

APROBACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: “**Interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica**”, del señor **Diego Rafael Viera Galarza**, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, mayo 2018.

TUTOR



Ing. Santiago Altamirano, Mg.

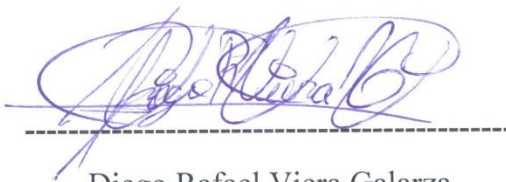
DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, mayo 2018.

AUTOR



Diego Rafael Viera Galarza

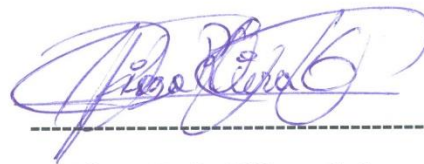
C.I.: 180463703-9

AUTORÍA

El presente Proyecto de Investigación titulado: **“Interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, mayo 2018.

AUTOR



Diego Rafael Viera Galarza

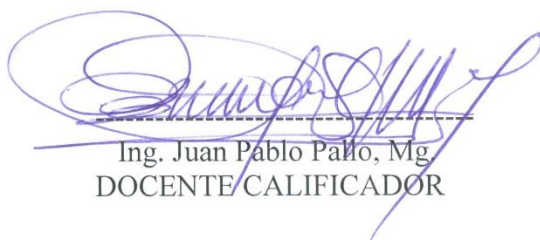
C.I.: 180463703-9

APROBACIÓN DE LA COMISIÓN CALIFICADORA


La Comisión Calificadora del presente trabajo conformada por los señores docentes **Ing. Juan Pablo Pallo** e **Ing. Carlos Gordón**, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “**Interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica**”, presentado por el señor **Diego Rafael Viera Galarza** de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Pilar Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL



Ing. Juan Pablo Pallo, Mg.
DOCENTE CALIFICADOR



Ing. Carlos Gordón, Mg.
DOCENTE CALIFICADOR

DEDICATORIA

A Dios por darme la vida, a mi familia por su apoyo incondicional en cada lucha de mi vida diaria, por guiarme por el camino correcto y estar presentes en los momentos más difíciles.

Diego Rafael Viera Galarza

AGRADECIMIENTO

A Dios por ser mi fortaleza y guiar mis pasos a lo largo de mi vida.

A mis padres por su dedicación, su paciencia y su amor, que han forjado en mí un hombre de bien.

A mis hermanos que han sido un soporte en los buenos y malos momentos.

Al Ing. Santiago Altamirano por su tiempo, apoyo y paciencia brindada a lo largo del presente proyecto.

Diego Rafael Viera Galarza

ÍNDICE GENERAL

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DE LA COMISIÓN CALIFICADORA	v
DEDICATORIA	vi
AGRADECIMIENTO.....	vii
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	xii
ÍNDICE DE TABLAS	xvi
RESUMEN	xvii
SUMMARY	xviii
GLOSARIO	xix
ACRÓNIMOS	xxii
INTRODUCCIÓN	xxiv
CAPÍTULO I	1
1.1. TEMA	1
1.2. PLANTEAMIENTO DEL PROBLEMA	1
1.3. DELIMITACIÓN	2
1.3.1 Delimitación de Contenido	2
1.3.2 Delimitación Espacial	2
1.3.3 Delimitación Temporal	2
1.4. JUSTIFICACIÓN	3

1.5. OBJETIVOS	4
1.5.1 Objetivo General	4
1.5.2 Objetivos Específicos	4
CAPÍTULO II	5
2.1. ANTECEDENTES INVESTIGATIVOS	5
2.2. FUNDAMENTACIÓN TEÓRICA	7
2.2.1 Breve descripción de los sistemas de comunicación por fibra óptica.....	7
2.2.2 Fundamentación de los sistemas de comunicaciones por fibra óptica	11
2.2.3 Parámetros fundamentales de un enlace de fibra óptica	30
2.2.4 Conceptos matemáticos en las comunicaciones por fibra óptica	32
2.2.5 Teoría de propagación en fibra óptica	34
2.2.6 Diseño del ancho de banda	39
2.2.7 Presupuesto de tiempo de subida	41
2.2.8 Ruido	43
2.2.9 Relación señal ruido	44
2.2.10 Reflectometría y transmisometría ópticas	44
2.2.11 Interferometría óptica	47
2.2.12 Reflectómetro óptico en el dominio del tiempo	50
2.2.13 Mediciones del OTDR	57
2.2.14 Software de proceso matemático	61
2.3. PROPUESTA DE SOLUCIÓN	69
CAPÍTULO III	70
3.1. MODALIDAD DE LA INVESTIGACIÓN	70
3.2. POBLACIÓN Y MUESTRA	70
3.3. RECOLECCIÓN DE INFORMACIÓN	70

3.4. PROCESAMIENTO Y ANÁLISIS DE DATOS	71
3.5. DESARROLLO DEL PROYECTO	71
CAPÍTULO IV	73
4.1. INTRODUCCION	73
4.2. ANÁLISIS DE FACTIBILIDAD	74
4.2.1 Factibilidad Operativa	74
4.2.2 Factibilidad Temporal	74
4.2.3 Factibilidad Técnica	74
4.2.4 Factibilidad Económica	75
4.2.5 Factibilidad Bibliográfica	75
4.2.6 Proyección a Futuro	75
4.3. FUNDAMENTACIÓN CIENTÍFICO-TÉCNICA	75
4.3.1 Recomendaciones de la UIT-T para fibras ópticas	75
4.3.2 Nomenclatura	77
4.3.3 Selección de Software para el SOTDR	78
4.3.4 Requerimientos de SOTDR	78
4.3.5 Diseño de la Interfaz Gráfica de Usuario (GUI) de SOTDR	81
4.3.6 SOTDR	90
4.3.7 Configuración simulador SOTDR	98
4.4. PRUEBAS Y RESULTADOS	100
4.5. PRESUPUESTO DEL PROYECTO	110
4.5.1 Costo de Hardware y software	110
CAPÍTULO V	111
5.1. CONCLUSIONES	111
5.2. RECOMENDACIONES	112

BIBLIOGRAFÍA	114
ANEXOS	119

ÍNDICE DE FIGURAS

Figura 2.1: Símbolo de la fibra óptica	8
Figura 2.2: Símbolo del transmisor óptico	10
Figura 2.3: Símbolo del receptor óptico	11
Figura 2.4: Diagrama de bloques de un sistema de comunicación óptico	12
Figura 2.5: Elementos de un sistema de comunicación óptica	13
Figura 2.6: Estructura de una fibra óptica	14
Figura 2.7: Fibra óptica multimodal (MM)	15
Figura 2.8: Fibra óptica monomodo (SM)	15
Figura 2.9: Diagrama de bloque de un transmisor óptico	16
Figura 2.10: Diagrama de un transmisor óptico de modulación interna	16
Figura 2.11: Diagrama de un transmisor óptico de modulación externa	17
Figura 2.12: Diodo emisor de luz de tipo lateral.....	17
Figura 2.13: Diodo láser.....	18
Figura 2.14: Diagrama esquemático del láser FP	18
Figura 2.15: Diagrama esquemático del láser DBF	19
Figura 2.16: Diagrama esquemático del láser FP.....	19
Figura 2.17: Clasificación de la fibra óptica	21
Figura 2.18: Fibra óptica multimodo de índice escalonado	21
Figura 2.19: Fibra óptica multimodo de índice gradual.....	22
Figura 2.20: Fibra óptica monomodo.....	23
Figura 2.21: Procedimiento de empalme en fibra óptica	24
Figura 2.22: Posibles errores al empalmar fibras.....	26
Figura 2.23: Esquema de algunas de las pérdidas intrínsecas en fibras ópticas	27
Figura 2.24: Forma de varios conectores de fibra óptica	30

Figura 2.25: Longitudes Operativas de Fibras	31
Figura 2.26: Cono de Aceptación	34
Figura 2.27: Tipos de Señalización.....	40
Figura 2.28: Tipos de dispersión en la fibra	42
Figura 2.29: Dispersión cromática en la fibra óptica	42
Figura 2.30: Óptica geométrica	45
Figura 2.31: Patrón de interferencia generado por un interferómetro de Michelson .	47
Figura 2.32: Detección directa	48
Figura 2.33: Detección coherente	49
Figura 2.34: Implementación en fibra óptica del interferómetro de Mach-Zehnder..	49
Figura 2.35: Modelo de OTDR de la familia JDSU T-BERD/MTS.....	51
Figura 2.36: Efectos de la macro flexión sobre una fibra	54
Figura 2.37: Efectos de la dispersión desplazada	55
Figura 2.38: Medición de la fibra utilizando diferentes anchos de pulso	56
Figura 2.39: Evento reflectante debido a un cruce del conector	58
Figura 2.40: Evento de reflexión en dos uniones de conectores muy cercanos.....	59
Figura 2.41: Evento de reflexión en un extremo de la fibra	59
Figura 2.42: Evento de no reflexión	60
Figura 2.43: Interfaz de MATLAB	63
Figura 2.44: Interfaz de SIMULINK	64
Figura 2.45: Interfaz de LABVIEW.....	65
Figura 2.46: Interfaz de SCILAB.....	66
Figura 2.47: Interfaz de GNU Octave	67
Figura 2.48: Interfaz de MAPLE	68
Figura 4.1: Esquema del OTDR implementado en MATLAB	79

Figura 4.2: Resultado del ejemplo con los cursores añadidos	80
Figura 4.3: Resultado de la medida de la atenuación en un Km	81
Figura 4.4: Ampliación de la zona a medir	81
Figura 4.5: Diagrama de bloques simulador SOTDR	82
Figura 4.6: Interacción entre herramientas de MATLAB	83
Figura 4.7: Algoritmo de la interfaz principal	84
Figura 4.8: Algoritmo elección de componentes	84
Figura 4.9: Algoritmo para codificar la simulación en GUIDE	85
Figura 4.10: Algoritmo del presupuesto de potencia y atenuación	86
Figura 4.11: Interfaz PORTADA del simulador SOTDR	87
Figura 4.12: Interfaz ENLACE ÓPTICO del simulador SOTDR	88
Figura 4.13: Interfaz SIMULADOR SOTDR	89
Figura 4.14: Interfaz AYUDA del simulador SOTDR	89
Figura 4.15: Parámetros del enlace óptico	90
Figura 4.16: Datos de los componentes	91
Figura 4.17: GUI Transmisor	91
Figura 4.18: GUI Fibra Óptica	92
Figura 4.19: GUI Receptor	93
Figura 4.20: Empalmes y Conectores	93
Figura 4.21: Resultados	94
Figura 4.22: Botón Simulador OTDR	95
Figura 4.23: GUI SOTDR	95
Figura 4.24: Cuadro de diálogo botón Salir	96
Figura 4.25: Botón Presupuesto Potencia y Atenuación	97
Figura 4.26: GUI Presupuesto Potencia	97

Figura 4.27: Resultado de la simulación uno en SOTDR	101
Figura 4.28: Cuadro resumen resultados de interés	102
Figura 4.29: Verificación de la atenuación por medio de los cursores	102
Figura 4.30: Segunda verificación diferencia entre cursores	103
Figura 4.31: Medida de la atenuación del conector de entrada.....	103
Figura 4.32: Diferencia entre cursores conector de entrada.....	103
Figura 4.33: Comprobación de la longitud de fábrica.....	104
Figura 4.34: Comprobación de la longitud de fábrica.....	104
Figura 4.35: Resultado de la simulación uno, segunda parte.....	105
Figura 4.36: Atenuación de la fibra diferencia de cursores	105
Figura 4.37: Resultado de la simulación dos	106
Figura 4.38: Resultado de la simulación dos, segunda parte	107
Figura 4.39: Resultado de la simulación tres	109
Figura 4.40: Resultado de la simulación tres, segunda parte	110

ÍNDICE DE TABLAS

Tabla 2.1: Conectores comunes, considerando sus pérdidas de inserción y de retorno	29
Tabla 2.2: Cuadro comparativo entre sistemas de cálculo numérico.....	62
Tabla 4.1: Resumen de estándares	76
Tabla 4.2: Longitud de onda operativa para fibra monomodo y multimodo	98
Tabla 4.3: Coeficiente de atenuación de la fibra óptica	99
Tabla 4.4: Conectores y su atenuación	100
Tabla 4.5: Datos principales para la simulación uno	101
Tabla 4.6: Datos principales para la simulación uno, segunda parte	104
Tabla 4.7: Datos principales para la simulación dos	106
Tabla 4.8: Datos principales para la simulación dos, segunda parte	107
Tabla 4.9: Datos principales para la simulación tres	108
Tabla 4.10: Datos principales para la simulación tres, segunda parte	109
Tabla 4.11: Tabla de presupuesto	110

RESUMEN

El propósito de esta investigación es la construcción de una interfaz gráfica para simulación de un OTDR (Reflectómetro Óptico en el Dominio del tiempo) aplicado en sistemas de comunicaciones por fibra óptica, esta interfaz permite diseñar un enlace de fibra óptica de forma sencilla, permitiendo además medir desde la atenuación de la fibra a la de un empalme; además, corrobora conceptos teóricos haciendo hincapié en los principales errores de un enlace de fibra óptica real.

SOTDR integra la simulación de variados estándares UIT-T para fibra ópticas monomodo y multimodo, así como su propagación y recepción con las diversas atenuaciones de los respectivos componentes que existen en un sistema real. El cálculo de la respuesta del sistema se realiza en el dominio del tiempo confiriendo al prototipo la flexibilidad suficiente para incorporar tipos de fuentes, adaptándose a las necesidades de simulación de un equipo OTDR comercial. Asimismo, la correcta programación de los módulos de visualización, cálculos y procesamiento de información ha permitido contrastar los resultados de SOTDR con los teóricos utilizando ejemplos sencillos.

SOTDR además de emular en gran medida la funcionalidad de un OTDR, permite evaluar enlaces de fibra óptica para garantizar la posterior implementación de las conexiones de forma factible. El prototipo propone la planificación de enlaces de fibra óptica al estar diseñado en un entorno de Interfaz Gráfica de Usuario GUI; intuitivo y amigable al operador; la implementación del trabajo de investigación concluyó poniéndolo a prueba bajo configuraciones reales de enlaces de fibra óptica.

Palabras clave: Interfaz, MATLAB, OTDR, fibra óptica, simulación, comunicaciones.

SUMMARY

The purpose of this investigation is the construction of a graphical interface for simulation of an OTDR (Optical Time Domain Reflectometer) applied in fiber optic communication systems, this interface allows to design a fiber optic link in a simple way, also allowing measure from the attenuation of the fiber to that of a splice. This incorporates an interface to design a fiber optic link; In addition, it corroborates theoretical concepts emphasizing the main errors of a real fiber optic link.

SOTDR integrates the simulation of various ITU-T standards for single mode and multimode optical fiber, as well as its propagation and reception with the different attenuations of the respective components that exist in a real system. The calculation of the response of the system is done in the time domain, giving the prototype enough flexibility to incorporate types of sources, adapting to the simulation needs of a commercial OTDR device. Likewise, the correct programming of the visualization, calculation and information processing modules has allowed to contrast the results of SOTDR with the theoretical ones using simple examples.

SOTDR, in addition to emulating to a large extent the functionality of an OTDR, allows the evaluation of fiber optic links to guarantee the subsequent implementation of the connections in a feasible way. The prototype proposes the planning of fiber optic links, as it is designed in a graphical user interface GUI environment; intuitive and friendly to the operator; the implementation of the research work concluded by putting it to the test under real configurations of fiber optic links.

Keywords: Interface, MATLAB, OTDR, fiber optics, simulation, communications.

GLOSARIO

Arámidas: Material ligero, normalmente amarillo o naranja que proporciona fortaleza y soporte a los mazos de fibra en el cable. El Kevlar es un tipo de hebras de arámida que posee una resistencia mecánica muy elevada.

Atenuación: Disminución de luz en una fibra óptica. Se expresa generalmente sin su signo negativo en dB o dB/Km. Cuando se especifica la atenuación., es muy importante indicar la longitud de onda utilizada. La atenuación en una fibra óptica es diferente para distintas longitudes de onda.

Banda ancha: Régimen de datos igual o superior a 45 Mbps (o DS3).

BER: Tasa de Error de Bit. Relación de bits recibidos con error respecto a los bits enviados. Es normal un BER de 10^{-7} (un bit de error recibido sobre mil millones de bits enviados).

BPON: Broadband PON. Estándar para PON establecido por la ITU (ITU-T G.983).

BROADCAST: En un sistema de cable, son aquellas señales de Radio Frecuencia comunes para todos los suscriptores, típicamente de video.

Buffer: Cubierta protectora de plástico o de otro material, comúnmente codificado mediante colores, que recubre las fibras ópticas. Un buffer puede ir ajustado, como en el caso de un cable adherido directamente al recubrimiento de la fibra óptica, o puede ir flojo, como en un entubado de cable, donde una o más fibras entrelazadas van holgadas por el interior del tubo buffer. El buffer debe ser pelado para efectuar los cortes de la fibra y sus empalmes.

Caja de empalmes: Recinto cerrado para albergar los empalmes de la fibra óptica y las bandejas de empalmes.

Conector: Dispositivo que une dos fibras ópticas, de manera repetible y con bajas pérdidas ópticas de conexión.

CWDM: Coarse Wave Division Multiplexing. División por multiplexación gruesa de longitud de onda.

dB: Decibelio. Medida logarítmica de la potencia óptica.

dBm: Decibelio referido a un miliwatio; $\text{dBm} = 10 \log (\text{salida potencia miliwatos} / 1 \text{ miliwatio})$.

DWDM: Dense Wave Division Multiplexing. División por multiplexación densa de longitud de onda.

EDFA: Erbium Doped Fiber Amplifier. Amplificador óptico dopado con Erbio.

