorio donde el usuario podrá introducir y modificar parámetros.
- Cambiar Vista: Este botón genera un menú en esta interfaz que permite al usuario decidir los resultados que desea visualizar, a continuación se indican las opciones disponibles (Además Véase 4.11):
 - Diagrama de Reflexiones, en conjunto con una reflectometría en el dominio temporal para un plano de estudio en una línea específica.
 - Diagrama de Reflexiones, acompañado de un plot que muestra la evolución del voltaje o corriente en la línea 1
 - Diagrama de Reflexiones, acompañado de un plot que muestra la evolución del voltaje o corriente en la línea 2

- Diagrama de Reflexiones, acompañado de un plot que muestra la evolución del voltaje o corriente en la línea 3
- La evolución de tensión o corriente en las tres líneas, más una reflectometría en el dominio temporal para un plano de estudio en una línea específica.
- Diagrama de reflexiones, en conjunto con La evolución de tensión o corriente en las tres líneas, más una reflectometría en el dominio temporal para un plano de estudio en una línea específica.

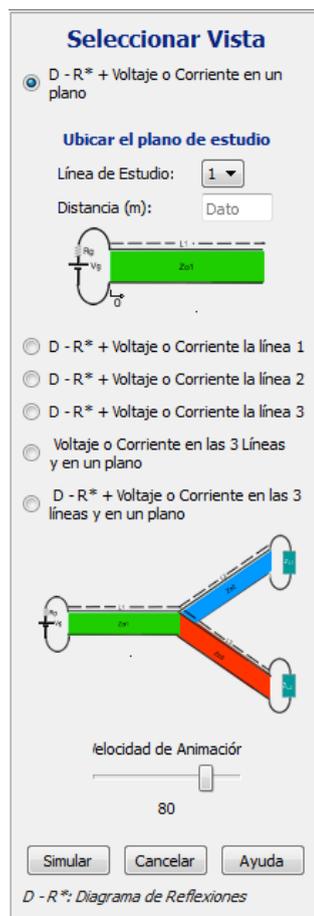
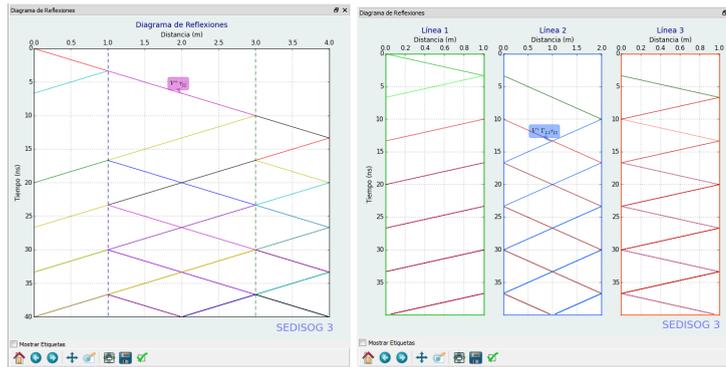


Figura 4.11: Menú para elegir vista

El diagrama de reflexiones tiene el siguiente aspecto:



(a) Diagrama de reflexiones para un arreglo de lineal

(b) Diagrama de reflexiones para un arreglo de tres líneas de transmisión

Figura 4.12: Diagrama de reflexiones

En la figura 4.12 se observan los diagramas de reflexiones para los 2 tipos de arreglos permitidos en el programa al momento de simular transitorios, en los cuales se puede ver la presencia de etiquetas que identifican el aporte de cada reflexión.

4.2.4. Ventana para el módulo relación de onda estacionaria (ROE)

Este módulo permite graficar el régimen de ondas estacionarias en la línea de transmisión para cualquier valor de carga e impedancia característica. En la figura 4.13 se muestra la apariencia de este entorno el cual cuenta con una interfaz para introducir parámetros y otra para visualizar el comportamiento de los parámetros de salida, mediante gráficas.

En la figura 4.14 se muestra la interfaz que presenta resultados de los parámetros de salida: $|V|$, $|I|$, $|Z|$, $|P|$ y $|Q|$ para los distintos puntos de la línea. Además dicha interfaz cuenta con etiquetas que muestran el valor de cada parámetro a una distancia establecida por el usuario.

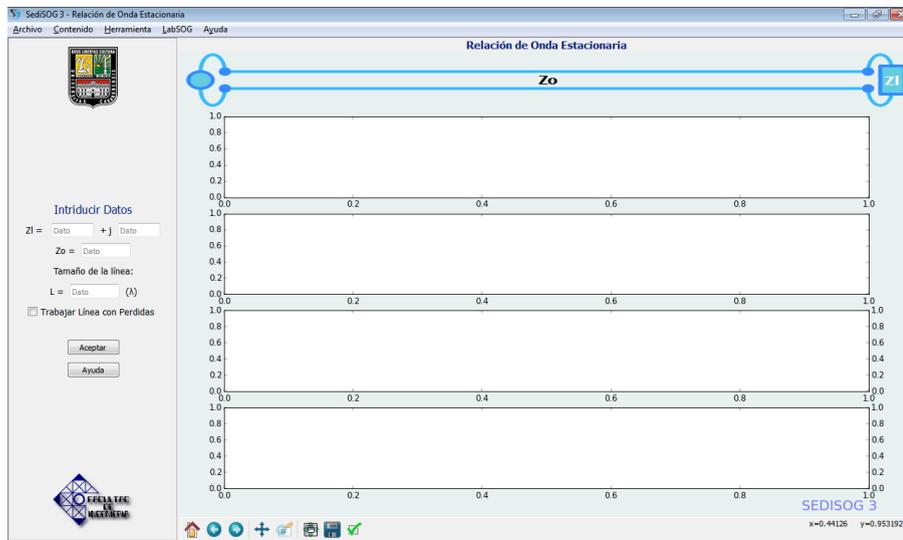


Figura 4.13: Interfaz para el módulo ROE

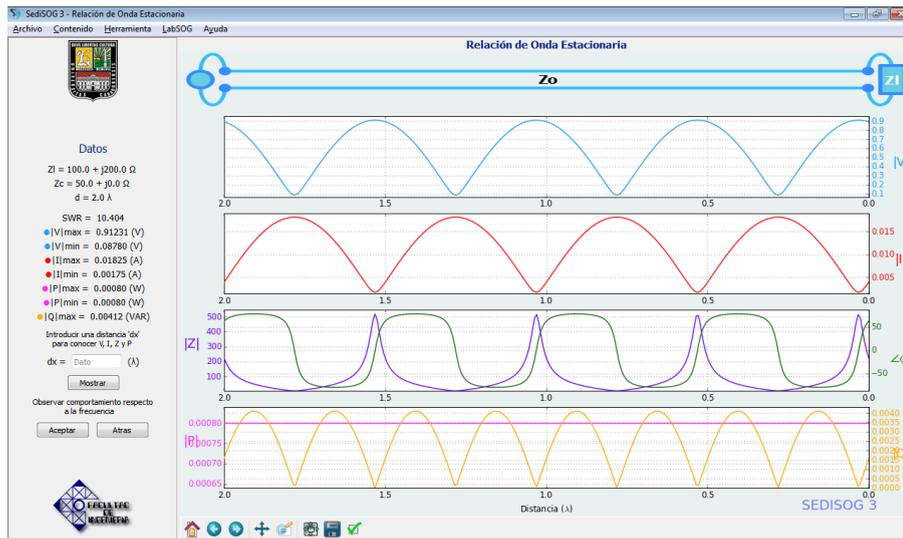


Figura 4.14: Interfaz para resultado del módulo ROE

4.2.5. Ventana para el módulo transformador de cuarto de onda

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un transformador de cuarto de onda, sirviendo al usuario de herramienta

para afrontar la complejidad visual y matemática del sistema. Posee una interfaz inicial, donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño y una interfaz secundaria donde se exhiben los parámetros de salida acompañados de las gráficas del patrón de onda estacionario de voltaje del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del modulo del coeficiente de reflexión en la sección adaptada de la línea en el ancho de banda calculado, además en esta interfaz de resultados se ofrece una pareja de botones de opción para alternar entre las dos opciones de sistemas que se pueden manejar considerando, si se quiere trabajar con una carga real mínima o máxima y un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura 4.15 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.16 se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con los botones de opción para variar la solución y el campo para variar la frecuencia.