Empalme: Unión permanente entre dos fibras originada por la fusión de sus extremos o por el método de unión mecánica.

Equipo óptico: Cualquier equipo electrónico de comunicaciones utilizado para comunicaciones mediante fibra óptica; también se conocen como equipos terminales ópticos o módems ópticos.

Férula: Porción rígida del centro de un conector de fibra óptica, que usualmente suele ser de acero o cerámico.

Fibras ópticas: Transmisión de luz a través de fibras ópticas para comunicaciones y señalización.

Fotodetector: Dispositivo que convierte la energía luminosa en energía eléctrica. Los fotodiodos habitualmente utilizados con las fibras ópticas son de silicio.

FTTH: Fiber to the Home. Fibra instalada hasta el interior de la vivienda. Arquitectura de fibra óptica donde se ha alcanzado directamente el interior del domicilio del usuario.

Fusión: Unión permanente de dos fibras ópticas realizada mediante una fusionadora o empalmadora de manera mecánica o automática a altas temperaturas.

GPON: Gigabit-capable Passive Optical Networks. Se refiere a un estándar para PON establecido por la ITU (ITU-T G.984.1).

Polietileno: Material termoplástico utilizado a menudo para el recubrimiento de los cables.

PON: Passive Optical Network. Red pasiva de fibra óptica.

Receptor óptico: Unidad electrónica que convierte las señales de luz en señales eléctricas.

Recubrimiento: Capa fina de plástico o de otro material, usualmente de un diámetro de 250 o 500 μm , codificado con colores, que a su vez cubre el revestimiento de la fibra. La mayoría de las fibras poseen un recubrimiento que debe ser desgarrado y suprimido para efectuar el corte y unión de una fibra.

Reflexión: Efecto de reflejar un rayo de luz en la interfaz de dos medios diferentes hacia el primer medio.

Refracción: Cambio de dirección y velocidad de un rayo de luz en la interfaz de dos medios diferentes.

Revestimiento: Capa de cristal o de otro material con un índice de refracción bajo que cubre al núcleo de la fibra, que consigue que la luz permanezca confinada dentro de dicho núcleo.

Ruido: En telecomunicaciones, son aquellas alteraciones no deseadas, superpuestas a una señal útil, que tienden a ocultar el contenido informativo.

ACRÓNIMOS

ADC:	Analog to Digital Converter.
BER:	Bit Error Rate.
DSP:	Digital Signal Processor.
DWDM:	Dense Wavelength Division Multiplexing.
EDFA:	Erbium Doped Fiber Amplifier.
EPON:	Ethernet Passive Optical Network.
FC/APC:	Fiber Connector/Angle Physical Contact.
FC/PC:	Fiber Connector/Physical Contact.
FPGA:	Field Programmable Gate Array.
FTTB:	Fiber To The Building.
FTTC:	Fiber To The Curb.
FTTH:	Fiber To The Home.
GPON:	Gigabit-capable Passive Optical Network.
ISO:	International Standards Organization.
ITU:	International Telecommunication Union.
ITU-T:	ITU Telecommunication Standardization Sector.
LED:	Light Emitting Diodes.
MZ:	Mach Zehnder.
NRZ:	Non Return to Zero.
OCDM:	Optical Code Division Multiplexing.

ODN:	Optical Distribution Network.
OLT:	Optical Line Terminal.
ONU:	Optical Network Unit.
ONT:	Optical Network Terminal.
OTDR:	Optical Time Domain Reflectometry.
OTM:	Optical Transceiver Monitoring.
PON:	Passive Optical Network.
QoS:	Quality of Service.
RZ:	Return to Zero.
SN:	Shot Noise.
SNR:	Signal to Noise Ratio.
SRS:	Stimulated Raman Scattering.
TDM:	Time Division Multiplexing.
WDM:	Wavelength Division Multiplexing.

INTRODUCCIÓN

Los sistemas de comunicaciones por fibra óptica o comunicaciones lightwave (por onda de luz) es la tecnología preferida para la transmisión de voz, datos, telemetría y vídeo en larga distancia; además de redes de área local, es adecuado para televisión por cable. La tecnología lightwave ofrece la enorme capacidad del usuario de transmitir a distancias lejanas con inmunidad de la interferencia electromagnética, y la relativa facilidad de la instalación. Estos sistemas de ondas de luz que emplean fibras ópticas han revolucionado la tecnología detrás de las telecomunicaciones. De hecho, la tecnología de ondas de luz junto con la microelectrónica, son factores importantes en la tecnología de la información y las comunicaciones.

Para la gestión de un sistema de comunicaciones por fibra óptica es necesario la utilización y el desarrollo de dispositivos que ayuden a la localización de posibles fallos o averías, que colaboren en el mantenimiento de las instalaciones de fibra óptica. Uno de los principales aparatos que verifican el diseño óptimo de una instalación óptica, es el OTDR (Optical Time domain Reflectometer), un instrumento que emite pulsos de luz a la longitud de onda deseada, para medir el tiempo que tarda en recibir una reflexión a lo largo de la fibra óptica.

El objetivo de este proyecto de investigación es simular un OTDR mediante una interfaz gráfica para emular un enlace de comunicación óptica, con fibra monomodo o multimodo. De manera práctica se podrá comprobar cómo varía la atenuación y la potencia en función de la distancia para distintas longitudes de onda; así como, para diferentes conectores y empalmes. Para una eficiente comprensión del prototipo, es necesario revisar la argumentación contextual sobre fibra óptica y conocer la forma en la que funciona el OTDR. A la información descrita en párrafos anteriores y en base a la argumentación del simulador SOTDR; en forma general el contenido de este trabajo de investigación; se encuentra comprendida en cinco capítulos:

En el **CAPÍTULO I**, se argumenta el Problema, la delimitación, la justificación y los objetivos del estudio.

En el **CAPÍTULO II**, se diverge acerca del Marco Teórico, los antecedentes investigativos, la fundamentación teórica y la propuesta solución.

En el CAPÍTULO III, se expone la Metodología utilizada en el trabajo de investigación, su modalidad, los métodos y técnicas para la recolección de información, el procesamiento y análisis de datos; para en última instancia, concebir el desarrollo del proyecto.

En el CAPÍTULO IV, se presenta el Desarrollo de la Propuesta, las recomendaciones UIT-T, los requerimientos de SOTDR, el diseño de la GUI SOTDR la configuración de la interfaz, las pruebas y resultados y el presupuesto del proyecto de investigación.

Finalmente, en el CAPÍTULO V, se redactan las Conclusiones y Recomendaciones. Conclusiones obtenidas durante los capítulos previos y todo el desarrollo del proyecto de investigación. Las recomendaciones podrán evitar complicaciones en posteriores trabajos investigativos que puedan ramificarse del presente. Al término de la tesis se incorpora la Bibliografía y los respectivos Anexos.

CAPÍTULO I

EL PROBLEMA

1.1. TEMA

“INTERFAZ GRÁFICA PARA SIMULACIÓN DE UN OTDR APLICADO EN SISTEMAS DE COMUNICACIONES POR FIBRA ÓPTICA”.

1.2. PLANTEAMIENTO DEL PROBLEMA

Al tener redes de fibra óptica en crecimiento, a lo largo y ancho de la geografía ecuatoriana; se hace indispensable tener un equipo OTDR (Reflectómetro Óptico en el Dominio del tiempo), ya sea; si se trabaja en el campo de la fibra óptica o si se consume algún tipo de servicio a través de fibra óptica. Razones como esta hacen interesante el disponer de un OTDR a modo de aplicación de ordenador. Los enlaces de fibra óptica, así como las redes FTTx (Fiber to the X) con tecnología de Red Óptica Pasiva PON (Passive Optical Network), están a la vanguardia de las telecomunicaciones en el Ecuador; ya que a través de este medio de comunicación la velocidad de transmisión, el ancho de banda y el contenido de información pueden ser utilizados de mejor manera. [1], [2]

En el Ecuador no existe desarrollado un OTDR como aplicación para computador de escritorio, esta limitación tecnológica impone desventaja con los países desarrollados que aprovechan la técnica para implementar redes de fibra óptica eficientes y con factores eficaces de calidad. Una red mal implementada da origen a una pésima calidad

de servicio, coste de mantenimiento y reparación; además de la insatisfacción del cliente. La fiscalización en el Ecuador de una instalación de fibra óptica con los proveedores de servicios de internet, televisión y telefonía son de una exigencia técnica coherente al desarrollo de las comunicaciones ópticas en el país; estableciendo normativas QoS en la atenuación del enlace, pérdida máxima y promedio por empalme, ORL (pérdida óptica de retorno) y reflectancia; factores que actúan inversamente proporcional con el ancho de banda del servicio. [1]

La empresa SINFOTECNIA no cuenta con un simulador OTDR que permita por medio de un computador portátil o de escritorio obtener las trazas reflectométricas de un enlace de fibra óptica y comprobar el estado óptimo de la instalación. La empresa ha realizado el estudio para simulación de un OTDR mediante una interfaz gráfica y analizar los resultados obtenidos. De forma sencilla se podrá comprobar cómo varía la potencia óptica en función de la distancia para distintas longitudes de ondas, así como, para distintas fibras ópticas, empalmes y conectores.

1.3. DELIMITACIÓN

1.3.1 Delimitación de Contenido

Área académica:	Comunicaciones.
Línea de investigación:	Tecnologías de la Información y de la Comunicación.
Sublínea de investigación:	Comunicaciones Ópticas.

1.3.2 Delimitación Espacial

El proyecto de investigación se realizará en la empresa de redes y comunicaciones SINFOTECNIA S.A. de la ciudad de Ambato, provincia de Tungurahua.

1.3.3 Delimitación Temporal

La presente investigación se desarrolló en el periodo académico de Septiembre/2017 a Febrero/2018 de acuerdo a lo establecido en el numeral 7.2 y numeral 9.1 de los

Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

1.4. JUSTIFICACIÓN

El Reflectómetro óptico en el dominio del tiempo (OTDR) es un dispositivo muy utilizado en sistemas de comunicaciones por fibra óptica, para conocer de una manera rápida y sencilla como varía la potencia óptica a lo largo de una instalación de fibra óptica, la efectividad de este equipo es proporcional a su coste de fabricación; mientras mejores características técnicas posean, puede llegar a un coste muy elevado superando los miles de dólares. Para esta problemática, surge la propuesta de desarrollar dicha tecnología por medio de herramientas informáticas; que permitan crear un simulador en forma de interfaz gráfica para ser usado en un computador. Este aspecto justificará el aporte científico de la ingeniería; en virtud de sustentar con un prototipo, la utilización de un equipo OTDR profesional de excesivo precio.

El proyecto de investigación es viable debido a su importancia y el impacto positivo que probablemente imprima en la manera de probar y verificar la instalación de una red óptica, además; su desarrollo contará con la asesoría de ingenieros de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato; beneficiando de forma directa a la empresa de redes y comunicaciones SINFOTECNIA S.A., al investigador y los usuarios del simulador. Asimismo, se beneficiarán directamente los estudiantes egresados de la carrera de Ingeniería en Electrónica y Comunicaciones, que busquen profundizar el tema de investigación con respecto a simuladores de comunicaciones por fibra óptica. Consecuentemente los beneficiarios indirectos, serán los lectores de este trabajo investigativo; a más de de la comunidad técnica inmersa en el área de las comunicaciones ópticas.

La implementación de una interfaz gráfica para simulación de un OTDR, es institucionalmente factible, debido a que consolida el trabajo de campo realizado por la empresa SINFOTECNIA S.A.; consiguiendo la instalación de redes de fibra óptica más eficaces sin utilizar un equipo OTDR como tal, sino emplear una herramienta de ordenador que pueda demostrar el funcionamiento de las mismas, elevando el grado de satisfacción de sus clientes. A su vez, el desarrollo del proyecto es técnicamente factible, debido a que la empresa antes mencionada, posee los equipos tecnológicos

necesarios tanto en hardware y software, dispone de computadores de alta gama con procesadores Intel Core i7, además de licencias de software y la infraestructura idónea para implantar el simulador.

Económicamente el desarrollo del proyecto es factible porque no requiere de un presupuesto o financiamiento excesivo, al disponer de los recursos proporcionados por la empresa SINFOTECNIA S.A. y recurrir a la adquisición de software en versión estudiantil para evitar un gasto irrisorio en la compra de licencias de uso informático. La empresa SINFOTECNIA S.A. conoce de la importancia del proyecto y los beneficios que seguramente aportará su uso, a corto y mediano plazo recuperando el capital invertido. A su vez, la información inmersa en el desarrollo del proyecto, es enriquecida y permitirá fundamentar la investigación desde diversas fuentes de información existentes, tanto primarias y secundarias; conjugadas en libros de comunicaciones ópticas, papers, revistas científicas, blogs, foros. El trabajo investigativo se apoyará con suficiente información técnica y herramientas robustas en el campo de los diseños de ingeniería, gráficos, texto, algoritmos, etc.

1.5. OBJETIVOS

1.5.1 Objetivo General

- Implementar una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica.

1.5.2 Objetivos Específicos

- Analizar la tecnología existente con respecto a herramientas para simulación de un OTDR (Reflectómetro Óptico en el Dominio del Tiempo).
- Examinar algoritmos, prototipos y diagramas que enmarquen la ingeniería enfocada a la simulación de un OTDR.
- Integrar los procesos ópticos y algoritmos matemáticos computacionales en una interfaz compatible con las tecnologías de comunicaciones ópticas vigentes en el mercado.

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES INVESTIGATIVOS

Dentro de la bibliografía revisada no se ha encontrado documentación referente algún diseño e implementación de un simulador de un reflectómetro óptico en el dominio del tiempo (OTDR). Al mismo tiempo, en los repositorios digitales de las diferentes Universidades y Escuelas Politécnicas se localizó investigaciones pertinentes a simulaciones de comunicaciones por fibra óptica, mismas que servirán de soporte a la presente investigación; las cuales se detallan a continuación:

Wilson Leonardo Gómez Pauta de la Universidad del Azuay, en el 2015 realizó el “Desarrollo de aplicación de software para automatizar el diseño de enlaces ópticos y su aplicación en un enlace para la central Minas-San Francisco”. El proyecto fue constituido en dos partes, la implementación de un enlace de fibra óptica real y la comprobación de los resultados del mismo con una aplicación de software desarrollada en NETBEANS de entorno de desarrollo Java. El hardware empleado para el desarrollo del proyecto fue un ordenador Intel Core i3; además de los recursos para el tendido de fibra óptica, como: el cable de fibra óptica ADSS G.652D de 24 hilos vanos de 150 m, el transceptor Cisco SFP + 10G-BASE LR, una manga tipo domo de 24 hilos y un ODF de 24 puertos SC monomodo G.652D pulido APC. Los resultados dejan proveer que la aplicación de software, se encarga de automatizar el proceso de diseño de un enlace óptico punto a punto; para esto calcula el balance de potencia y

ancho de banda que presenta el sistema según los parámetros de ingreso, garantizando el correcto funcionamiento del diseño. La aplicación es capaz de graficar la ruta que sigue la fibra en un mapa real, y al posicionar el puntero en cualquier punto del enlace se visualiza la atenuación existente desde el inicio hasta esa ubicación. [2]

Ulisses Weyl Da Cunha Costa de la Universidade Federal Do Pará, en el 2013 realizó el “Análisis de la influencia de monitoreo OTDR en WDM-PON: Un estudio basado en simulaciones”. Se diseñó una aplicación que simula un enlace de fibra óptica de una red PON punto a punto, o punto a multipunto; la misma fue perfeccionada en dos herramientas informáticas como son: Optisystem y VPI Transmission marker; los dos programas se complementan estableciendo parámetros reales del comportamiento de una red de fibra óptica. Dicha aplicación permite al usuario ver las trazas reflectométricas de manera precisa presentando cada evento como: empalmes, fusiones, conectores, etc. El hardware empleado para el desarrollo de este proyecto fue un ordenador Intel Core i5. Los resultados cuantifican por medio de simulaciones el impacto de la técnica de reflectometría en el dominio del tiempo con interferencia en transmisiones típicas de una red WDM-PON, además; de la efectividad del prototipo en el monitoreo responsable y supervisión de tal tecnología, donde el OTDR es comúnmente aplicado. [3]

Luis Romero Cortés de la Universidad de Sevilla, en el 2012 realizó la “Reflectometría óptica en el dominio de la frecuencia (OFDR) para la caracterización de componentes y dispositivos ópticos”, Se elaboró un software analizador vectorial de dispositivos ópticos capaz de registrar el retardo del dispositivo ante diferentes fuentes de luz, el analizador fue construido en MATLAB como tecnología de software; de manera adicional se utilizó SIMULINK para la lectura de los dispositivos ópticos. La tecnología de hardware utilizada fue representada por un computador Intel Core i3. El diseño se ha realizado en base a técnicas de reflectometría e interferometría óptica ampliamente utilizadas en los dispositivos de instrumentación óptica comerciales. El simulador es netamente un analizador vectorial de componentes y dispositivos ópticos OVA, capaz de registrar medidas de módulo y fase de los coeficientes de reflexión y transmisión de dichos dispositivos, el método de medida empleado en su diseño es la reflectometría óptica en el dominio de la frecuencia OFDR. [4]

Bruno Tonello de la Università degli Studi di Padova, en el 2015 realizó el “Monitoreo basado en OTDR en Redes Ópticas Pasivas (PON)”. Se implementó un sistema experimental de monitoreo de redes ópticas pasivas usando un equipo OTDR como soporte para la verificación de resultados. Este sistema informático utiliza SIMULINK y MATLAB, por medio de una interfaz permite acceder a los datos proporcionados por el equipo OTDR y obtener resultados directrices a las técnicas TDM-PON (Time Division Multiplexing PON), WDM-PON (Wavelength Division Multiplexing PON) y TWDM-PON (Time and Wavelength Division Multiplexing PON). Usando una computadora Intel Core i7 como tecnología de hardware, además de un OTDR EXFO FTB100, el autor del proyecto obtiene como resultados que la implementación del sistema de monitoreo basado en OTDR para redes ópticas pasivas, actúa eficazmente con fibras monomodo y multimodo para los modelos de redes ópticas pasivas TDM, WDM y TWDM; señalando que es posible identificar los parámetros de un enlace y observar las trazas resultantes, además de mostrar la pérdida del enlace y su respectiva refracción. [5]

2.2. FUNDAMENTACIÓN TEÓRICA

2.2.1 Breve descripción de los sistemas de comunicación por fibra óptica

Los sistemas de comunicación por fibra óptica transmiten información modulada en el seno de un campo electromagnético que viaja en forma de onda a través de la fibra óptica. Dicho campo electromagnético oscila a frecuencias ópticas, las cuales son demasiado elevadas en comparación con las frecuencias de trabajo de los sistemas electrónicos convencionales de comunicaciones, tales como los sistemas de radiocomunicación o los sistemas guiados por líneas de transmisión y guías de onda, que operan en banda de microondas. [6]

La señal de comunicaciones por fibra óptica:

Es un campo electromagnético consistente en una onda de muy alta frecuencia procedente de un transmisor óptico que es modulada por una señal de corriente eléctrica que alimenta bien al transmisor, o bien a un modulador electro-óptico externo. En el primer caso se dice que el sistema de comunicaciones es IMDD (Intensity Modulation and Direct Detection), en el segundo, el sistema implementa detección

coherente y el proceso de demodulación involucra la detección individual de las componentes en fase y cuadratura de la señal recibida para la recuperación de su fase. [6]

Debido a sus características de atenuación y dispersión, la señal óptica se construye preferentemente con luz infrarroja, lo cual da lugar a sistemas que operan en torno a los cientos de terahercios (muy alta frecuencia). [6]

El medio de transmisión óptico:

La fibra óptica es una guíaonda dieléctrica cilíndrica formada por dos capas coaxiales de compuesto semiconductor (núcleo y revestimiento) con dopajes adecuados al perfil de índice de refracción deseado. En función de dicho perfil, se clasifica a las fibras ópticas en fibras salto de índice y fibras de índice gradual. La figura 2.1 muestra el símbolo que representa a la fibra óptica. [6]

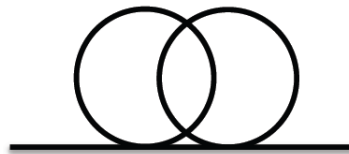


Figura 2.1. Símbolo de la fibra óptica. [6]

Una fibra óptica transporta un perfil transversal estacionario de potencia electromagnética sobre una onda longitudinal progresiva que viaja a lo largo de su eje de simetría. Según si la fibra acopla la potencia óptica incidente en uno o varios modos de propagación, se clasifica a las fibras ópticas en monomodo y multimodo respectivamente. El número de modos soportados por la fibra es función del radio de su núcleo y la frecuencia de la señal óptica. [6]

En una fibra óptica el índice de refracción efectivo determina el índice de refracción que experimenta un modo de propagación en relación a su velocidad de grupo. Si se trata de una cantidad compleja, la parte imaginaria indica la atenuación por unidad de longitud que sufre una onda electromagnética en el interior de la fibra. [6]

Para cada modo guiado en la fibra definimos una constante de propagación compleja dependiente de la frecuencia cuya parte real recibe el nombre de constante de atenuación α y cuya parte imaginaria recibe el nombre de constante de fase β . [6]

$$\gamma = -(\alpha + j\beta) = -\alpha - j \frac{2\pi}{c} f n \quad (1)$$

Con esto, una fibra óptica de longitud L puede modelarse en primera aproximación como un canal LTI de retardo y atenuación cuya función de transferencia compleja se muestra en la siguiente ecuación:

$$H_{FO}(f) = e^{\gamma L} = e^{-\alpha L} e^{-j\beta L} = e^{-\alpha L} e^{-j2\pi f \frac{L}{c} n} \quad (2)$$

Un modelo más complejo de la fibra óptica incluye los efectos de la dispersión, que introduce un ensanchamiento temporal de los pulsos que viajan por la fibra. Este modelo implica una aproximación no lineal de la constante de fase. [6]

Considerando que el ancho de banda efectivo de un sistema de comunicaciones es del orden de su frecuencia de explotación, la fibra óptica resulta un medio físico excelente a la hora de transmitir grandes cantidades de información a elevadas tasas binarias, ya que su ancho de banda es del orden de los terahercios. Existe un amplio catálogo de clases de fibras ópticas que cubren gran parte de la casuística existente en los sistemas de comunicaciones ópticas, algunos ejemplos son S-MMF (Standard MultiMode Fiber), S-SMF (Standard Single Mode Fiber), NZ-DSF (NonZero Dispersion-Shifted Fiber), NLF (Highly NonLinear Fiber), PCF (Photonic Crystal Fiber), etc. [6]

El transmisor óptico:

Idealmente, todo transmisor de comunicaciones consiste en un dispositivo generador de una señal portadora de gran pureza espectral y un modulador analógica o digital. En el caso de los sistemas de comunicaciones ópticas, tal dispositivo se construye en forma de sistema de niveles discretos y estado gaseoso o bien de diodo emisor de luz, un componente electrónico de estado sólido que realiza una conversión electro-óptica al emitir en forma de luz infrarroja una señal paso banda obtenida a partir de su corriente de alimentación. [7]

Este tipo de diodos se clasifican en LED (Light Emitting Diode) y láser (Light Amplification by Stimulated Emission of Radiation), donde los primeros se caracterizan por tener un espectro de emisión de banda ancha y los segundos se caracterizan por tener un espectro de emisión idealmente monocromático transportado

por un haz de luz coherente y colimado. La figura 2.2 muestra el símbolo que representa al láser. [7]

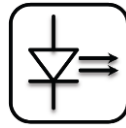


Figura 2.2. Símbolo del transmisor óptico. [7]

La eficiencia espectral de los sistemas de comunicaciones en fibra óptica es fuertemente dependiente del grado de monocromaticidad de su fuente de portada. [7]

En el contexto de la caracterización de sistemas ópticos resulta tan interesante la posibilidad de contar con fuentes ópticas altamente monocromáticas como la de contar con fuentes de espectro plano y muy amplio ancho de banda. Concretamente en este último caso, dada la complejidad que conlleva la implementación de delta de Dirac, la forma más precisa de conseguir una fuente de luz blanca (esto es, una fuente de espectro plano y amplio ancho de banda) coherente y con alta energía, es mediante el empleo de procesos no lineales que permiten la generación de un tipo de pulsos ultracortos denominados solitones ópticos, los cuales se vuelven inestables en medios no lineales y degeneran en pulsos aún más cortos, ensanchando así su espectro. La característica fundamental de los solitones es, a parte de su corta duración temporal, que compensan la dispersión, por lo que no sufren ensanchamiento temporal alguno en su propagación a través de medios ópticos confinados. Esto los convierte en excelentes aproximaciones físicas a la delta de Dirac. [8]

Esta técnica recibe el nombre de supercontinuo y su implementación es complicada y altamente inestable si no se realiza en el laboratorio bajo condiciones de trabajo controladas y muy específicas, además es necesario contar con equipamiento de alta precisión, que resulta muy costoso. Es por ello que, aunque este tipo de fuente es ideal para la caracterización de sistemas ópticos, es preferible y casi imprescindible evitar su uso en el diseño de instrumentos de medida. [8]

El receptor óptico:

El receptor de comunicaciones ópticas es un dispositivo fotodetector que realiza una conversión opto-electrónica de la señal óptica recibida. Consiste en un fotodiodo

polarizado en inversa cuya señal de salida es una corriente eléctrica proporcional a la potencia óptica recibida. [8]

$$I = \Re P_{opt} = \Re |E|^2 \quad (3)$$

Donde \Re es la responsividad del fotodetector, I es la corriente fotodetectada, P_{opt} la potencia óptica incidente y E el campo eléctrico incidente. La figura 2.3 muestra el símbolo que representa al fotodetector. [8]

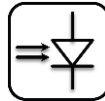


Figura 2.3. Símbolo del receptor óptico. [8]

La ecuación anterior desprende un corolario, del hecho de que la detección óptica se realice en función de magnitudes de potencia y no de campo electromagnético. [8]

2.2.2 Fundamentación de los sistemas de comunicaciones por fibra óptica

Los sistemas de comunicación óptica difieren en principio de los sistemas de microondas sólo en la gama de frecuencia de portadora utilizada para transportar la información. Las frecuencias de portadora en sistemas ópticos son de 200 THz, que difieren mucho de las frecuencias de portadora de sistemas de microondas de 1 GHz. [9]

Tomando, por ejemplo, el 1% como el valor límite, sistemas de comunicaciones ópticas tienen el potencial de llevar la información a velocidades de 1 Tbps. Este es el ancho de banda de enorme potencial de los sistemas de comunicación óptica que es la fuerza motriz detrás del desarrollo de todo el mundo y el despliegue de sistemas de ondas de luz. Los sistemas actuales operan a velocidades de 10 Gbps, lo que indica que hay un amplio margen para seguir realizando mejoras. [9]

En la figura 2.4 se ilustra el diagrama de bloques genérico de un sistema de comunicación óptica, que dispone de un transmisor (óptico), un canal de comunicación (fibra óptica), y un receptor (óptico), los tres elementos son comunes a todos los sistemas de comunicaciones. Las comunicaciones por fibra óptica se clasifican en guiadas y no guiadas. Celis B., I., sostiene que los sistemas ópticos guiados, emiten un

haz óptico (Tx) que es confinada mediante el uso de fibras ópticas, por este motivo se denomina sistema de comunicación por fibra óptica. [9]

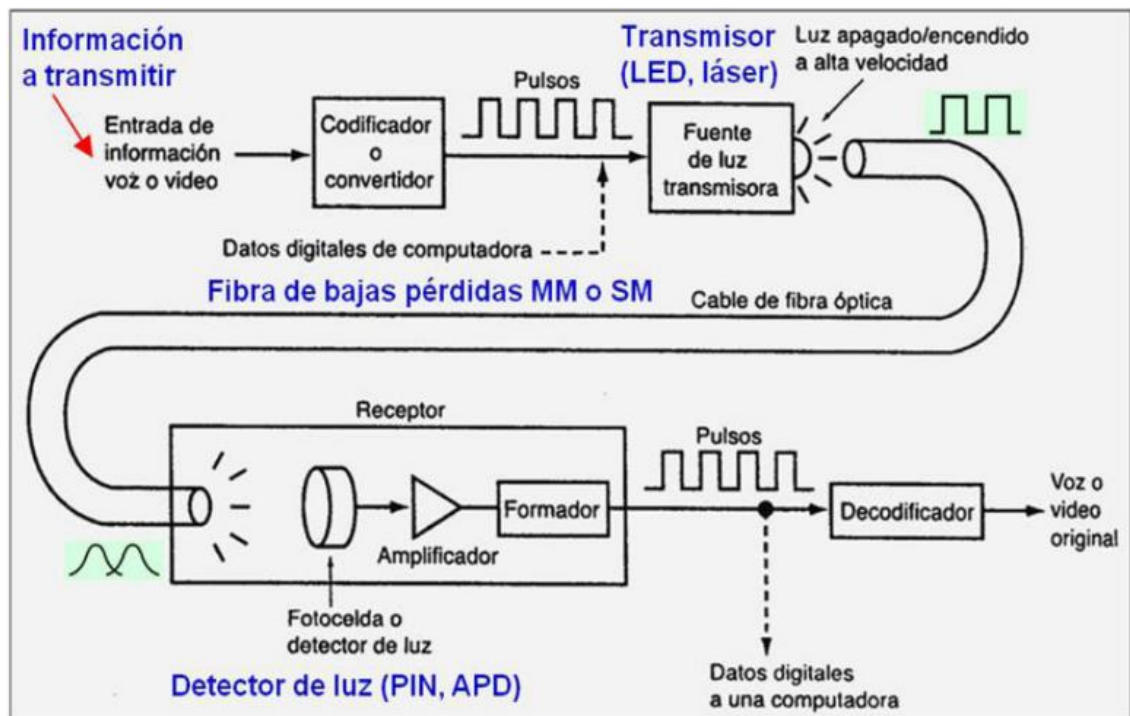


Figura 2.4. Diagrama de bloques de un sistema de comunicación óptico. [9]

Mientras que los sistemas de comunicación óptica no guiados, lo haces óptico emitidos por el transmisor son propagados de forma similar a las microondas. Estos son los menos apropiados para aplicaciones de radiodifusión que los sistemas de microondas, debido a que los haces ópticos se extienden principalmente en la dirección hacia adelante (como resultado de su longitud de onda corta). [9]

En el caso de la propagación terrestre, la señal en sistemas no guiados puede deteriorarse considerablemente por la dispersión dentro de la atmósfera. Este problema, por supuesto, desaparece en las comunicaciones en el espacio libre por encima de la atmósfera de la tierra (por ejemplo, las comunicaciones entre satélites). Aunque se necesitan los sistemas de comunicación óptica de espacio libre para ciertas aplicaciones, la mayoría de aplicaciones terrestres hacen uso de sistemas de comunicación de fibra óptica. [9]

Las aplicaciones de telecomunicaciones se pueden clasificar ampliamente en dos categorías, a largo y corto recorrido, que estarán siempre en función de las señales ópticas a transmitirse para distancias pequeñas o grandes, comparación con las interurbanas que son de 100 Km. Los sistemas de telecomunicaciones utilizados para distancias extensas, necesitan de líneas troncalizadas de alta capacidad. De hecho, la tecnología detrás de comunicación óptica es a menudo impulsada por las aplicaciones de larga distancia. [9]

Cada generación sucesiva de los sistemas de comunicaciones ópticos fue capaz de funcionar a velocidades más altas y en distancias más largas. La regeneración periódica de las señales ópticas mediante el uso de repetidores todavía es solicitada en la mayoría de los sistemas de larga distancia. Las distancias de separación para realizar transmisiones fueron de miles de kilómetros mediante el uso de amplificadores ópticos. [9]

Elementos de un sistema de comunicaciones ópticas:

Los elementos que conforman los sistemas de comunicaciones ópticas se muestran en el diagrama de bloques de la figura 2.5 en la cual el medio de transmisión son los cables de fibra óptica. Mientras que los otros componentes, es decir, el transmisor óptico, fuente óptica, detector óptico y receptor óptico, son diseñados para satisfacer las necesidades específicas de un canal de comunicación. [10]

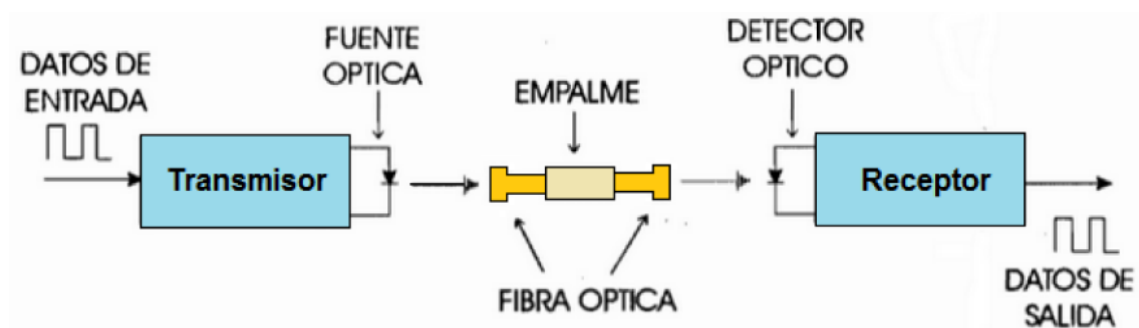


Figura 2.5. Elementos de un sistema de comunicación óptica. [10]

En este apartado se describe brevemente a la fibra óptica, más adelante se explicará en detalle lo relacionado a la fibra óptica. Es decir, este apartado discute las cuestiones generales relacionadas con el papel de la fibra óptica como medio de comunicación. [10]

La fibra óptica como medio de comunicación

El papel del medio de comunicación es transportar la señal óptica del transmisor al receptor sin distorsionarla. La mayoría de los sistemas ópticos emplean a la fibra óptica como medio de comunicación, debido a que la fibra de silicio emite haces de luz con bajas pérdidas $\leq 0.2 \text{ dB/Km}$. Con este tipo de fibra, también se redujo la potencia óptica en 1% después de 100 Km. [10]

Por eso, las pérdidas en las fibras ópticas todavía son un problema a la hora de diseñar un sistema óptica de larga distancia, así como también el escoger correctamente las repetidoras o amplificadores. Otro problema que se presenta en el diseño es la dispersión de la fibra, lo que conduce a la ampliación de los impulsos ópticos individuales con propagación. [10]