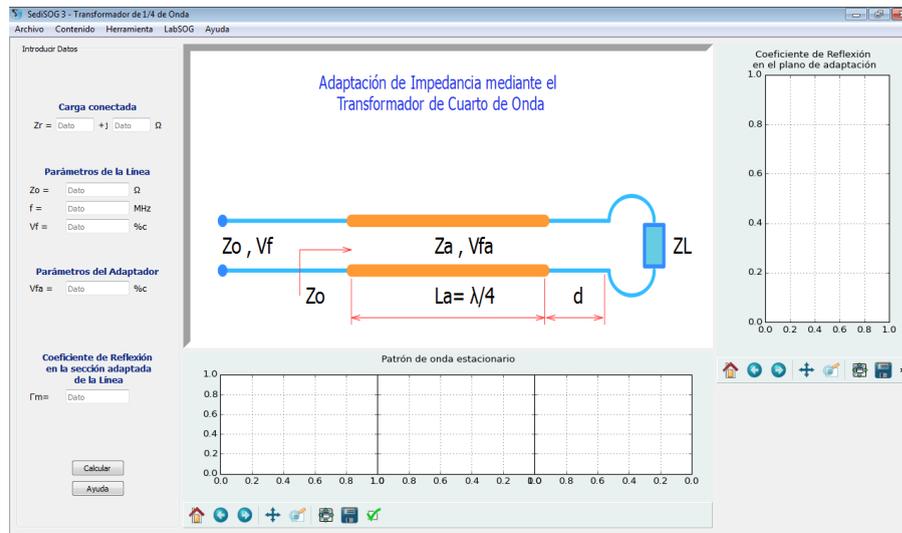


Figura 4.15: Interfaz de introducción de datos del modulo adaptador de impedancias mediante el transformador de cuarto de onda

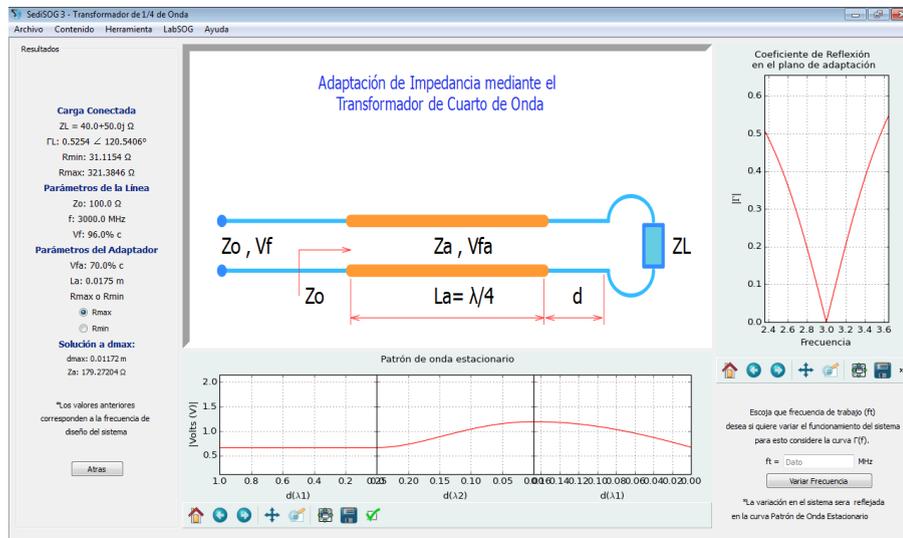


Figura 4.16: Interfaz de resultados del modulo adaptador de impedancias mediante el transformador de cuarto de onda

4.2.6. Ventana para el módulo transformador Binomial

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un transformador Binomial, sirviendo al usuario de herramienta para afrontar la complejidad visual y matemática del sistema. Posee una interfaz inicial donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño y una interfaz secundaria donde se exhiben los parámetros de salida acompañados de dos gráficas donde se muestran el patrón de onda estacionario de voltaje del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del módulo del coeficiente de reflexión en la sección adaptada de la línea en el ancho de banda calculado, además en esta interfaz de resultados se ofrece una pareja de botones de opción para alternar entre las dos opciones de sistemas que se pueden manejar considerando, si se quiere trabajar con una carga real mínima o máxima y un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura 4.17 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.18 se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con los botones de opción para variar la solución y el campo para variar la frecuencia.

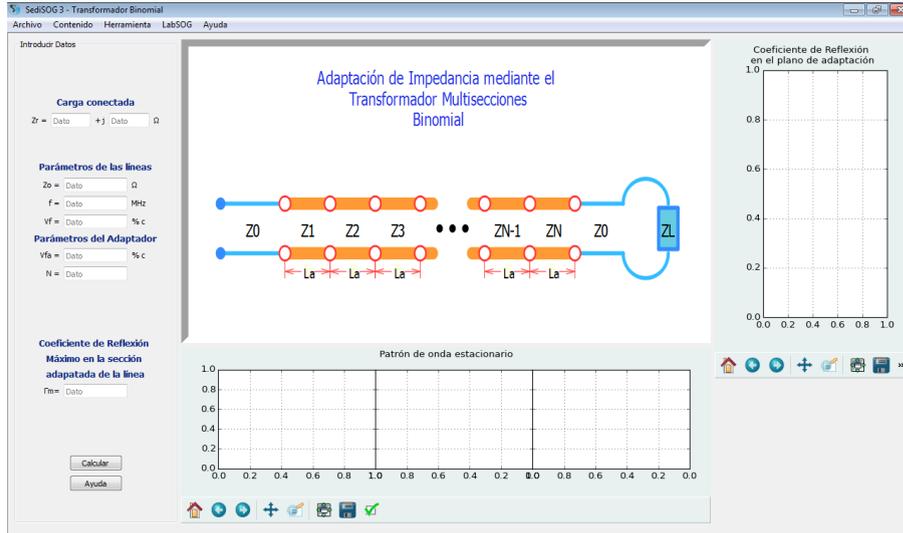


Figura 4.17: Interfaz de introducción de datos del modulo adaptador de impedancias mediante el transformador Binomial

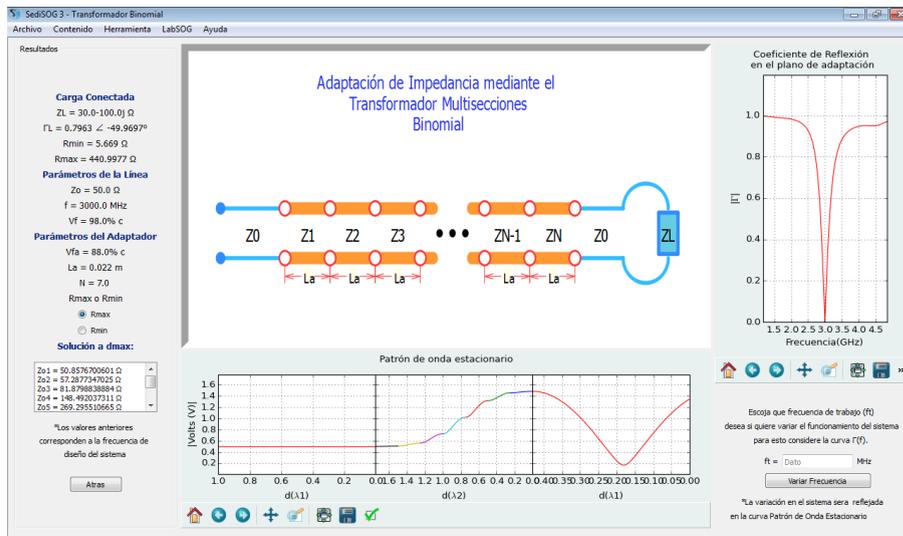


Figura 4.18: Interfaz de resultados del modulo adaptador de impedancias mediante el transformador Binomial

4.2.7. Ventana para el módulo transformador Chebyshev

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un transformador Binomial, sirviendo al usuario de herramienta para afrontar la complejidad visual y matemática del sistema. Posee una interfaz inicial donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño y una interfaz secundaria donde se exhiben los parámetros de salida acompañados de dos gráficas donde se muestran el patrón de onda estacionario de voltaje del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del modulo del coeficiente de reflexión en la sección adaptada de la línea en el ancho de banda calculado, además en esta interfaz de resultados se ofrece una pareja de botones de opción para alternar entre las dos opciones de sistemas que se pueden manejar considerando, si se quiere trabajar con una carga real mínima o máxima y un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura 4.19 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.20 se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con los botones de opción para variar la solución y el campo para variar la frecuencia.

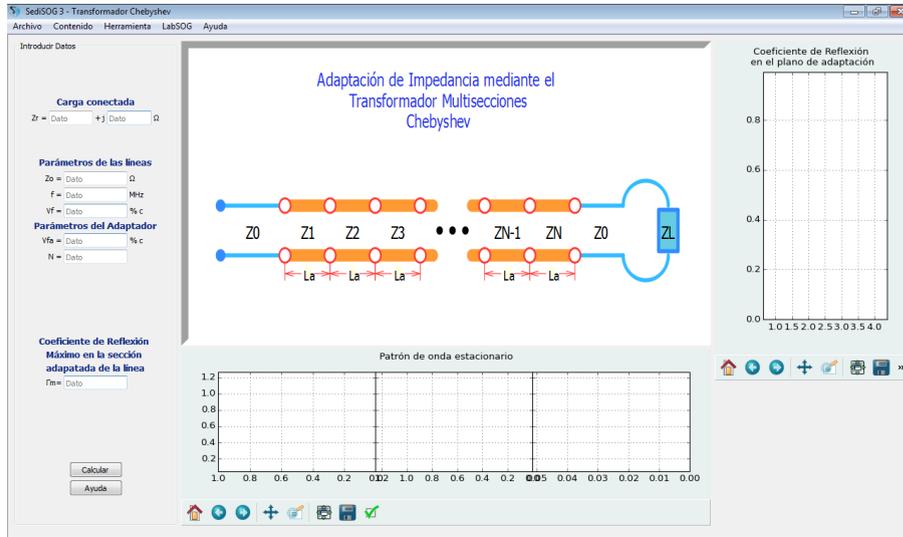


Figura 4.19: Interfaz de introducción de datos del módulo adaptador de impedancias mediante el transformador Chebyshev

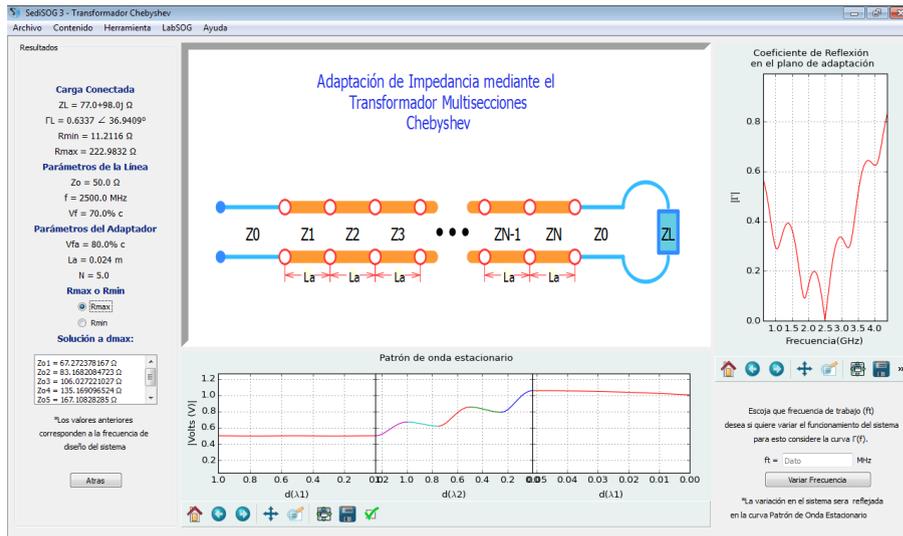


Figura 4.20: Interfaz de resultados del módulo adaptador de impedancias mediante el transformador Chebyshev

4.2.8. Ventana para el módulo adaptación con un stub

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un sistema de adaptación de un stub, sirviendo al usuario de herramienta

para afrontar la complejidad visual y matemática del sistema. Posee una interfaz de inicio donde se escoge el formato de la carga a introducir inicialmente, luego una segunda interfaz donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño y por ultimo una interfaz donde se exhiben los parámetros de salida acompañados de dos gráficas donde se muestran el patrón de onda estacionario de voltaje del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del modulo del coeficiente de reflexión en la sección adaptada de la línea en el ancho de banda especificado, además en esta interfaz de resultados se ofrece una pareja de botones de opción para alternar entre las dos opciones de sistemas que se pueden manejar considerando los parámetros de entrada y un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura 4.21 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.22 se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con los botones de opción para variar la solución y el campo para variar la frecuencia.

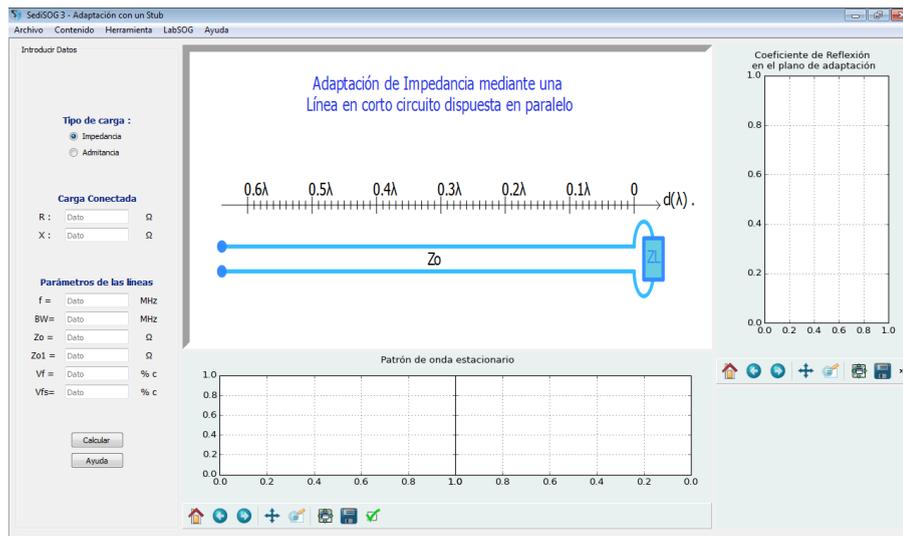


Figura 4.21: Interfaz de introducción de datos del modulo adaptador de impedancias mediante un stub

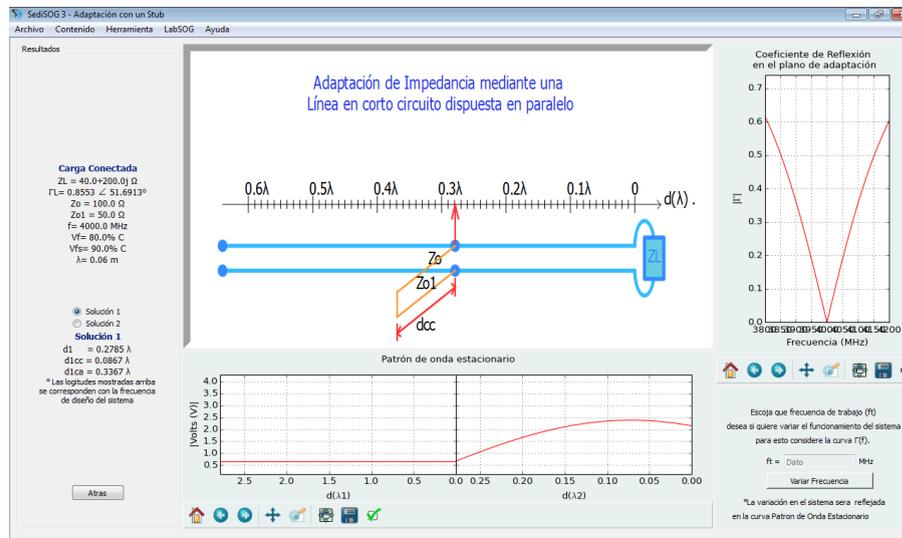


Figura 4.22: Interfaz de resultados del modulo adaptador de impedancias mediante un stub