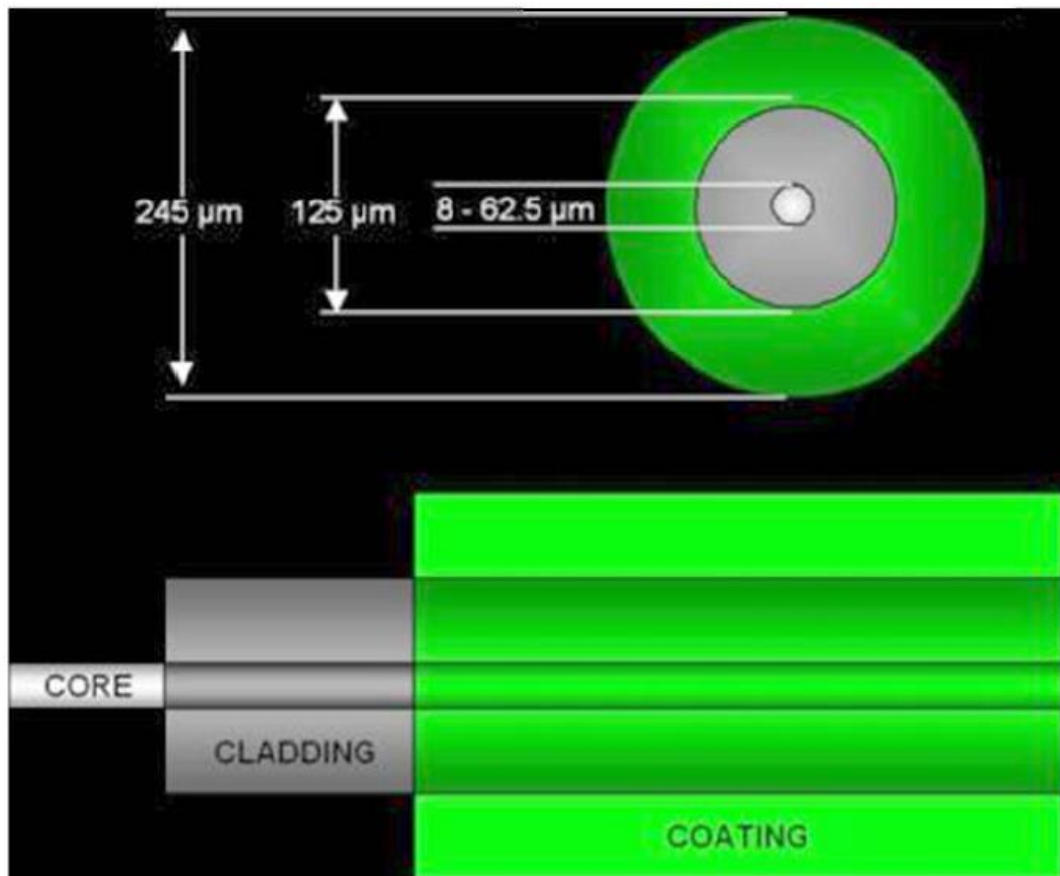


Figura 2.6. Estructura de una fibra óptica. [10]

Celis B., sostiene que los pulsos ópticos eran propagados fuera de la ranura de bit que le asignaban, es por esto que la señal transmitida se degradará severamente. Finalmente, se hace imposible recuperar la señal original con una alta precisión. [10]

Este inconveniente resulta ser más grave en las fibras multimodo (MM), debido a que los impulsos se propagan rápidamente (típicamente a una velocidad de 10 ns/Km) a diferentes velocidades asociadas con diferentes modos de fibra, tal como se ilustra en la figura 2.7. [10]



Figura 2.7. Fibra óptica multimodal (MM). [10]

Es por este motivo que la mayoría de sistemas ópticos emplean las fibras monomodo (SM) para transmisiones a largas distancias debido a que se propaga en una sola dirección, sin descartar la dispersión que conduce a la ampliación de pulso (típicamente $<0,1$ ns/Km), pero es muy pequeño como para ser aceptable y se reducen aún más a través del control de la anchura espectral de la fuente óptica, figura 2.8. [10]



Figura 2.8. Fibra óptica monomodo (SM). [10]

Transmisores ópticos

El papel de un transmisor óptico es la de convertir la señal eléctrica en forma óptica y para lanzar la señal óptica resultante en la fibra óptica. La figura 2.9 se muestra el diagrama de bloques de un transmisor óptico, el cual se compone de una fuente óptica, un modulador y un acoplador de canal. [12]

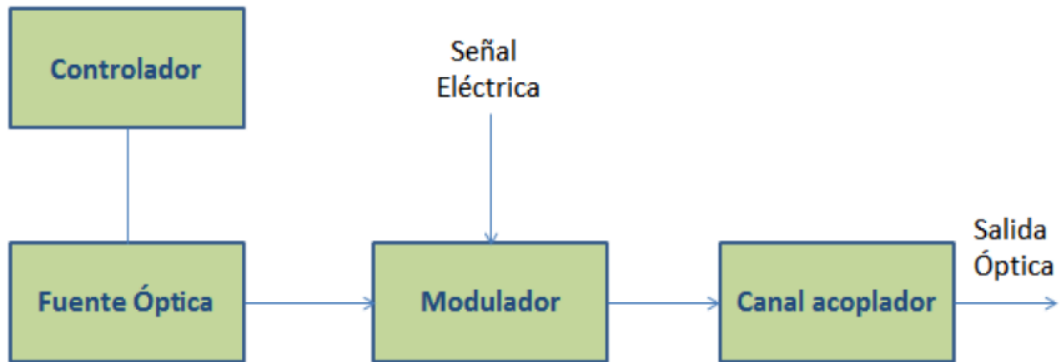


Figura 2.9. Diagrama de bloque de un transmisor óptico. [12]

Mientras que en las figuras 2.10 y 2.11 se muestran diagramas de bloques como ejemplos de transmisores ópticos con modulación interna y externa respectivamente.

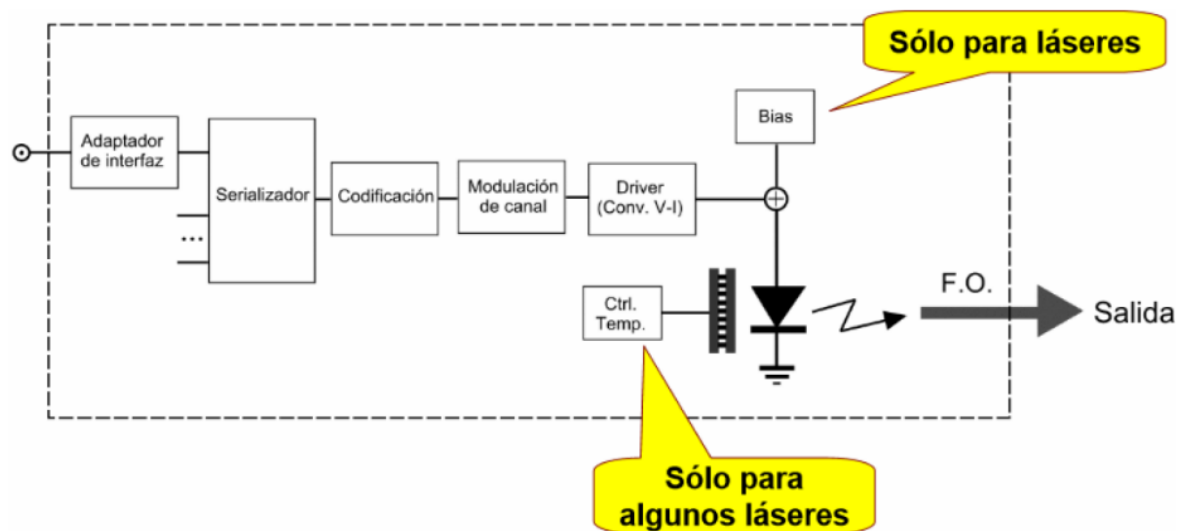


Figura 2.10. Diagrama de un transmisor óptico de modulación interna. [12]

Los láseres semiconductores o diodos emisores de luz se usan como fuentes ópticas, debido a su compatibilidad con el canal de comunicación de fibra óptica. La señal óptica se genera mediante la modulación de la onda portadora óptica. Un diodo emisor

de luz (LED) es un dispositivo semiconductor que emite un espectro estrecho de luz. Este efecto es una forma de electroluminiscencia. [12]

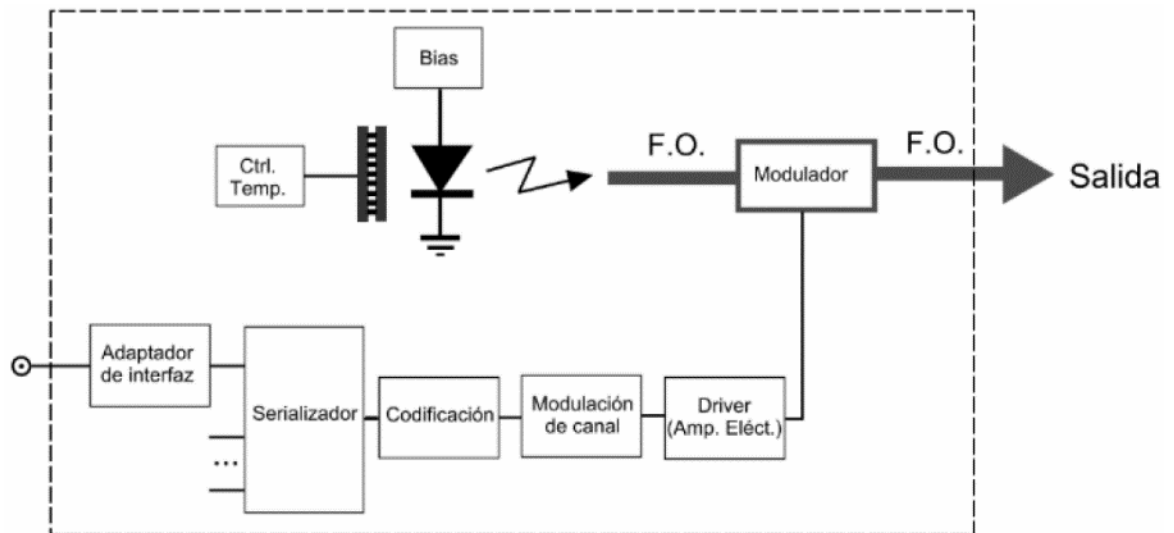


Figura 2.11. Diagrama de un transmisor óptico de modulación externa. [12]

En general, los LEDs son menos potentes que los láseres, pero son mucho menos caros. En la figura 2.12 se muestra un tipo de LED empleado en comunicaciones ópticas. [12]



Figura 2.12. Diodo emisor de luz de tipo lateral. [12]

Un láser semiconductor, es una fuente óptica que emite fotones en un haz coherente, es decir, que permite la amplificación de luz por emisión estimulada de radiación. La luz del láser consiste en una sola longitud de onda emitida en un haz estrecho. En la figura 2.13 se muestra un tipo de diodo láser (LD) o láser semiconductor. [12]

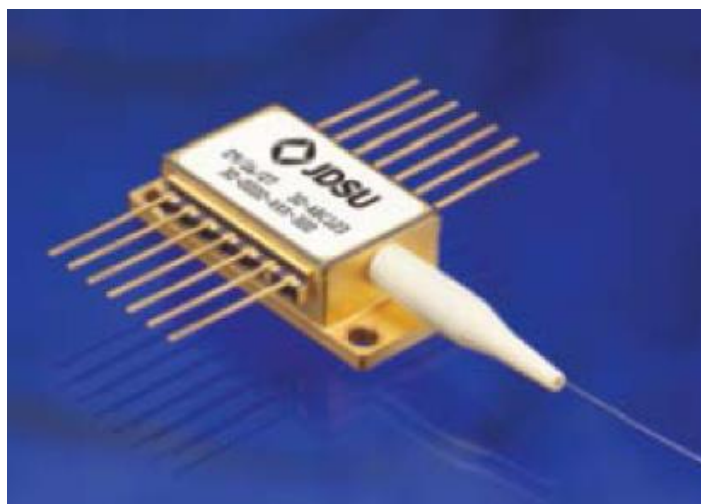


Figura 2.13. Diodo láser. [12]

Los láseres Fabry-Perot (FP) emiten luz en una serie de longitudes de onda discreta, la entrega de una anchura espectral entre 5 y 8 nm. La figura 2.14 muestra el diagrama esquemático de del láser FP. [15]

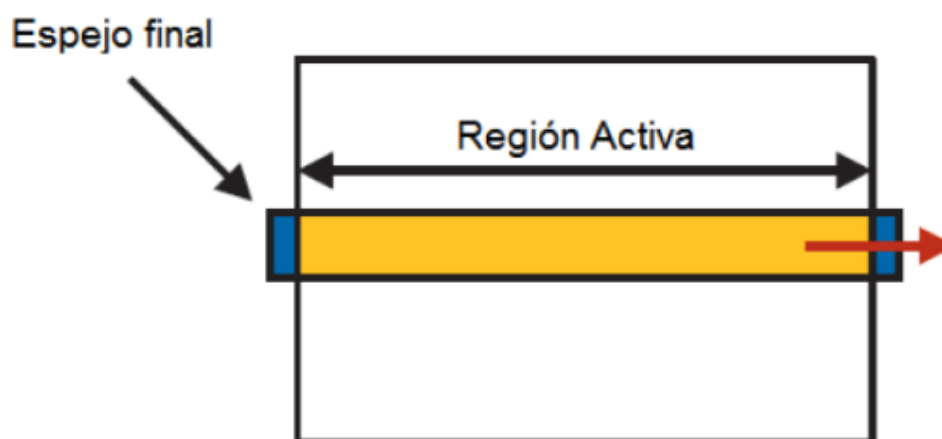


Figura 2.14. Diagrama esquemático del láser FP. [15]

Otro tipo de láser es el de retroalimentación distribuida (DFB), que es mucho más precisa que un simple láser FP, pero su capacidad de suministro de potencia de salida es mucho menor. Los láseres FP emiten una gran cantidad de armónicos en un rango de longitud de onda de 5 y 8 nm. Mientras que los láseres DFB, por otro lado seleccionan sólo una longitud de onda principal en el espectro del láser FP, proporcionando una anchura espectral estrecha de $<0,1$ nm. En la figura 2.15 se muestra el diagrama esquemático del láser DFB. [15]

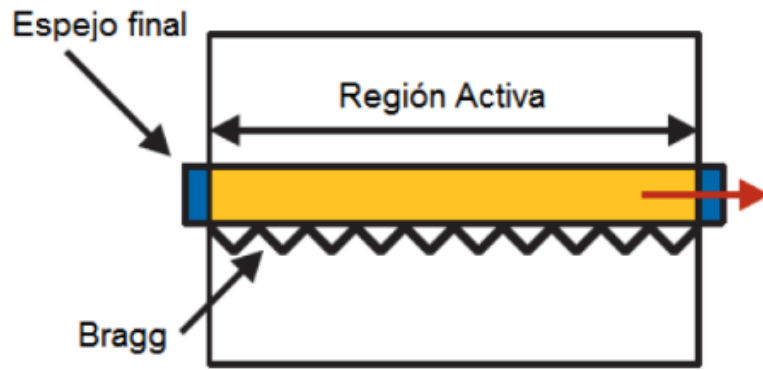


Figura 2.15. Diagrama esquemático del láser DBF. [15]

Es necesario conocer que la potencia inyectada a menudo se expresa en unidades de <<dBm>> con 1 mW como nivel de referencia, cuya fórmula es:

$$P(dBm) = 10 \log \left(\frac{P}{1mW} \right) \quad (4)$$

Los LEDs están limitados sus capacidades de modulación, es por eso que la mayoría de los sistemas ópticos emplean los láseres semiconductores como fuentes ópticas. [15]

Receptores ópticos

Los receptores ópticos convierten las señales ópticas recibidas en señales eléctricas en el extremo de salida del receptor óptico. En la figura 2.16 se observa el diagrama de bloques del receptor óptico compuesto por un acoplador, un fotodetector y un demodulador. El canal acoplador capta la señal óptica y la envía al fotodetector, que a su vez se utilizan como fotodetectores debido a su compatibilidad con todo el sistema. Mientras que el demodulador se encarga en presentar en la salida una señal eléctrica. [12]

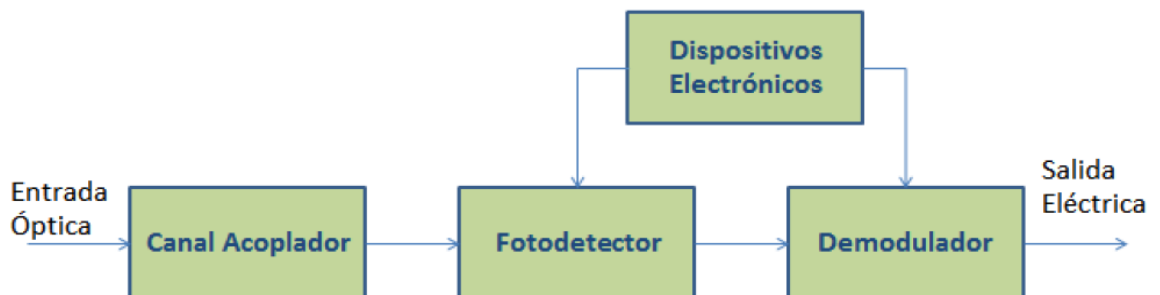


Figura 2.16. Diagrama esquemático del receptor óptico. [12]

Dentro del proceso de la demodulación en los receptores ópticos va a depender del tipo de modulación a escoger FSK o PSK, que adicionalmente requieren de sistemas heterodinos u homodinos. [11]

El rendimiento de un sistema óptico se caracteriza a través de la tasa de error de bits (BER). Aunque el BER se puede definir como el número de errores cometidos por segundo, tal definición hace que la tasa de bits BER sea dependiente. Es habitual definir al BER como la probabilidad media de identificación de bits correctos. Un $BER = 10^{-6}$ es la probabilidad de un error por cada millón de bits. La mayoría de los sistemas ópticos especifican un $BER = 10^{-9}$ como el requisito de funcionamiento; algunos incluso requieren de un $BER = 10^{-14}$. [11]

Un parámetro importante para cualquier receptor es la sensibilidad del receptor. Por lo general se define como la potencia óptica promedio mínimo requerido para realizar un $BER = 10^{-9}$. Los receptores ópticos son muy sensibles ante la presencia de la relación señal/ruido (SNR), sin dejar de incluir otros factores que generen ruido, corrompiendo así la señal recibida. [11]

Aunque en el mercado hay receptores perfectos, algo de ruido se introducen debido al proceso de fotodetección, conocido como ruido cuántico o el ruido de disparo, ya que tiene su origen en la naturaleza de la partícula de los electrones. Los receptores ópticos que operan en el límite de ruido de disparo se denominan receptores cuánticos limitados por el ruido. En la práctica no existen receptores que operen en el límite cuántico debido a la presencia de varias otras fuentes de ruido. [13]