4.2.9. Ventana para el módulo adaptación con dos stubs

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un sistema de adaptación con dos stub, sirviendo al usuario de herramienta para afrontar la complejidad visual y matemática del sistema. Posee una interfaz de inicio donde se escoge el formato de la carga a introducir inicialmente, luego una segunda interfaz donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño y por ultimo una interfaz donde se exhiben los parámetros de salida acompañados de dos gráficas donde se muestran el patrón de onda estacionario de voltaje del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del modulo del coeficiente de reflexión en la sección adaptada de la línea en el ancho de banda especificado, además en esta interfaz de resultados se ofrece una pareja de botones de opción para alternar entre las dos opciones de sistemas que se pueden manejar considerando los parámetros de entrada y un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura 4.23 se muestra la interfaz que contiene los campos de entrada don-

de el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.24 se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con los botones de opción para variar la solución y el campo para variar la frecuencia.

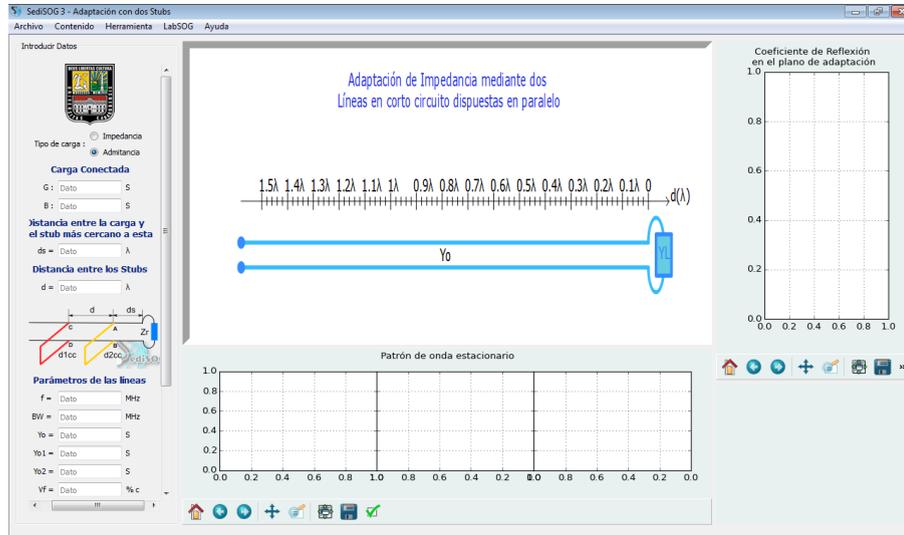


Figura 4.23: Interfaz de introduccion de datos del modulo adaptador de impedancias mediante dos stub

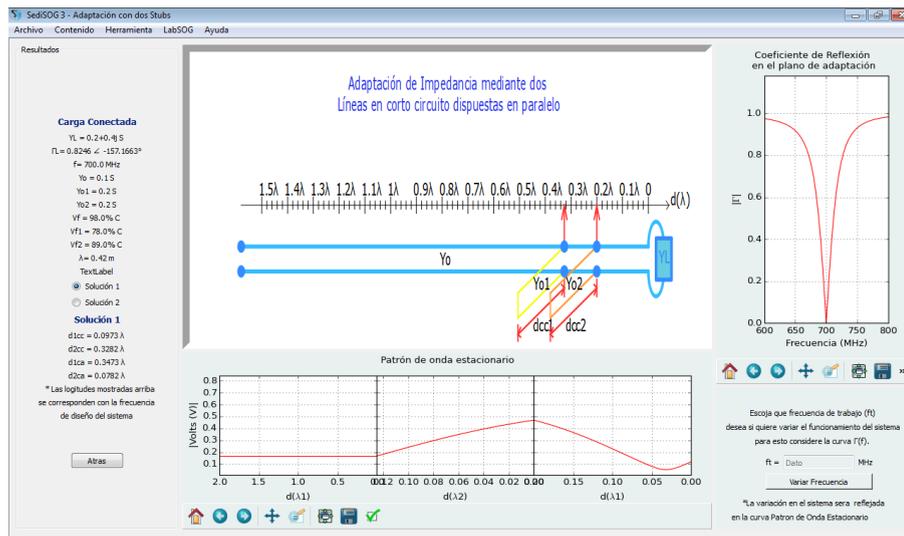


Figura 4.24: Interfaz de resultados del modulo adaptador de impedancias mediante dos stub

4.2.10. Ventana para el módulo balun con elementos concentrados

Esta ventana ofrece un calculador de los parámetros propios de un sistema Balun de elementos concentrados, sirviendo al usuario de herramienta para afrontar la complejidad matemática del sistema. Posee una interfaz inicial donde se encuentran campos donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño y una interfaz secundaria de resultados en donde se exhiben los parámetros de salida.

En la figura 4.25 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.26 se muestra la interfaz que contiene los resultados.

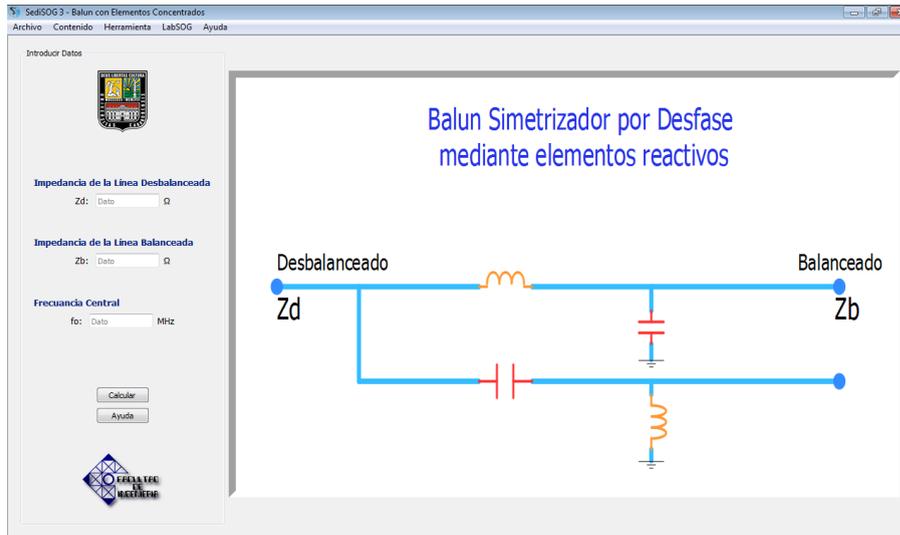


Figura 4.25: Interfaz de introducción de datos del módulo balun de elementos concentrados

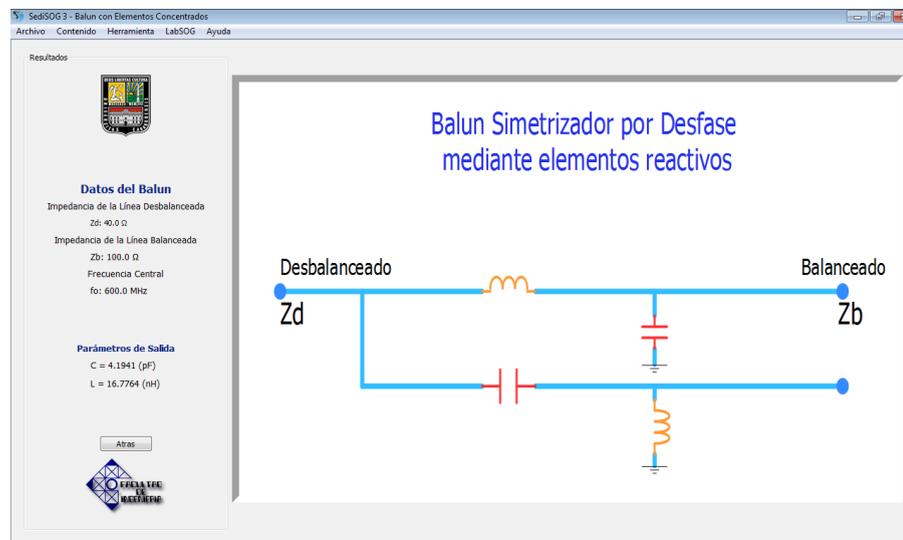


Figura 4.26: Interfaz de resultados del modulo balun de elementos concentrados

4.2.11. Ventana para el módulo balun con elementos distribuidos

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un sistema Balun de elementos distribuidos, sirviendo al usuario de herramienta para afrontar la complejidad visual y matemática del sistema. Posee una interfaz de inicio donde se escoge si la adaptación es de un sistema balanceado a uno desbalanceado o viceversa, acompañado por los campos donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño. La segunda interfaz es de resultados, esta exhibe los parámetros de salida acompañados de dos gráficas donde se muestran el patrón de onda estacionario de voltaje del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del modulo del coeficiente de reflexión en la sección adaptada de la línea en un ancho de banda de 10 %, además posee un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura 4.27 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.28 se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con el campo para variar la frecuencia.

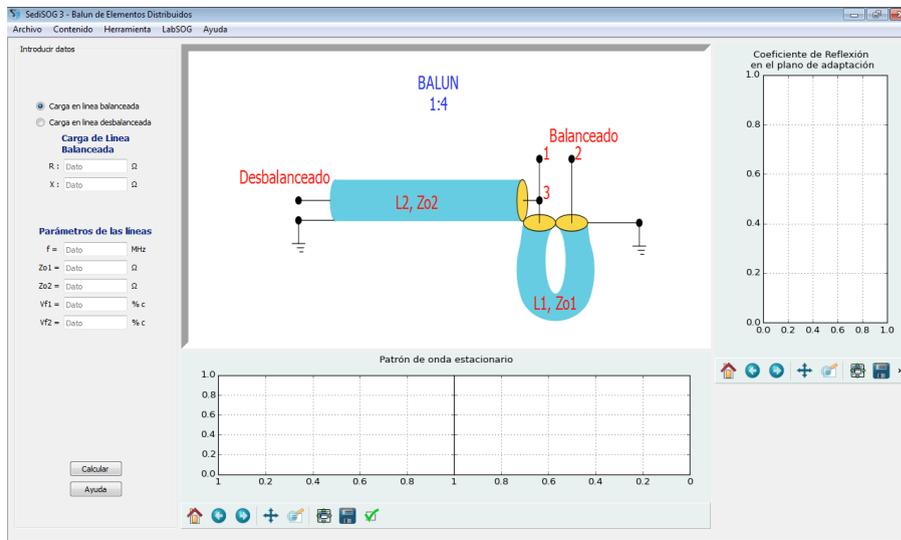


Figura 4.27: Interfaz de introducción de datos del modulo balun de elementos distribuidos

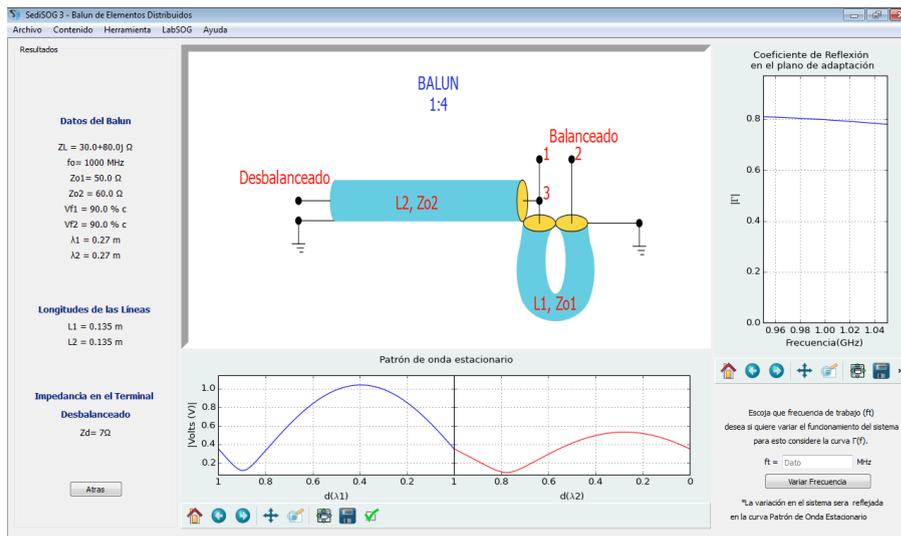


Figura 4.28: Interfaz de resultados del modulo balun de elementos distribuidos

4.2.12. Ventana para el módulo Carta de Smith

Con esta herramienta se puede resolver diferentes problemas de Sistemas de Ondas Guiadas de forma gráfica por medio de una Carta de Smith, estos problemas pueden ser: determinar la impedancia en cualquier punto de la Línea de trans-

misión, determinar los parámetros necesario para adaptar con una y dos líneas terminadas en corto circuito (Stubs) y adaptar con elementos concentrados. Cada problema tiene la opción de ser resultado por un proceso animado que indica el procedimiento detallado para cada paso de la animación.

En la figura 4.29 se puede observar la estructura de interfaz Carta de Smith, que cuenta con un menú, una Carta de Smith ubicada en el centro y dos subventanas, una para mostrar los detalles de un punto ubicado en la carta y la otra para indicar al usuario el procedimiento detallado para resolver el problema seleccionado.

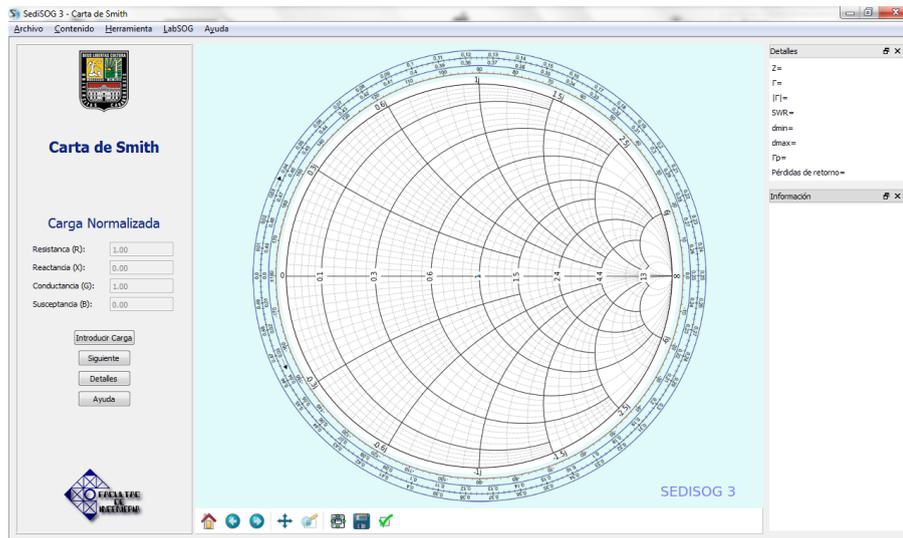


Figura 4.29: Interfaz para el módulo Carta de Smith

En la figura 4.30 se muestra la interfaz Carta de Smith con el resultado del proceso de adaptación de impedancia con elementos concentrados.