Las fibras ópticas

La fibra óptica actualmente es el elemento básico en infraestructuras de sistemas de telecomunicaciones, debido a la capacidad del gran ancho de banda y las características de baja atenuación, lo que hace que sean ideales para la transmisión con altas velocidades (gigabit y más allá). En la figura 2.17 se muestran los tres tipos básicos de cable de fibra óptica que se utilizan en sistemas de comunicaciones ópticas. [13]

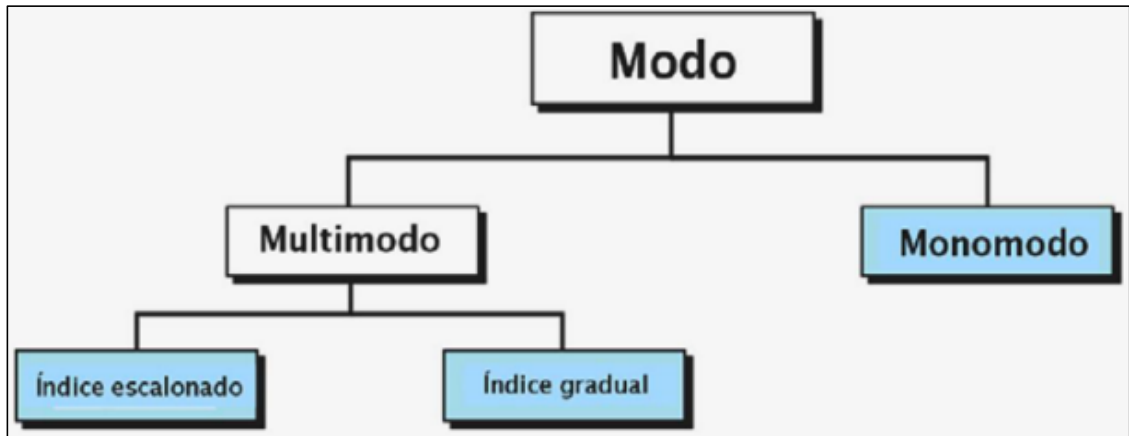


Figura 2.17. Clasificación de la fibra óptica. [13]

Fibra óptica multimodo de índice escalonado: Las fibras multimodo (MM) de índice escalonado (SI) fue el primer diseño de la fibra, pero es demasiado lento para la mayoría de usos, debido a la dispersión causada por las diferentes longitudes de trayectoria de los distintos modos. Este tipo de fibra apenas se utiliza en aplicaciones de telecomunicaciones y comunicación de datos actuales. [13]

Las fibras MM de índice escalonado, permite guiar rayos de luz a través de la reflexión total en el límite entre el núcleo y el revestimiento. El índice de refracción es uniforme en el núcleo, cuyo diámetro mínimo es 50 o 62.5 μm , un diámetro de revestimiento entre 100 y 140 μm , y una apertura numérica entre 0,2 y 0,5. La figura 2.18 muestra la forma de propagación de la señal de luz en una fibra multimodo de índice escalonado. [13]

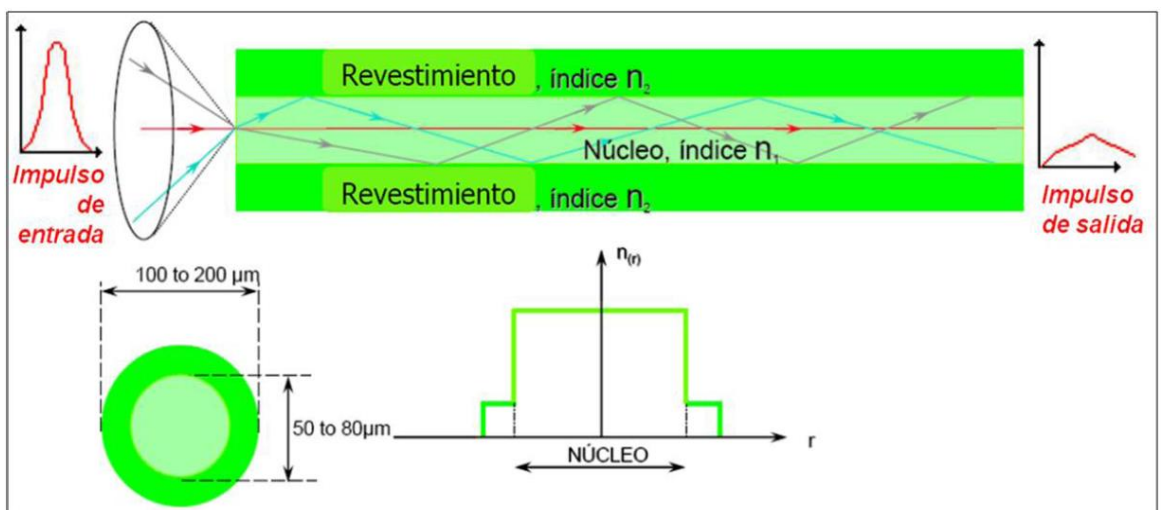


Figura 2.18. Fibra óptica multimodo de índice escalonado. [13],

Fibra óptica multimodo de índice gradual: La fibra multimodo de índice gradual (GI), es utilizada por las variaciones en la composición del vidrio en el núcleo para compensar las diferentes longitudes de trayectoria de los modos. Ofrece cientos de veces más ancho de banda que la fibra de índice escalonado. [13]

El núcleo de índice graduado (GI) de fibra multimodo posee un índice de refracción no uniforme, que disminuye gradualmente desde el eje central para el revestimiento. Esta variación del índice del núcleo obliga a los rayos de luz para la propagación a través de la fibra en forma sinusoidal. [13]

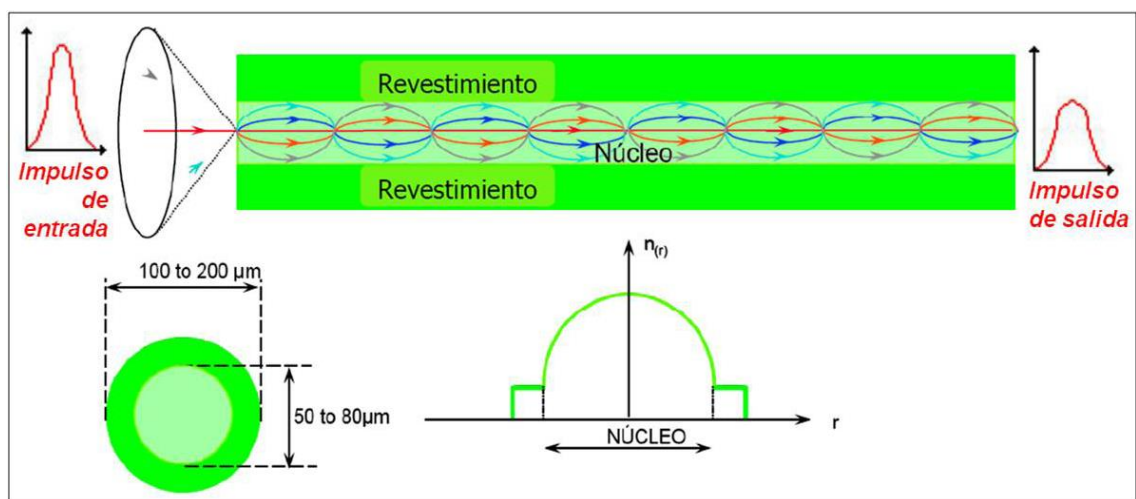


Figura 2.19. Fibra óptica multimodo de índice gradual. [13]

Fibra óptica monomodo (SM): Las fibras ópticas monomodo (SM) cuenta con atenuación inferior a las fibras multimodo (MM) y retiene mejor cada pulso de luz, ya que no muestra la dispersión causada por múltiples modos. Por lo tanto, la información se puede transmitir en distancias más largas. Al igual que la fibra multimodo, la fibra monomodo era generalmente caracterizada como la fibra de índice escalonado, que significaba que el índice de refracción del núcleo de la fibra es un paso por encima de la del revestimiento y no se graduó como lo es en la fibra de índice gradual. [13]

Las fibras monomodo modernas han evolucionado hacia diseños más complejos, como revestido emparejado, revestimiento reprimido y otras estructuras. [13]

Otra de las ventajas de utilizar la fibra monomodo, es su mayor rendimiento con respecto al ancho de banda. El diámetro del núcleo reducido de fibra monomodo limita la luz a un solo modo de propagación, lo que elimina la dispersión modal por completo.

Con los componentes adecuados de dispersión de compensación, una fibra monomodo puede transportar señales de 10 y 40 Gbps o superior a largas distancias. [13]

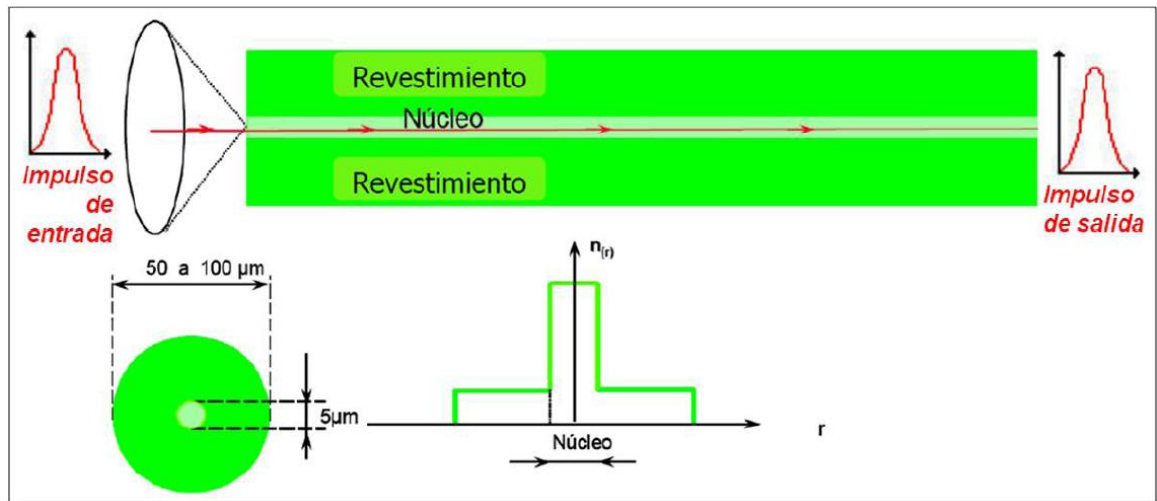


Figura 2.20. Fibra óptica monomodo. [13]

El sistema de la capacidad de carga se puede aumentar mediante la inyección de múltiples señales de diferentes longitudes de onda, también conocido como Multiplexación por División de Longitud de Onda (Wavelength Division Multiplexing). [13]

Empalmes

El diámetro del núcleo de la fibra óptica es muy pequeño, tanto que una fibra óptica con recubrimiento añadido tiene un tamaño similar a un cabello humano. Por ello cualquier irregularidad, por el motivo que sea, como un empalme puede suponer una pérdida significativa de potencia. Para conseguir la máxima transferencia de potencia de una fibra a otra, deben estar minuciosamente colocadas y estar lo más cerca posible para que entre los extremos de las dos fibras haya mínima cantidad de aire. [14]

La forma en la que viaja la luz por una fibra óptica no es igual a la que viaja la electricidad por un cable, la luz es una onda guiada por el interior de la fibra por lo que cualquier imperfección conlleva grandes pérdidas. [14]

A pesar de tantos inconvenientes hay varias formas de empalmar dos fibras. Estas son: Empalme por fusión, uso de materiales de índice refracción epoxi-coincidentes y con conectores mecánicos de diferentes tipos. [14]

Lo más importante en los tres métodos es que las fibras queden enfrentadas perfectamente. Sin embargo, se es capaz de alinear las fibras y no de alinear los núcleos, que es verdaderamente lo importante ya que es por donde viaja la luz fundamentalmente. El núcleo no es siempre el centro de la fibra. Los fabricantes tratan de situar el núcleo en el centro pero siempre hay un error denominado concetricidad, habiendo siempre pérdidas debidas a este fenómeno. [14]

Empalmes por fusión: La unión de dos fibras es de alguna manera una soldadura. Los extremos de las fibras a unir se cortan, se limpian de posibles impurezas y por último se unen y funden por el calor; dicho sea de paso, para fibras de sílice se necesitan altas temperaturas para conseguir la fusión. La atenuación media por empalme es de 0.1 dB, que es un buen valor, sin embargo, se debe dar cuenta de que 0.1 dB en una fibra de tan solo 0.5 Km es demasiado. [15], [16]

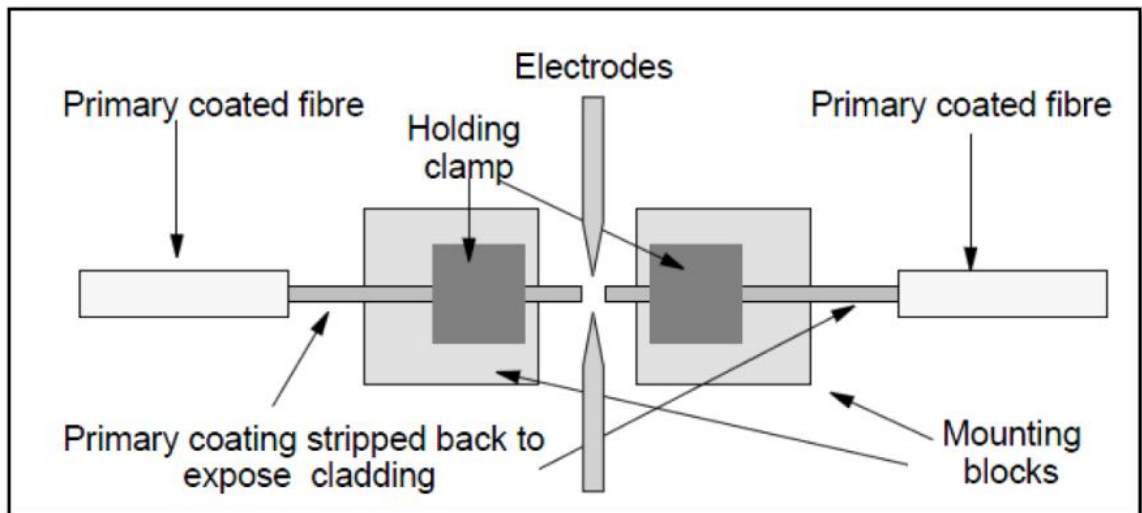


Figura 2.21. Procedimiento de empalme en fibra óptica. [15], [16]

En la figura 2.21 se observa la forma de unión de dos fibras por fusión, cuyo método trabaja de la siguiente manera:

- 1.- En primer lugar, se corta parte del revestimiento de la fibra y se pela la misma hasta que se consiga que la fibra en sí “esté al aire”. [16], [17]
- 2.- A continuación, los extremos de las fibras se colocan el uno del otro a pocos mm, sujetando ambos para que no se muevan durante la soldadura. [17]
- 3.- Las fibras se alinean una con la otra y se aproximan entre sí. [17]

4.- Cuando la alineación es correcta, entonces los dos electrodos entran en funcionamiento y se procede a la fusión de las dos fibras. [17]

Uso de materiales de índice de refracción epoxi-coincidentes: En esta técnica de unión de fibras, los extremos de las mismas son cortados y pulidos y alineados correctamente igual que en el caso anterior, pero su unión se realiza gracias a un epoxi-resina. En este caso, la unión de las fibras se denomina “mecánica” ya que no alinea las fibras teniendo en cuenta el lugar en el que se encuentra el núcleo. Hay muchas maneras de alinear la parte exterior de dos fibras ópticas:

- Un método común es usar un tubo de vidrio en el que cada fibra se introduce por un extremo. Una pequeña cantidad del epoxi-resina se coloca en el extremo de una de las fibras antes de introducirla al tubo. Suele haber un pequeño agujero en el centro del tubo, de manera que un exceso de epoxi-resina pueda escapar al exterior. [18]
- Hay muchos otros métodos de conseguir una correcta alineación de las fibras ópticas, entre ellas ranuras y varillas de alineación. [18]

No es necesario decir que esta técnica de empalme de fibras es la más barata, pero la calidad no es tan elevada. La calidad de la unión depende de:

- La concentricidad de la fibra.
- El “acabado” de la parte exterior de la fibra.
- La circularidad de la parte exterior de la fibra.
- La precisión del método de unión.