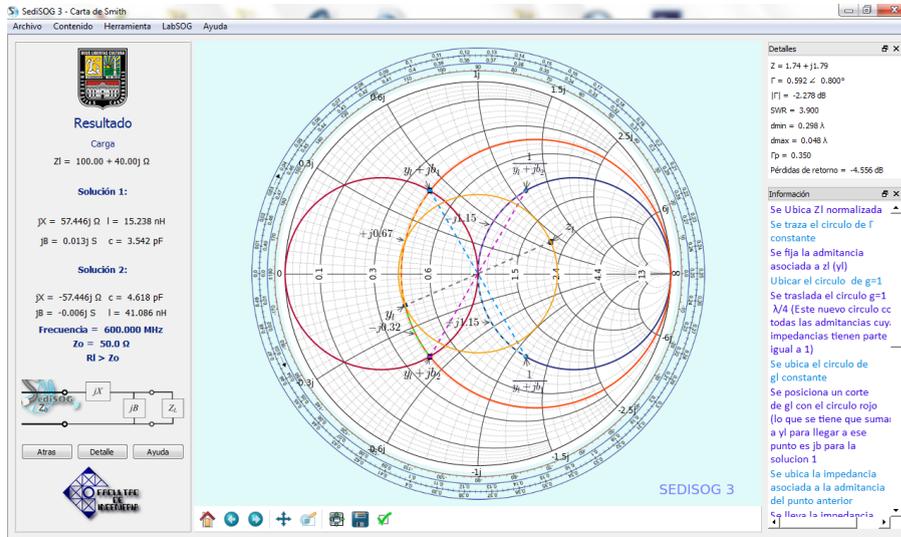


Figura 4.30: Adaptar impedancia con elementos concentrados mediante el uso de la Carta de Smith

En la figura 4.31 se muestra la ubicación del asistente que indica el procedimiento aplicado para llegar a la solución de la opción requerida por el usuario. Dicho asistente tiene la opción de observar en la Carta de Smith el paso que desee haciendo clic sobre el comentario.

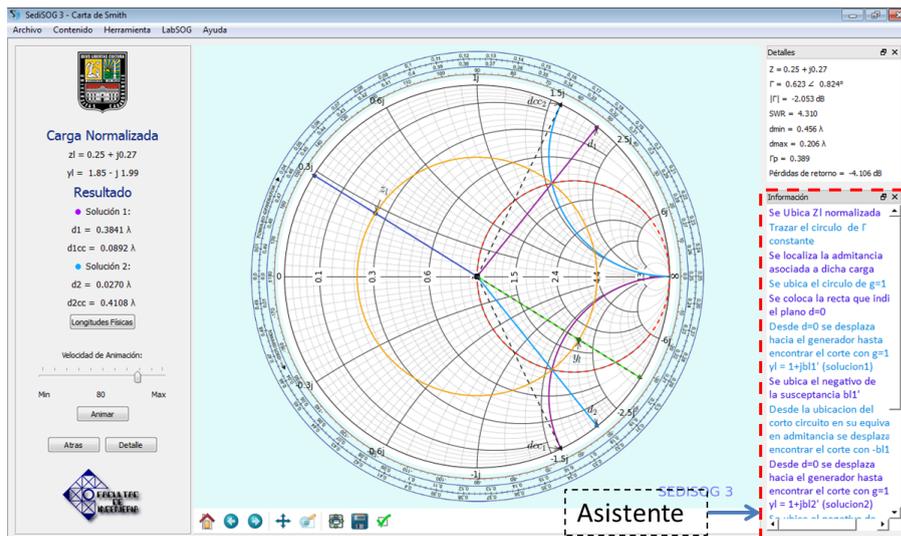


Figura 4.31: Asistente del módulo Carta de Smith

4.2.13. Ventana para el módulo línea no uniforme

Esta ventana ofrece un calculador y graficador de los parámetros y curvas propios de un sistema adaptador de impedancias, línea no uniforme, sirviendo al usuario de herramienta para afrontar la complejidad visual y matemática del sistema. Posee una interfaz de inicio donde se escoge si la carga esta en el extremo derecho o izquierdo de la línea no uniforme (teniendo por referencia el esquema ofrecido en las dos interfaces del modulo), acompañado por los campos donde se introducen los parámetros que caracterizaran el sistema de adaptación a la frecuencia de diseño. La interfaz de resultados es donde se exhiben los parámetros de salida acompañados de dos gráficas donde se muestran el patrón de onda estacionario de impedancia del sistema a la frecuencia de trabajo escogida y la respuesta en frecuencia del modulo del coeficiente de reflexión en la sección adaptada de la línea en un ancho de banda pre-definido, además posee un campo para variar la frecuencia de trabajo del sistema dentro del ancho de banda definido.

En la figura [4.32](#) se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura [4.33](#) se muestra la interfaz que contiene los resultados y las curvas mencionadas junto con el campo para variar la frecuencia.

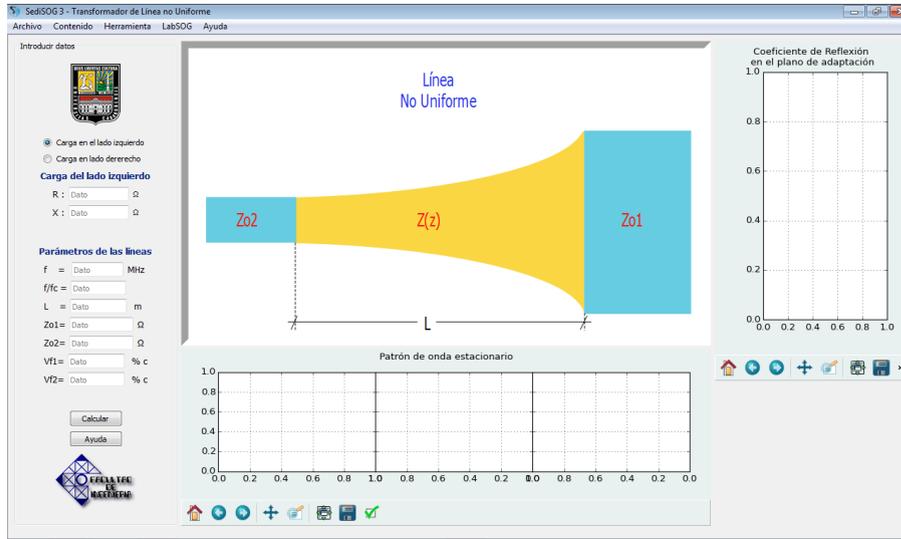


Figura 4.32: Interfaz de introducción de datos del modulo Línea No Uniforme

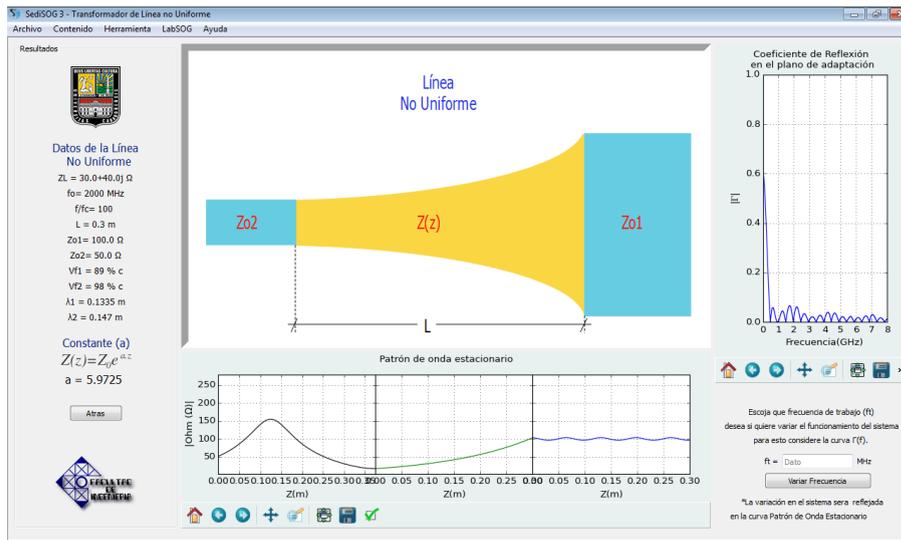


Figura 4.33: Interfaz de resultados del modulo Línea No Uniforme

4.2.14. Ventana para el módulo parámetros de dispersión

Los resultados de este modulo sirven al usuario para conocer los parámetros S, Z, Y y ABCD sin la necesidad de hacer cálculos. Posee una interfaz de inicio donde se encuentran campos donde se introducen los parámetros que caracterizaran

el sistema a la frecuencia de diseño, luego una segunda interfaz de resultados en donde se exhiben los parámetros de salida.

En la figura 4.34 se muestra la interfaz que contiene los campos de entrada donde el usuario puede introducir o modificar parámetros del sistema.

En la figura 4.35 se muestra la interfaz que contiene los resultados.

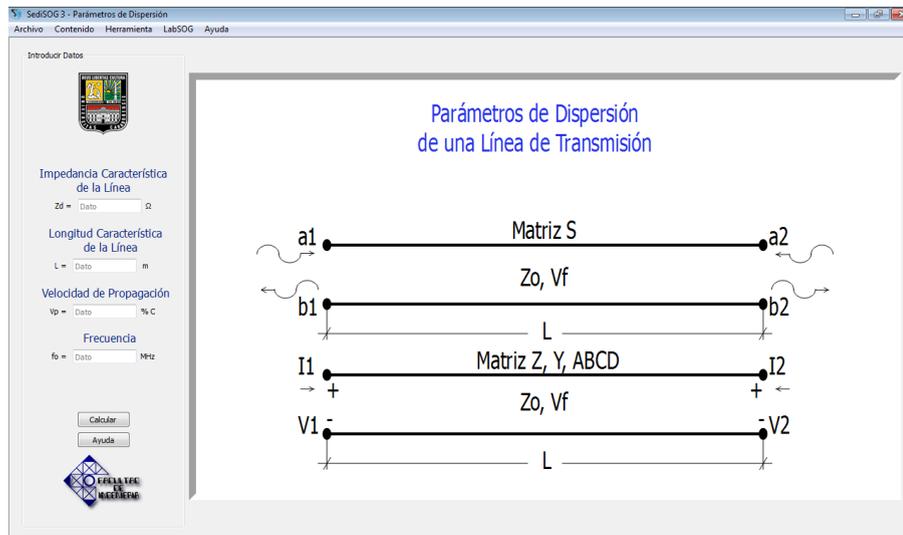


Figura 4.34: Interfaz de introducción de datos del modulo Parámetros de Dispersión de una Línea de transmisión

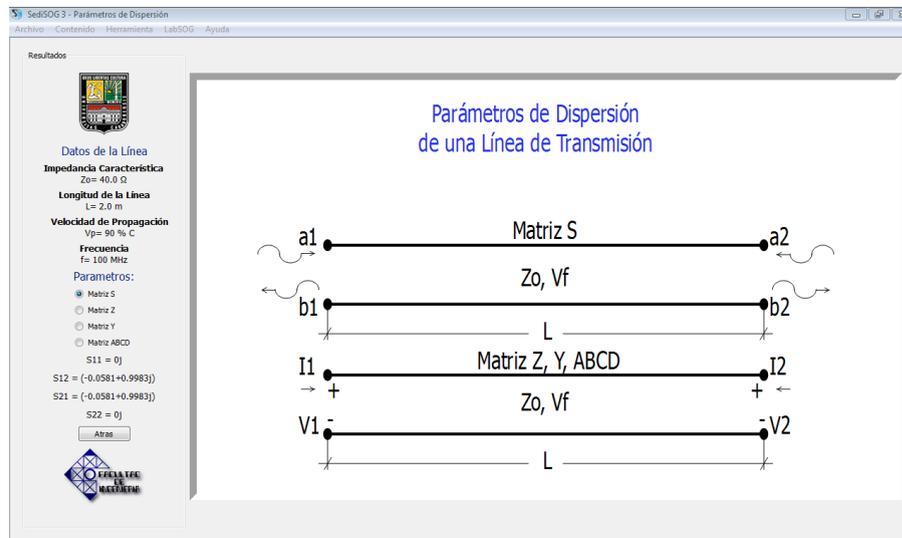


Figura 4.35: Interfaz de resultados del modulo Parámetros de Dispersión de una Línea de transmisión

4.2.15. Ventana para el laboratorio virtual de SOG

En la barra de herramientas se encuentra el ítem LabSOG, la cual, al ser seleccionado despliega un menú con la opción Laboratorio Virtual, como se muestra en la figura 4.36. Otra opción de acceder a esta interfaz es presionando Ctrl+L simultáneamente.

En el laboratorio virtual se presenta un entorno donde los instrumentos y accesorios del laboratorio de la asignatura fueron simulados con el fin de ofrecer al usuario una interacción mas real al momento de realizar las prácticas pertinentes a la materia. En la figura 4.38 se muestra la apariencia de la interfaz para este módulo el cual cuenta con dos secciones: la sección de herramientas y la sección de trabajo, en la primera se encuentran los elementos a utilizar, al momento de realizar la práctica, mientras que la otra sección es donde se realiza el montaje de las prácticas mediante la conexión de los objetos seleccionados en el bloque anterior. Además la interfaz ofrece botones para la navegación dentro de ella, tal como: borrar y mover.

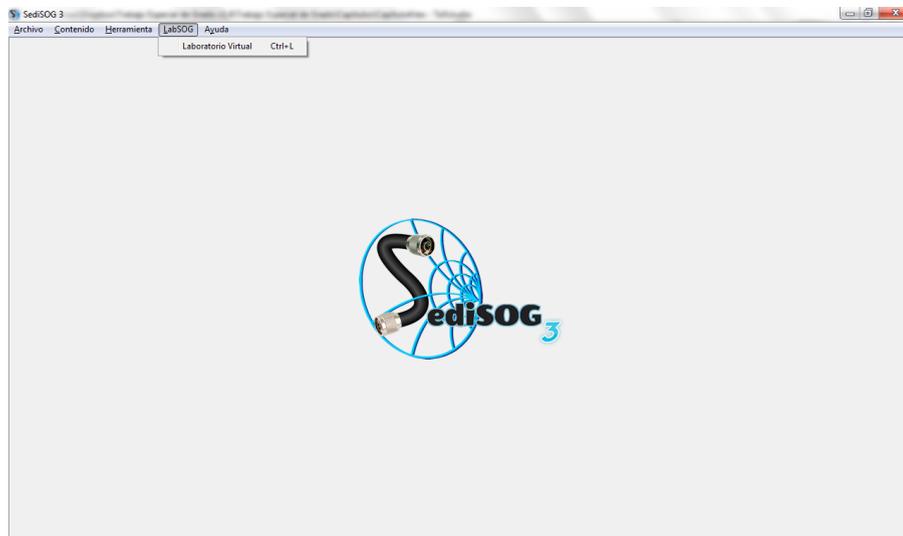


Figura 4.36: Menú LabSOG



Figura 4.37: Interfaz del Laboratorio Virtual

En la figura 4.38 se muestra el montaje de la práctica "adaptación de impedancia con dos stubs".