A pesar de la facilidad de la unión, hay un posible pequeño inconveniente (se dice posible ya que no es seguro que se produzca). Se trata de la estabilidad del epoxi-resina a lo largo del tiempo y su pérdida de cualidades pudiendo generar dispersiones en el tiempo si esta se rompiera. Por ello, actualmente se está desarrollando un nuevo tipo de epoxi-resina que mejore este posible inconveniente. [18]

Pérdidas por empalmes

Las pérdidas por empalmes se clasifican fundamentalmente en dos grupos:

1.- Pérdidas extrínsecas, son aquellas que se refieren solo a las pérdidas del empalme y no al tipo de fibra o fabricación. [18]

2.- Pérdidas intrínsecas, son aquellas inherentes al tipo de fibra o fabricación. [18]

Pérdidas extrínsecas: En la figura 2.22 se pueden apreciar los principales problemas de un mal empalme. [18]

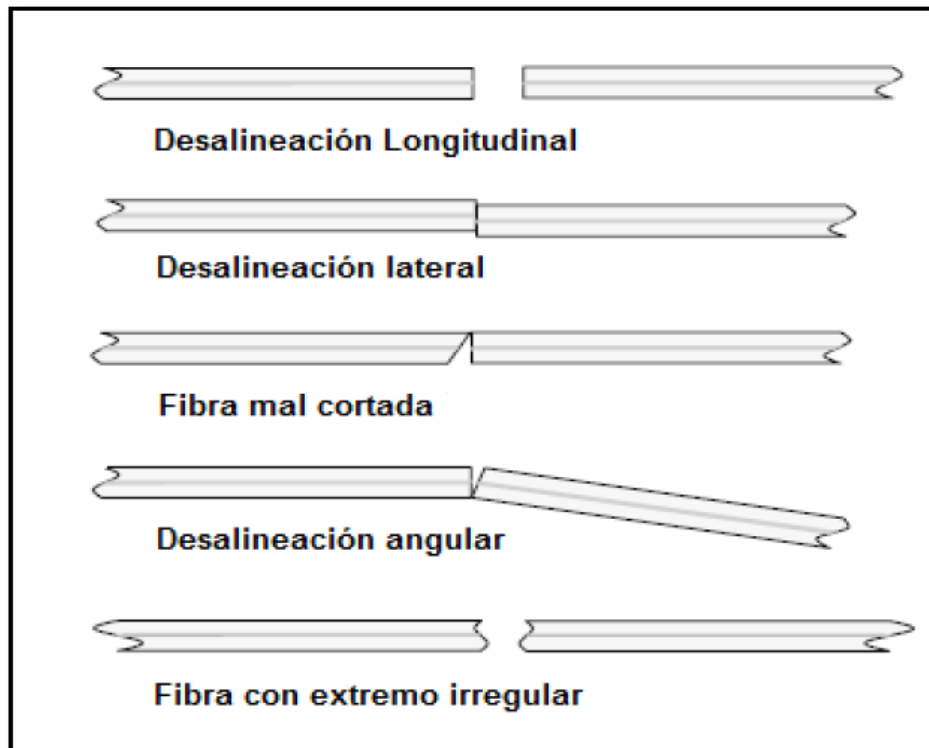


Figura 2.22. Posibles errores al empalmar fibras. [18]

Las pérdidas más comunes producidas por las irregularidades ilustradas en la imagen anterior a la hora de realizar un empalme, son las siguientes:

1.- Desalineación longitudinal: Tiene dos inconvenientes:

- Posibilidad de pérdida de luz hacia el exterior.
- “Creación” de un interferómetro de Fabry-Perot entre los dos extremos pudiendo reflejarse desde el 0 al 100% de la potencia. [18]

2.- Desalineación lateral: Es uno de los grandes problemas a la hora de realizar un empalme. Un desplazamiento de únicamente una micra provoca pérdidas de 0.2 dB (en fibra monomodo). [18]

3.- Fibra mal cortada: Se debe a un corte irregular y por ello la imposibilidad de unir las fibras ópticas correctamente. [18]

4.- Desalineación angular: Este inconveniente es mucho más perjudicial en fibras monomodo, puesto que su apertura numérica (AN) es mucho más pequeña. Una desalineación de 28 se traduce en casi 1 dB de atenuación. [18]

5.- Fibra con extremo irregular: Las pérdidas de esta irregularidad son difíciles de predecir. El principal problema es parecido al de la fibra mal cortada, es decir; la imposibilidad para unir correctamente las dos fibras. [18]

Pérdidas intrínsecas: Las pérdidas intrínsecas son aquellas que dependen del tipo de fibra óptica y de la manera en que la misma está fabricada. En la figura 2.23 se pueden apreciar algunas de las pérdidas intrínsecas en fibras ópticas. [18]

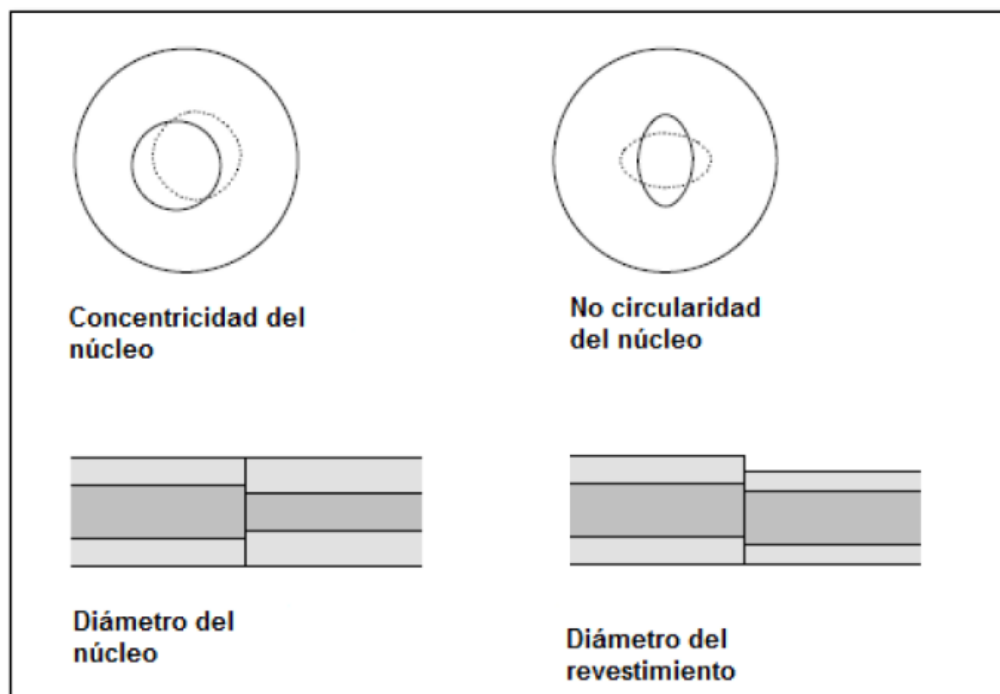


Figura 2.23. Esquema de algunas de las pérdidas intrínsecas en fibras ópticas. [18]

En la imagen anterior se puede apreciar algunos ejemplos. A continuación, se explican con más detalle:

1.- Errores de concentricidad en el núcleo: Como se ha explicado anteriormente, esta es una de las causas de mayores pérdidas a la hora de realizar empalmes. Se debe a que el núcleo de la fibra no está exactamente en el centro de la misma, por lo que al

realizar los empalmes los núcleos de las fibras no quedan perfectamente alineados. [18]

2.- No circularidad del núcleo: No importa la precisión con la que se realice el núcleo de la fibra, siempre tendrá una cierta desviación y no será perfectamente circular. Cuando se corta una fibra, para después empalmada el núcleo puede sufrir alteraciones. Esto no es muy importante en fibras multimodo, sin embargo, en fibras monomodo se produce el fenómeno de birrefringencia. [18]

3.- Diámetro del núcleo: Los diámetros de dos fibras que se van a empalmar nunca son exactamente iguales, por lo que a pesar de que estén perfectamente alineados siempre se escapa parte de la potencia. Un dato curioso es que en fibras perfectamente alineadas, pasa esto si la siguiente fibra tiene un núcleo más pequeño, ya que si este fuera más grande no sufriría pérdida alguna de potencia. [18]

4.- Diámetro del revestimiento: No se sabe nunca con certeza donde está el núcleo exactamente, por lo que las fibras deben alinearse exteriormente. Por lo que si el revestimiento de las dos fibras que componen el empalme es diferente los núcleos no podrán estar alineados. [18]

Además de estos problemas, se puede destacar lo que ocurre en fibras multimodo cuando se unen dos fibras de distinta apertura numérica. Dado que su cono de aceptación es diferente es posible que algunos modos no atraviesen el empalme. [18]

En resumen, los empalmes permiten la unión permanente de dos fibras ópticas, estas uniones producen una atenuación de valores pequeños, pero que, acumulados en varios kilómetros de trayecto puede ser una limitación importante en la longitud del enlace. [18]

Conectores

En muchas situaciones es necesario poder cambiar las conexiones entre fibras rápida y fácilmente, con este propósito se utilizan los conectores. Los conectores sujetan a la fibra en la posición correcta y mediante una suave presión se consigue una buena conexión. Por supuesto esto depende de la precisión con la que se haya fabricado el conector. Debido al pequeño tamaño de la fibra óptica, estos conectores necesitan en

su fabricación un extra de precisión ya que si no las pérdidas serían muy elevadas. No hace falta decir que la fabricación de estos conectores es muy complicada y costosa. Algunos de estos conectores vienen ya unidos a la fibra óptica, consiguiéndose de esta manera una mayor precisión que los conectores que se pueden montar y desmontar. Antes de utilizar un conector se debe asegurar de que está libre de suciedad, ya que la más mínima mota de polvo puede obstruir la fibra (núcleo de la fibra monomodo $\approx 8\mu m$). Los conectores suelen usarse sobre todo en las centrales de fibra óptica en las que frecuentemente es necesario realizar cambios rápidamente. [19]

En la actualidad, los conectores poseen una alta precisión lográndose a las 0.2 dB de pérdidas, sin embargo, la mayoría de los fabricantes todavía sugieren en sus hojas de características una pérdida de alrededor 1 dB por conector. [19] Un buen conector debe cumplir las siguientes características:

- Pérdidas de inserción bajas.
- Pérdidas de retorno bajas.
- Facilidad de instalación.
- Fiabilidad.

A continuación, se puede observar en la tabla 2.1 distintos tipos de conectores y en sus características típicas. En la primera columna está el nombre del conector, en la segunda aparecen las pérdidas del conector para fibras monomodo, en la tercera de igual manera que para la segunda columna solo que para fibras multimodo y en la última columna se indican las pérdidas de retorno de conector. Las pérdidas de retorno son la diferencia entre la potencia que se transmite y la potencia que se refleja, en este caso al pasar por el conector:

Tabla 2.1. Conectores comunes, considerando sus pérdidas de inserción y de retorno. [19]

Tipo de conector	Pérdidas de inserción (Fibra monomodo)	Pérdidas de inserción (Fibra multimodo)	Pérdidas de retorno
ST	0,25 dB	0,2 dB	40 dB
SC	0,25 dB	0,2 dB	40 dB
SMA	1,5 dB	-----	40 dB
FSD	0,5 dB	-----	40 dB
FC	0,25 dB	0,2 dB	40 dB
D4	0,25 dB	0,2 dB	35 dB
DIN	0,25 dB	0,2 dB	40 dB

Fuente. La Investigación

En la figura 2.24 se muestra la forma de varios conectores de fibra óptica empleado en sistemas de comunicaciones ópticas. [19]

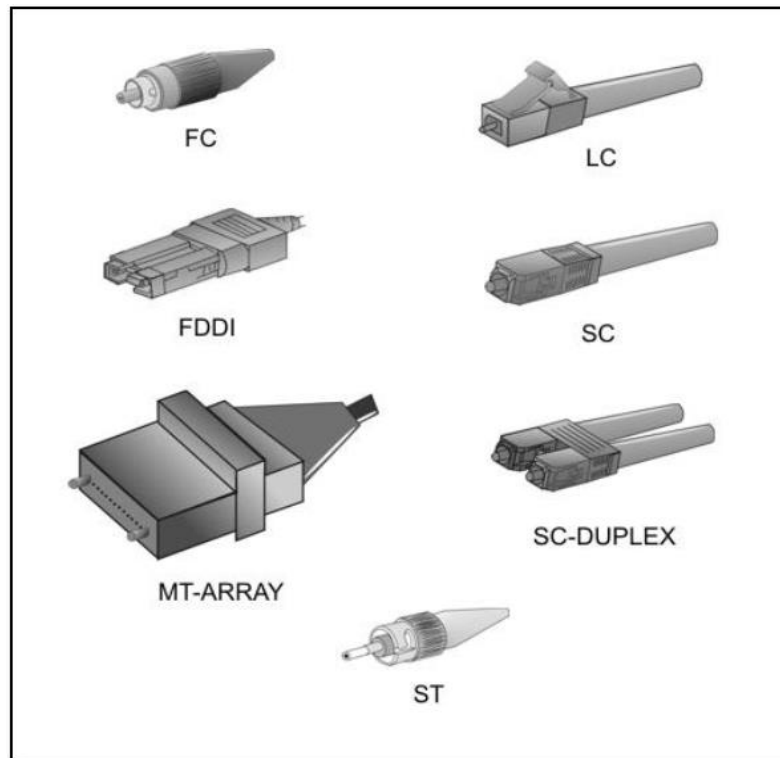


Figura 2.24. Forma de varios conectores de fibra óptica. [19]

En resumen, en un empalme, las pérdidas son menores que 0.01dB. En un conector son de 0.2dB. La razón es que en un empalme los extremos de las fibras se tocan. En un conector, se deja una brecha de aire para que las superficies pulidas de la fibra no se dañen durante el proceso de conectar o desconectar, aunque existen conectores que permiten el contacto físico de los núcleos. [19]

2.2.3 Parámetros fundamentales de un enlace de fibra óptica

Un enlace de fibra óptica está definido por varios parámetros: estructurales, de transmisión y de costos. Los parámetros estructurales son los inherentes a la fibra como: perfil de índice de refracción, apertura numérica, diámetro del núcleo, longitud de onda de corte. [20]

Longitud de Onda

La longitud de onda es la distancia que ocupa un ciclo completo de onda. En frecuencias luminosas la longitud de onda se la puede expresar en términos de micras ($1 \mu\text{m} = 10^{-6} \text{ m}$), nanómetros ($1 \text{ nm} = 10^{-9} \text{ m} = 0.001 \mu\text{m}$), aunque en el espectro visible es común ver medidas en Angstrom ($1 \text{ \AA} = 10^{-10} \text{ m} = 0.0001 \mu\text{m}$). [20]

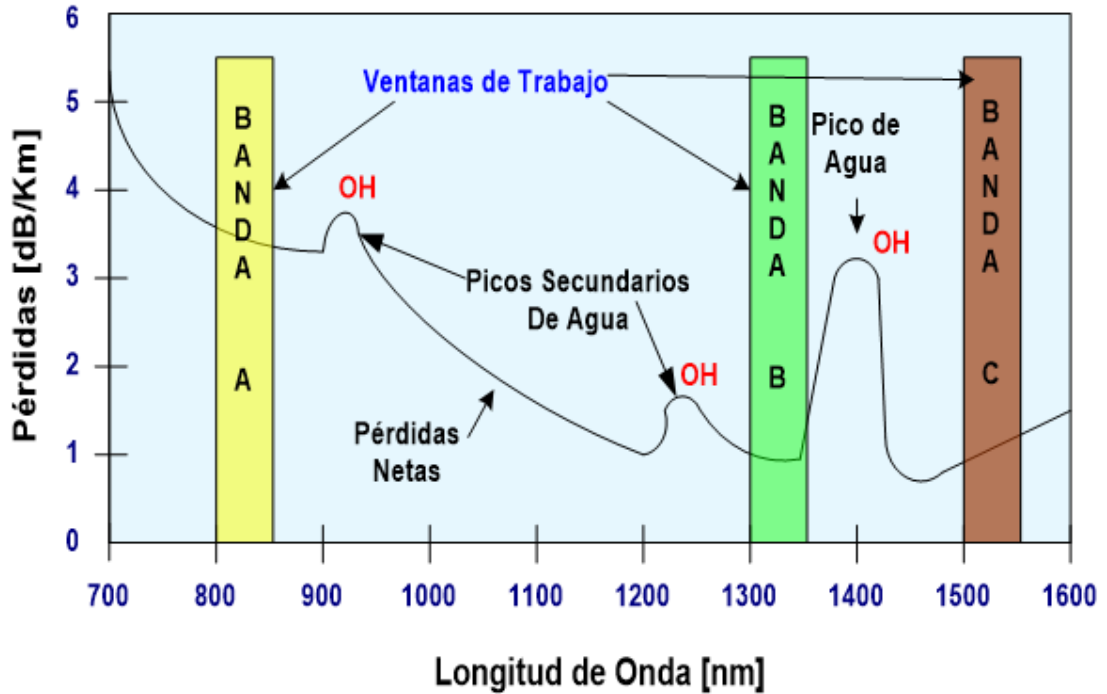


Figura 2.25. Longitudes Operativas de Fibras. [20]

La región que se suele llamar óptica es la que abarca aproximadamente desde 50 nm (ultravioleta) hasta 100 μm . Una fibra óptica presenta tres ventanas de transmisión, como se aprecia en la figura 2.25, la primera está situada entre 800 y 900 nm, la segunda entre 1300 y 1400 nm y la tercera entre 1500 y 1600 nm. [20]

Ancho de banda

En una fibra óptica el ancho de banda es una medida de la capacidad de transmisión de datos que ésta puede soportar. La dispersión (ensanchamiento de los pulsos de luz) es el fenómeno que limita el ancho de banda. Cuando las distancias son muy grandes, o la velocidad de transmisión muy alta, los pulsos se ensanchan, solapándose unos con otros perdiéndose información. Para evitarlo hay la máxima velocidad de datos. La

naturaleza de alta frecuencia de la luz permite llevar información a altas velocidades. En la actualidad los equipos de transmisión pueden modular luz a 2.4 Gbps, una frecuencia mayor que la que soportan los sistemas eléctricos convencionales. El concepto de comunicación óptica puede también ser aplicado a escalas pequeñas. Mientras que redes con Ethernet operan con aproximadamente 100 Mbps, las redes basadas en fibras ópticas pueden llegar hasta una frecuencia de 10 Gbps. [20]

Distancia

El efecto de la dispersión aumenta con la longitud de la fibra, y la dispersión limita el ancho de banda por lo que ambos parámetros están ligados. Por este motivo en las especificaciones técnicas de las fibras se suele indicar el producto BL, ancho de banda por distancia. Consecuentemente al aumentar la distancia de transmisión de una fibra se sacrifica la máxima tasa de datos que se puede alcanzar en dicha fibra y viceversa. Para largas distancias de transmisión son preferidas las fibras monomodo, a su vez éstas son usadas con fuentes de precisión como láseres; por lo que las largas distancias implican también un mayor costo en la implementación de un enlace de fibra óptica. [20]

2.2.4 Conceptos matemáticos en las comunicaciones por fibra óptica

En el siguiente tópico se detalla los puntos teóricos más relevantes para la comprensión de la transmisión de datos a través de la fibra óptica. [20]

Ley de Snell

La luz, al tener propiedades ondulatorias se propaga a distintas velocidades según el medio en que se encuentre, esto explica el fenómeno de refracción. Cuando un rayo de luz incide en una frontera hacia un nuevo material a cierto ángulo, el ángulo del haz refractado se acercará, o se alejará de la normal si el segundo medio tiene mayor o menor índice de refracción respectivamente. [20], [21]