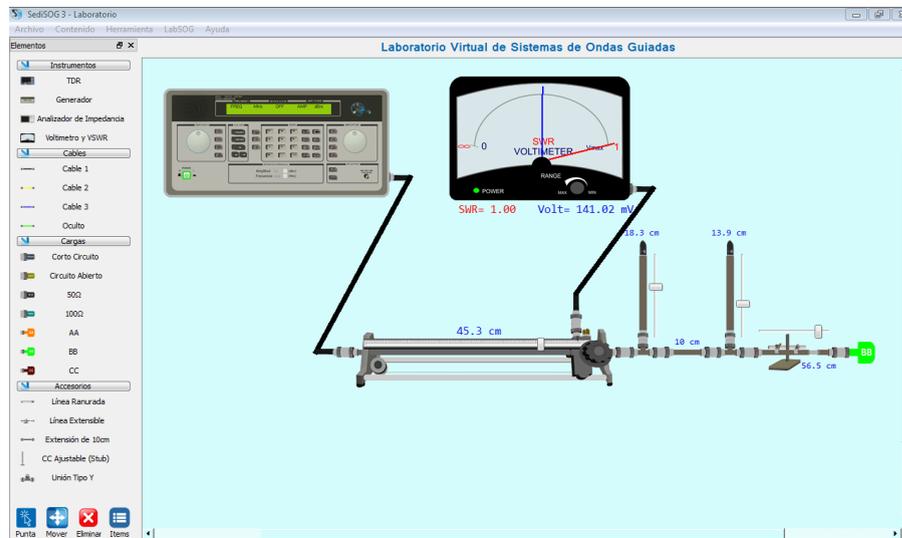


Figura 4.38: Montaje en el Laboratorio Virtual

Las practicas que se pueden desarrollar en este laboratorio virtual son las siguientes:

- Reflectometría en el dominio temporal.
- Caracterización de una línea de transmisión.
- Medición de impedancia con la línea ranurada
- Adaptación con una línea terminada en cortocircuito.
- Adaptación con dos líneas terminadas en cortocircuito

Para realizar cada una de estas practicas el usuario debe ser capaz de seleccionar los equipos y accesorios necesarios, además de conectarlos correctamente para lograr el objetivo del ensayo. Debido a este enfoque el estudiante tendrá mayor autonomía y versatilidad para poner en práctica los conceptos vistos en la asignatura y así tener mejor rendimiento al momento de dirigirse al laboratorio de sistemas de ondas guiadas.

4.3. Requerimientos del sistema

Para ejecutar el programa en un computador se creó un archivo "setup", el cual realiza un proceso de instalación en el sistema operativo Windows, fácil para el usuario generando un icono en el escritorio del PC. Mientras que para el sistema operativo Linux se generó un archivo BASH, el cual acondiciona el computador para que la aplicación pueda ejecutarse con éxito.

4.3.1. Requerimientos de hardware

Los requerimientos mínimos:

- Procesador Intel Atom 1600 MHz
- 1 GB memoria RAM
- 200 MB disponibles en el disco duro
- Mouse y teclado

4.3.2. Requerimientos de software

- Windows:
 - A partir de Windows XP hasta Windows 10.
- Ubuntu (Linux):
 - A partir de Ubuntu 12.04 hasta Ubuntu 16.04.

Capítulo V

Conclusiones y recomendaciones

5.1. Conclusiones

- Los paquetes de Python seleccionados para desarrollar la aplicación SEDI-SOG 3, cumplieron con los requerimientos establecidos, tales como: diseño visual más actualizado, de fácil aprendizaje, capaz de generar animaciones de forma sencilla y visualizarlas en una interfaz gráfica de usuario.
- Los algoritmos y códigos generados en python, haciendo uso del método de FDTD, permitieron recrear y simular los transitorios presentes en un arreglo de líneas de transmisión, con diferentes tipos de fuente (pulso gaussiano, escalón, trapezoidal y sinusoidal), discontinuidades (R serie y paralelo) y varios tipos de carga (R, RL serie y paralelo, RC serie y paralelo).
- El módulo transitorios ahora cuenta con la posibilidad de realizar simulaciones para arreglos de líneas de transmisión tales como: una línea, dos líneas una tras otra, tres líneas dispuestas de la misma forma y tres líneas dispuestas en paralelo, los tres primeros tipos de arreglo tienen la opción de añadir una discontinuidad resistiva en cualquier posición del arreglo. Este módulo además cuenta con un diagrama de reflexiones animado, para cada tipo de arreglo que se desee simular.

- Se añadieron los módulos transformador de línea no uniforme y balun de elementos distribuidos.
- Los módulos Transformador de Cuarto de Onda, Transformador Binomial, Transformador Chebyshev, Adaptación con un Stub, Adaptación con dos Stubs, permiten visualizar el comportamiento del valor absoluto del coeficiente de reflexión en función de la frecuencia en el plano de adaptación, además de observar el patrón de ondas estacionario existente en el sistema bajo estudio.
- Al módulo Carta de Smith, se añadió la opción de adaptar con elementos concentrados, además a todas las opciones presentes en este módulo, cuentan con un asistente que indica el procedimiento aplicado, para la resolución del problema.
- El laboratorio virtual permite al usuario realizar prácticas de forma interactiva, con mayor grado de autonomía y versatilidad, para así tener una experiencia más realista y sin problemas de imperfecciones, en los equipos y herramientas existentes en el laboratorio de la asignatura.
- El programa SEDISOG 3 ofrece al usuario un completo contenido para el estudio y análisis relacionado con líneas de transmisión. Además sirve de apoyo para complementar los esfuerzos docentes en la asignatura "Sistemas de Ondas Guiadas".

5.2. Recomendaciones

- El módulo asociado al laboratorio virtual se puede mejorar el estilo de mover y realizar las conexiones de un montaje, para así ofrecer al usuario mayor facilidad y dinamismo en el momento de usar la herramienta.
- Crear este tipo de herramientas para asignaturas de la escuela de Ingeniería Eléctrica e Ingeniería de telecomunicaciones, las cuales requieren de un alto nivel de razonamiento abstracto, tal como: Teoría Electromagnética I y II, Microondas, entre otras.

- Antes de ingresar al laboratorio de la asignatura, se debe requerir a los cursantes de la misma, una revisión al programa, para tener una noción mas clara al momento de realizar un ensayo, lo que permite hacer uso eficiente del tiempo, en el laboratorio.

Referencias Bibliográficas

- [1] Mejías, Carlos y Hernández, Orangel. *Desarrollo de un Software Educativo Didáctico Interactivo para Sistemas de Ondas Guiadas Versión 2*. Universidad de Carabobo. 2008.
- [2] Aldo N Bianchi. *Sistemas de ondas guiadas*. 1980.
- [3] Paulino Del Pino. «Estudio y Aplicacion al electromagnetismo del metodo de las diferencias finitas en el dominio del tiempo FDTD.» Trabajo de Ascenso. Universidad de Carabobo, 2008.
- [4] Mark Lutz. *Programming python*. Vol. 8. O'Reilly, 1996.
- [5] Guido van Rossum. *Python 2.7.10 documentation*. Sep. de 2016. URL: <https://docs.python.org/2/>.
- [6] Guido van Rossum. *History and License*. 2001. URL: <https://docs.python.org/3/license.html>.
- [7] Raúl González Duque. «Python para todos». En: *Creative Commons Reconocimiento 2* (2006).
- [8] Numpy Developers. *Numpy*. URL: <http://www.numpy.org/>.
- [9] *Matplotlib*. The matplotlib development team. Sep. de 2016. URL: <http://matplotlib.org/>.
- [10] *PyQt*. Riverbank Computing Limited. 2016. URL: <https://riverbankcomputing.com/software/pyqt/intro>.
- [11] David M Pozar. *Microwave engineering*. 4ta. John Wiley & Sons, 2009.

-
- [12] James R Nagel. «The Finite-Difference Time-Domain (FDTD) Algorithm». En: *Department of Electrical And Computer Engineering, University of Utah*. Dostupné z WWW: http://ece.utah.edu/~ece3300/Labs/lab4/FDTD_Introduction.pdf. [cit. 2012-04-20] (2010).
- [13] Allen Taflove y Susan C Hagness. *Computational electrodynamics*. Artech house publishers, 2000.
- [14] Jan Carlsson. «A FDTD program for computing responses on branched multi-conductor transmission lines». En: *SP Rapport* (1998), pág. 16.