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (5)$$

Ángulo crítico

Cuando la luz viaja desde un medio de mayor índice de refracción a uno de menor existe un valor de ángulo de incidencia en el que la luz refractada es paralela a la frontera, al valor de dicho ángulo se lo conoce como ángulo crítico; está dado por. [20], [21]

$$\text{sen } \theta_c = \frac{n_2}{n_1} \quad (6)$$

Reflexión total interna (TIR)

Los rayos de luz en una fibra óptica viajan confinados, esto se logra cuando el ángulo de incidencia dentro de la fibra es mayor que el ángulo crítico y se produce la reflexión total interna del haz de luz. En caso de ser una fibra escalón la luz viaja en forma de zigzag por el núcleo de la fibra. [20], [21]

$$\theta_i > \theta_c \Leftrightarrow TIR \quad (7)$$

Ángulo de aceptación y cono de aceptación

Para lograr TIR en la fibra hay que considerar que la luz se refracta primeramente desde el aire al núcleo de la fibra y después alcanza reflexión total interna si el ángulo de incidencia a la frontera entre el núcleo y el revestimiento es mayor que el ángulo crítico. Esto es posible si el ángulo refractado de la frontera aire – núcleo es el menor posible, lo que implica que el ángulo de incidencia a la fibra debe tener un valor máximo, a este ángulo se le denomina ángulo de aceptación. Como muestra la figura 2.26 al rotar este ángulo alrededor del eje se sigue respetando la misma geometría por lo que a nivel tridimensional se habla de cono de aceptación. El cono de aceptación es la región en el espacio donde se puede lanzar un haz de luz y asegurar la reflexión total interna. [21]

$$\theta_a = \text{arc sen}\left(\sqrt{n_1^2 - n_2^2}\right) \quad (8)$$

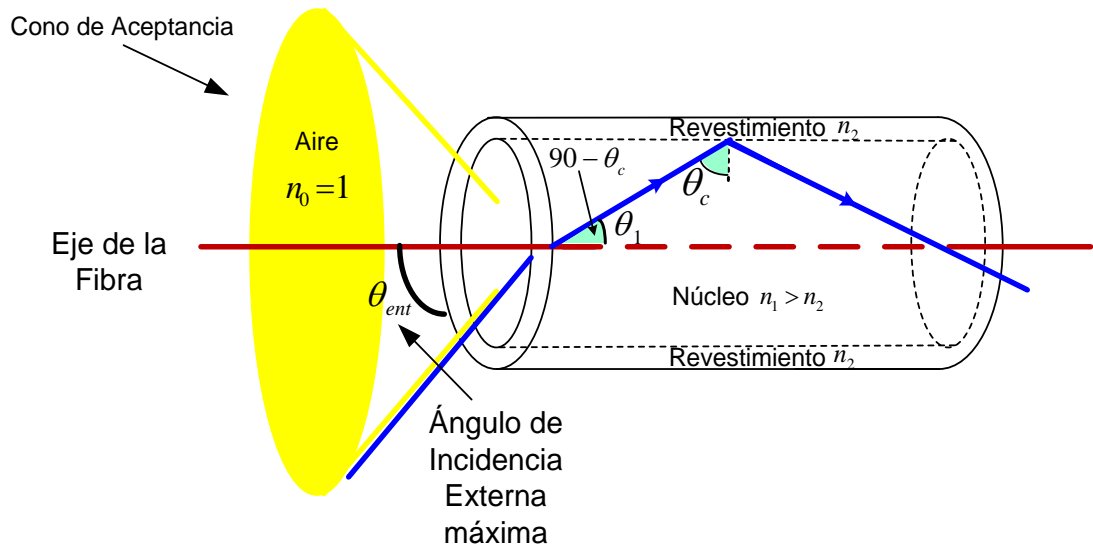


Figura 2.26. Cono de Aceptación. [20]

2.2.5 Teoría de propagación en fibra óptica

Para encontrar una fórmula matemática que modele el comportamiento del campo eléctrico a lo largo de la propagación en el medio confinado es necesario recurrir a las ecuaciones de Maxwell. El análisis es extenso y complejo y no se abordará en este documento. La evolución de la envolvente compleja de los pulsos ópticos a lo largo de una fibra óptica monomodo es gobernada por la ecuación no lineal de Schrödinger. [20]

Ecuación no lineal de Schrödinger

La ecuación no lineal de Schrödinger en ausencia de los efectos de polarización para un campo eléctrico $A(z, t)$, siendo z la distancia y t el tiempo, en notación de ingeniería es la ecuación diferencial parcial que se tiene a continuación. [20]

$$\frac{\partial A(z,t)}{\partial z} = -\frac{\alpha}{2}A - \beta_1 \frac{\partial A}{\partial t} + j \frac{\beta_2}{2} \frac{\partial^2 A}{\partial t^2} + \frac{\beta_3}{6} \frac{\partial^3 A}{\partial t^3} - j\gamma|A|^2A \quad (9)$$

Donde j la unidad imaginaria; α es la atenuación de la fibra; $\beta_k = \left. \frac{d^k \beta}{d\omega^k} \right|_{\omega=\omega_0}$ siendo $\beta(\omega)$ la constante de propagación de onda y $\omega_0 = 2\pi f_0 = 2\pi c/\lambda_0$ con f_0 , λ_0 la frecuencia/longitud de onda central de $A(z,t)$, respectivamente, siendo c la velocidad de la luz; γ es el coeficiente no lineal. Tales parámetros satisfacen la ecuación. [20]

Constante de propagación

En medios guiados la constante de propagación efectiva, β , es la proyección del número de onda k sobre el eje de la fibra. La velocidad a la que se desplaza la luz en un medio confinado puede escribirse como. [20]

$$v = \frac{\omega}{\beta} = \frac{c}{n} \quad (10)$$

El índice de refracción es función de la longitud de onda, por tanto, también de la frecuencia. Rescribiendo la constante de propagación en función de la frecuencia se tiene. [20]

$$\beta(\omega) = \frac{\omega n(\omega)}{c} \quad (11)$$

La constante de propagación se expande usando las series de Taylor alrededor de una frecuencia central ω_0 :

$$\beta(\omega) = \beta(\omega_0) + \beta'(\omega_0)(\omega - \omega_0) + \frac{1}{2}\beta''(\omega_0)(\omega - \omega_0)^2 + \frac{1}{6}\beta'''(\omega_0)(\omega - \omega_0)^3 \quad (12)$$

Los parámetros $\beta_k = \left. \frac{d^k \beta}{d\omega^k} \right|_{\omega=\omega_0} = \beta^{(k)}(\omega_0)$ tienen diferentes significados físicos. β_0 está relacionado con la velocidad de fase de la portadora óptica. β_1 determina la velocidad de grupo. β_2 es conocido como dispersión de velocidad de grupo, GVD, aunque en las especificaciones técnicas de fabricantes el factor que representa la dispersión cromática es D (ps/nm km), -del cual hablaremos más adelante en el presupuesto de potencia- y se relaciona con β_2 en la siguiente expresión. [21]

$$D = -\left(\frac{2\pi c}{\lambda^2}\right)\beta_2 \quad (13)$$

β_3 aporta a calcular la pendiente de dispersión $S(\lambda) = \frac{dD}{d\lambda}$ el cual es esencial para transmitir a grandes velocidades. [21]

Split Step Fourier Method

Es el método numérico para resolver la ecuación no lineal de Schrödinger, y así el modelo de propagación de pulsos a lo largo de una fibra monomodo. El método consiste en dividir la longitud de la fibra en un gran número de segmentos de longitud δz . SSMF asume que alrededor de una pequeña longitud los efectos de dispersión y linealidad sobre la propagación del campo óptico son independientes. La amplitud compleja de los pulsos ópticos es calculada usando la siguiente expresión. [20]

$$A(z + h, T) \approx \exp(h\hat{D}) \exp(h\hat{N}) A(z, T) \quad (14)$$

Donde \hat{D} es el operador lineal relacionado con dispersión y atenuación mientras \hat{N} representa la no linealidad. [20]

Presupuesto de pérdidas de un enlace de fibra óptica

Al diseñar un sistema de fibra óptica, uno de los principales elementos de diseño es el presupuesto de pérdida óptica, que es un cálculo detallado de potencia óptica disponible a la salida de la fuente de transmisión, la que se requiere en el detector del receptor y la disipación o atenuación de la potencia óptica en la trayectoria óptica entre el transmisor y el receptor. En esencia, el presupuesto de pérdida óptica es una indicación de cómo se utilizará la potencia óptica disponible. [21]

Potencia del transmisor

Es la potencia en las terminales de salida del transmisor que se entrega a la fibra. [21]

Sensitividad del receptor

Es la potencia mínima a la que el receptor puede percibir la información sin dificultad, si la potencia con la que llega la información que recibe es menor que su sensibilidad, éste no podrá identificar la información y será como si no la recibiera, así mismo, si la potencia recibida es mayor que la potencia máxima del receptor, este quedará cegado y no podrá distinguir la información. [20]

Ganancia del sistema

Es el total de potencia óptica disponible entre la potencia del emisor y la del receptor. El valor de la ganancia del sistema se halla por medio de la ecuación siguiente. [20]

$$SG(dBm) = P_{TX} - S_{RX} \quad (15)$$

Margen de seguridad

Es un valor de pérdidas que se estima en el enlace debido a: envejecimiento de los componentes (vida útil), factores ambientales, factor de dispersión, factor Jitter (es un ensanchamiento en el pulso digital, esto ocurre por fluctuaciones del reloj al cual se sincronizan los equipos), factor de reparo, factor por errores en el diseño del enlace. Este valor suele estimarse entre 5 a 10 dB o puede calcularse con la ecuación. [20]

$$SM = \text{factores ambientales} + \text{vida útil} + \text{dispersión (DCM)} + \text{margen de diseño [dB]} \quad (16)$$

Presupuesto de pérdidas

Es el valor máximo de pérdidas que pueden ocurrir en el enlace, producto de los conectores, empalmes y atenuación. Se calcula como la diferencia entre la ganancia del sistema y el margen de seguridad, como se muestra en la ecuación. [20]

$$LB = SG - SM \text{ [dB]} \quad (17)$$

Pérdidas por conectores

Las pérdidas en un conector se producen principalmente por factores como: mala alineación (radial y angular), reflexión en las superficies aire-vidrio, separación entre las fibras, variaciones en el tamaño del núcleo, de la apertura numérica de la fibra, etc. [21]

En el enlace se pueden calcular las pérdidas totales causadas por los conectores usando la ecuación. [21]

$$L_{CT} = L_C * N_C \text{ [dB]} \quad (18)$$

Pérdidas por empalmes

Las pérdidas por empalme ocurren en cada empalme que se coloque en el enlace de fibra óptica y se pueden atribuir a un gran número de factores, incluyendo un mal corte, el desalineamiento de los núcleos de las fibras, contaminación, desadaptación del índice de refracción, desadaptación del diámetro del núcleo, etc. En el enlace se pueden calcular las pérdidas totales causadas por los empalmes usando la ecuación. [21]

$$L_{ST} = L_S * N_S \text{ [dB]} \quad (19)$$

Pérdidas por otros componentes ópticos

Existen pérdidas por otros componentes ópticos pasivos que se utilicen adicionalmente en el enlace como splitters, combinadores, estrellas, etc. Se calculan con la siguiente ecuación, sumando las pérdidas individuales de cada componente. [21]

$$L_{OC} = L_{SPLITTER} * N_{SPLITTERS} + \dots \text{ [dB]} \quad (20)$$

Atenuación en el cable disponible

Es la disminución de la potencia de la señal a medida que esta se propaga a lo largo de la fibra. [21]

$$\alpha_{cd} = LB + L_C + L_S + L_{OC} \text{ [dB]} \quad (21)$$

Para un determinado modo de propagación, dicha reducción de potencia se produce de manera exponencial con respecto a la longitud recorrida. Al expresar esta relación con respecto a la longitud de la fibra se obtiene la constante de atenuación, según se calcula con la siguiente ecuación. [21]

$$\alpha_{long} = \frac{\alpha_{cd}[\text{dB}]}{\text{Longitud Total de la Fibra}[\text{Km}]} \quad (22)$$

Pérdida total en la fibra

Se puede calcular las pérdidas por atenuación en cada tramo de la fibra, usando la constante de atenuación, usando la ecuación. [21]

$$L_f = \text{Longitud Fibra [km]} * \alpha_{long} \text{ [dB]} \quad (23)$$

Potencia en el receptor

Es la cantidad de potencia que llega al emisor proveniente de la potencia emisor luego de haberse atenuado por todas las pérdidas del sistema, se calcula según la ecuación.

[21]

$$P_{RX} = P_{TX} + L_F + L_C + L_S + L_{OC} \quad [dB] \quad (24)$$

Rango dinámico

El receptor tendrá un límite máximo de Potencia de señal que este podrá recibir sin que la señal se distorsione. La diferencia entre la potencia máxima del receptor y la sensibilidad se conoce como rango dinámico. [21], [22]

Es importante que este valor no se exceda, por lo que debe ser tomado muy en cuenta cuando se diseñe los enlaces de fibra óptica, para ello se usa la condición descrita en la siguiente ecuación. [21], [22]

$$P_{RX} < S_{RX} + R_D \quad (25)$$

2.2.6 Diseño del ancho de banda

Otros factores que se debe considerar en el diseño de un enlace de fibra óptica son los relacionados a la información que se desea transmitir, lo que conlleva a tomar la decisión de escoger la tasa de datos, la frecuencia y la señalización para transmitir. [22]

Señalización

Existen tres principales formas de pulso en que se transmite información en un sistema digital: el filtro RC, el pulso rectangular y el pulso Gaussiano. En la figura 2.27 se muestran los diferentes tipos de señalización y se puede observar que cada uno de ellos posee una tasa de datos diferente, aunque tengan el mismo periodo. [22]

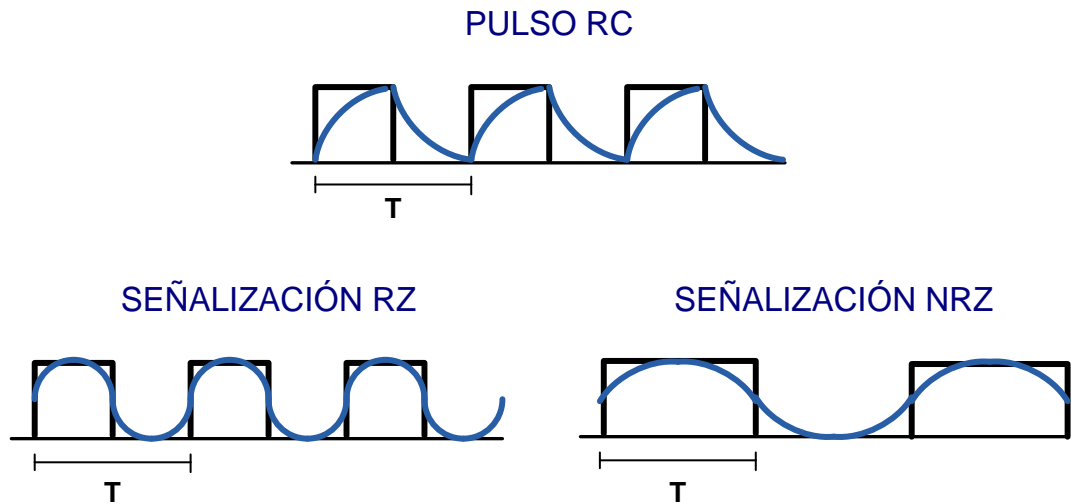


Figura 2.27. Tipos de Señalización. [22]

Tiempo de Subida de una Señal

El propósito del presupuesto del tiempo de subida es asegurar que el sistema sea capaz de operar apropiadamente a la tasa de bits deseada. En un filtro RC, el ancho de banda de salida depende del valor de resistencia y capacitor que se utilice en el circuito. Para un circuito de carga RC el tiempo entre dos puntos de voltaje está dado por. [22]

$$\Delta t = RC \ln \left(\frac{V_{fuente} - V_0}{V_{fuente} - V_f} \right) \quad (26)$$

El tiempo de subida se define como el lapso de tiempo transcurrido entre el 10% y el 90% de la respuesta de estado estable cuando la entrada es cambiada. Por tanto. [22]

$$t_r = RC \ln \left(\frac{V_{m\acute{a}x} - 0.1V_{m\acute{a}x}}{V_{m\acute{a}x} - 0.9V_{m\acute{a}x}} \right) = RC \ln 9 = 2.2 RC \quad (27)$$

La frecuencia de corte de un filtro RC es. [22]

$$\Delta f = \frac{1}{2\pi RC} \quad (28)$$

$$\Delta f = \frac{1}{2\pi \left(\frac{t_r}{2.2} \right)} = \frac{2.2}{2\pi t_r} = \frac{0.35}{t_r} \quad (29)$$

En Comunicaciones los dos formatos más usados son: el de No Retorno a Cero (NRZ) y el de Retorno a Cero (RZ), para el primer caso $\Delta f = R/2$ y para el segundo $\Delta f = R$. En consecuencia, la relación entre los tiempos de subida y la tasa de datos será. [22]

$$(a) T_{NRZ} = \frac{0.7}{\Delta f} \quad (b) T_{RZ} = \frac{0.35}{\Delta f} \quad (30)$$

Tasa de datos

La tasa de datos de un sistema está relacionada inversamente al tiempo de subida, el cual depende del tipo de señalización que se utilice. [22]

$$(a) R_{NRZ} = \frac{1}{T_{NRZ}} \quad (b) R_{RZ} = \frac{1}{T_{RZ}} \quad (31)$$

2.2.7 Presupuesto de tiempo de subida

Para calcular el presupuesto de tiempo de subida en un enlace de fibra óptica se necesitan los tiempos de subida del transmisor, el receptor y el de la fibra. [22]

$$T_{sys} = \sqrt{(T_{TX})^2 + (T_{RX})^2 + (T_F)^2 + (T_{MODAL})^2} \quad (32)$$

El tiempo de subida del transmisor viene dado por la electrónica de su diseño, por lo que viene especificado en la hoja de datos del fabricante, los valores típicos oscilan entre 0.05 ns para los láseres y de 2 ns para LEDs. En el caso del receptor viene dado por circuito de control de la luz. [22]

El tiempo de subida de la fibra está dado por la dispersión en la misma. [22]

La dispersión total en la fibra óptica que puede dividirse en dos categorías principales, como se muestra en la figura 2.28, del lado izquierdo se encuentra la dispersión presente en las fibras monomodo y del lado derecho la de la fibra multimodo. [22]

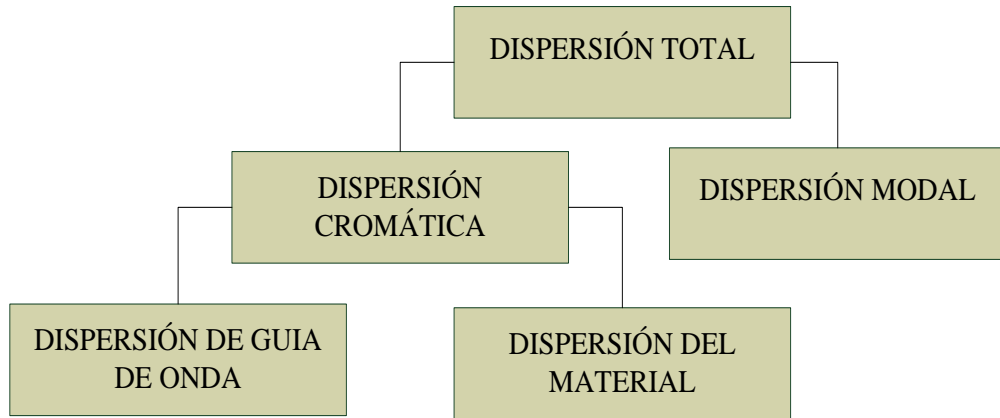


Figura 2.28. Tipos de dispersión en la fibra. [22]

Por lo tanto, el tiempo de subida para una fibra monomodo sería:

$$T_{sys-monomodo} = \sqrt{(T_{TX})^2 + (T_{RX})^2 + (T_{CH})^2} \quad (33)$$

Y para la fibra multimodo:

$$T_{sys-multimodo} = \sqrt{(T_{TX})^2 + (T_{RX})^2 + (T_M)^2} \quad (34)$$

Dispersión Cromática

Las fibras monomodo, usadas en redes ópticas rápidas, están sujetas a la dispersión cromática (CD), que causa un ensanchamiento de los pulsos de luz según la longitud de onda, véase figura 2.29. [22]

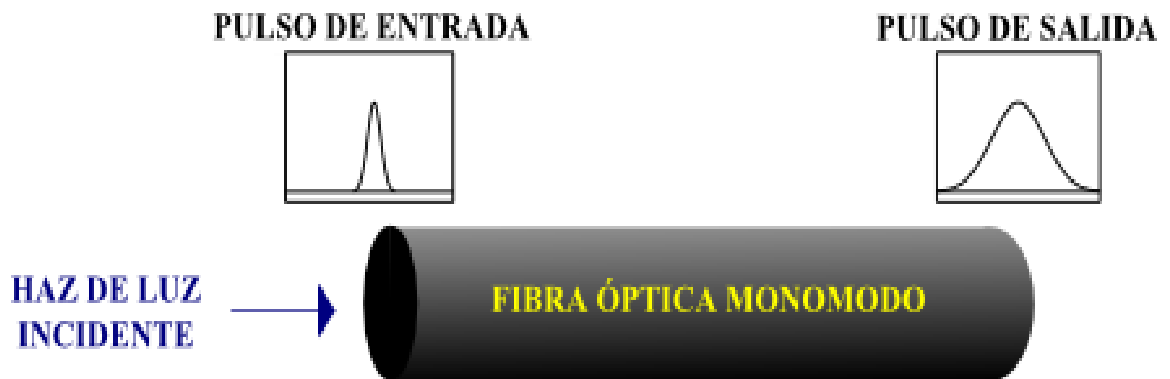


Figura 2.29. Dispersión cromática en la fibra óptica. [22]

Es importante destacar que la contribución de CD es determinante en el funcionamiento del sistema para largas distancias y altas tasas de transmisión. En

distancias cortas de pocos kilómetros la contribución a la dispersión total es pequeña y no afecta gravemente al comportamiento del sistema. [22]

La forma de calcular el tiempo de subida causado por la dispersión cromática se muestra en la ecuación:

$$T_F = T_{CH} = |D_{CH}| \sigma_\lambda L \quad \text{donde } D_{CH} = |D_m + D_{wg}| \quad (35)$$

Los valores de D_m y D_{wg} se encuentran dados por el fabricante.

2.2.8 Ruido

Es todo aquello que interfiere o distorsiona la señal que viaja a través de la fibra óptica y al momento de recuperar la señal enviada. El ruido nunca va a desaparecer, pero se procura hacer que la relación de la potencia de señal a ruido sea mínima. [22]

En los fotodetectores se tienen dos tipos de ruido: Ruido Térmico y Ruido de Disparo. Ambos son ruidos blancos, es decir de energía repartida en todas las componentes espectrales, algo que es imposible de simular, sin embargo, para efectos prácticos el ruido tiene un ancho espectral igual al tiempo de muestreo de nuestro proyecto. El ruido blanco gaussiano $w(t, \zeta)$ es un proceso estocástico que cumple con la siguiente propiedad:

$$Var[kw(t, \zeta)] = k^2 Var[w(t, \zeta)]; \quad k \in \mathbb{R} \quad (36)$$

Esta propiedad es usada al crear ruido gaussiano de media cero y varianza uno, pero que al ser multiplicado por una constante, la señal resultante tiene valor cuadrático medio (desviación estándar) la misma constante. [22]

Ruido Térmico o de Johnson

Este es el tipo de ruido más conocido en los sistemas de comunicaciones electrónicas. Se produce por el movimiento de electrones o iones en el interior de los conductores, este movimiento hace que se produzca un incremento en la temperatura del sistema., lo que puede llegar a contribuir de manera significativa al Ruido General existente en la recepción de la señal óptica. [22]

$$int_{rms}^2 = \frac{4 K_B T B}{R_L} \quad (37)$$

Ruido de disparo

En el fotodiodo semiconductor el ruido de disparo se origina por la generación y recombinación de electrones vacíos. Esto produce fluctuaciones que afectan a la relación señal ruido. Para calcular el valor del ruido de disparo es necesario conocer el valor de la corriente oscura y la corriente del fotodetector, la última viene dada como dato del fabricante. El ruido de disparo está dado por la ecuación. [22]

$$ins_{rms}^2 = [2 e (I_d + I_{ph}) B] \quad (38)$$

2.2.9 Relación señal ruido

Es un término que se utiliza para cuantificar la cantidad presente de ruido en alguna señal dada. Su expresión para el caso de la fibra óptica es. [11], [22]

$$SNR = \frac{\text{Potencia Señal}}{\text{Potencia de Ruido Total}} = \frac{P_{S_{rms}}}{P_{nt_{rms}} + P_{ns_{rms}}} \quad (39)$$

Para potencia óptica incidente constante:

$$SNR = \frac{(\Re P_O)^2 R_L}{4 K_B T B + [2 e (I_d + I_{ph}) B] R_L} \quad (40)$$

Para potencia óptica sinusoidal:

$$SNR = \frac{\frac{m^2}{2} (\Re P_O M)^2 R_L}{4 K_B T B + [2 e (I_d + I_{ph}) B] M R_L} \quad (41)$$

2.2.10 Reflectometría y transmisometría ópticas

La reflectometría óptica es una disciplina enmarcada en el campo de la metrología óptica encargada del estudio de los efectos producidos por la reflexión de ondas electromagnéticas en el dominio de las frecuencias ópticas en su desplazamiento a lo largo de medios de propagación no homogéneos. Cuando el estudio se centra en la onda refractada en lugar de la reflejada, a la disciplina se la conoce como

transmisometría, aunque generalmente el nombre de reflectometría se suele emplear haciendo referencia a las dos disciplinas en conjunto. [23]

Toda onda electromagnética incidente sobre una discontinuidad de su medio de propagación sufre una reflexión y una refracción, descomponiéndose en dos nuevas ondas cuyas direcciones de propagación viene dada por la ley de Snell. La figura 2.30 muestra una gráfica del proceso de la óptica geométrica. [23]

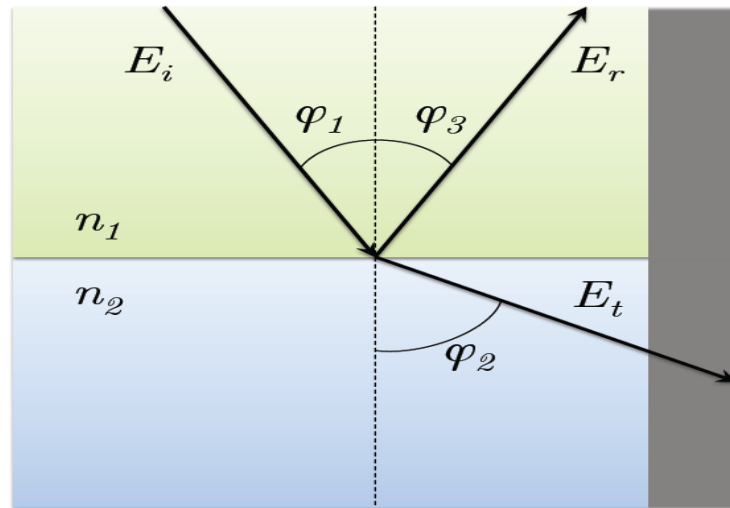


Figura 2.30. Óptica geométrica. [23]

$$n_1 = n_3 \quad n_1 \text{sen}(\varphi_1) = n_2 \text{sen}(\varphi_2) \quad (42)$$

La relación de amplitudes y fases de las amplitudes complejas de las ondas reflejadas y refractadas viene dada por una ponderación a cargo de dos magnitudes complejas llamadas coeficientes de reflexión y coeficiente de transmisión. [23]

$$E_i(f) = p(f)E_r(f) + \tau(f)E_t(f) \quad (43)$$

Los coeficientes de reflexión y transmisión son dependientes de la frecuencia, por lo que constituyen dos funciones de transferencia complejas de la discontinuidad. [23]

Se puede generalizar este análisis a un medio en el que, en lugar de existir discontinuidades localizadas, existe una inhomogeneidad distribuida a lo largo de sus dimensiones físicas. Este es, en general, el modelo físico de todo dispositivo fotónico. La naturaleza de dicha inhomogeneidad será la responsable de configurar la respuesta impulsiva y la función de transferencia compleja del dispositivo. La medida de los

coeficientes de reflexión y transmisión de los sistemas ópticos es el objetivo de la reflectometría. Los dispositivos basados en reflectometría están contruidos mediante montajes interferométricos. [23]

Reflectometría óptica en el dominio del tiempo:

Los métodos reflectométricos en el dominio del tiempo miden los retardos producidos entre las diferentes reflexiones que tienen lugar en el interior de los dispositivos ópticos bajo pruebas. Para ello se excita el sistema bajo pruebas con un pulso óptico procedente de un láser, cuyas características de amplitud y anchura temporal definirán la resolución espacial de la medida y la máxima longitud permitida del DUT. Las señales recibidas corresponden a las ondas reflejadas en las discontinuidades del índice de refracción del DUT y se detectan en sentido contrapropagante a la señal de prueba. [24]

Conocida la velocidad de fase de la señal de prueba en el medio de propagación empleado (esto es, conocido el índice de refracción efectivo del medio) y teniendo en cuenta que los pulsos recorren una distancia igual al doble de la longitud del DUT en un camino de ida y vuelta, es posible establecer una relación tiempo-espacio para una medida de la localización espacial de los eventos que producen las reflexiones. [24]

$$z = \frac{c}{2n} t \quad (44)$$

Esta técnica es empleada por un instrumento de medida en comunicaciones ópticas, el OTDR (Optical Time-Domain Reflectometer), usado para la monitorización de enlaces de fibra óptica de longitudes en el orden de los kilómetros. [24]

Reflectometría óptica en el dominio de la frecuencia:

La reflectometría óptica en el dominio de la frecuencia emplea técnicas interferométricas para la caracterización de dispositivos ópticos de longitudes interiores a las que puede medir un OTDR. Sus técnicas se emplean para la construcción de instrumentos de medida de componentes y dispositivos ópticos de longitudes del orden de los pocos centímetros a los pocos metros en función de la resolución frecuencial del instrumento. [24], [25]

Este tipo de técnicas emplean la interferometría óptica para implementar detección coherente, la cual permite realizar medidas de la fase del DUT. Además, permite la implementación de medida en tiempo real y métodos de gran resolución. [27]

2.2.11 Interferometría óptica

La interferometría óptica es una disciplina enmarcada en el campo de la metrología óptica encargada del estudio de los efectos producidos por la interferencia entre ondas electromagnéticas a frecuencias ópticas. [28]

Se basa en el estudio de la interferencia entre ondas electromagnéticas de prueba, una de las cuales procederá del paso a través de un medio físico o dispositivo que se desea estudiar y la otra procederá de un camino de referencia. El patrón de interferencia obtenido es máximo cuando las ondas interfieren en fase y nulo cuando interfieren en oposición de fase. Estudiando la forma de onda de este patrón de interferencia se puede obtener información sobre la naturaleza física del dispositivo estudiado. La figura 2.31 muestra el patrón de interferencia generado por un interferómetro de Michelson. [28]

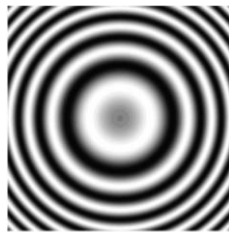


Figura 2.31. Patrón de interferencia generado por un interferómetro de Michelson. [28]

En metrología de sistemas ópticos, esta disciplina se emplea para medir la respuesta impulsiva. Se trata de una herramienta particularmente potente en la caracterización de dispositivos fotónicos ya que, mediante el montaje adecuado de una red óptica, es posible medir la fase del sistema para la estimación de su retardo de grupo. Dicha red óptica y cualquier instrumento diseñado a tal efecto, recibe el nombre de interferómetro. [28]

Un interferómetro es un sistema óptico diseñado bien en soporte de fibra, bien en espacio libre, en el que dos señales interfieren tras haber recorrido dos caminos ópticos diferentes a los que se denomina brazos del interferómetro. En tecnología de espacio libre se emplean componentes micro-ópticos como lentes y espejos para realizar el

guiado de las señales. En fibra óptica se emplean componentes fabricados en fibra como circuladores ópticos y acopladores direccionales para realizar el guiado de las señales. [29]

El interferómetro por sí solo produce una función de filtrado en peine debida a la interferencia de las señales procedentes de los dos brazos. La diferencia de caminos introduce un desfase entre las señales que determina el FSR (Free Spectral Range), esto es, el ancho de los lóbulos del peine. En el caso de un interferómetro FPI (Fabry-Perot Interferometer) la presencia de lóbulos laterales se debe a las resonancias producidas en cada camino de ida y vuelta. En el caso de un interferómetro de Mach-Zehnder la forma de filtro en peine se debe al desfase que introduce la diferencia de longitud entre los dos brazos del interferómetro. [29]

Detección directa vs. Detección coherente:

Los receptores de comunicaciones ópticas son capaces de generar señales de corriente proporcionales a la potencia óptica incidente. Este hecho impide la realización de medidas directas de la fase de las señales recibidas. Los instrumentos de medida que emplean esta configuración se dice que implementan detección directa. La figura 2.32 muestra el símbolo de la detección directa. [29]

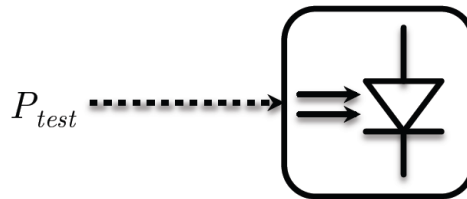


Figura 2.32. Detección directa. [29]

$$I_{pd} = \Re P_{test} \tag{45}$$

Es posible preparar un montaje que entregue a un fotodiodo la potencia óptica procedente de dos señales diferentes. El resultado de la interferencia de ambas señales contiene información sobre su diferencia de fases. Si ambas señales tienen una fuente común y una de ellas procede del paso por un DUT, se dice que el interferómetro implementa detección coherente. La figura 2.33 muestra el símbolo de la detección coherente. [29]

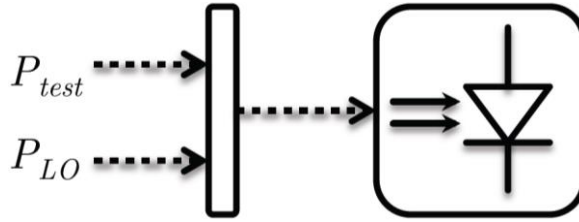


Figura 2.33. Detección coherente. [28]

$$I_{pd} = \Re(P_{test} + P_{LO} + 2\sqrt{P_{test}P_{LO}} \cos(\Delta\phi)) \quad (46)$$

A la señal procedente del DUT la llamamos señal de prueba, mientras que a la procedente de la fuente la llamamos oscilador local. El término de interferencia se diferencia en frecuencia del resto de términos al estar modulado por la diferencia de fases entre señal de prueba y el oscilador local. La anterior ecuación muestra cómo es posible aprovechar el término de interferencia para medir la fase del DUT, algo imposible de hacer mediante detección directa. [29]

El interferómetro de Mach-Zehnder:

Un montaje interferométrico muy práctico a la hora de diseñar instrumentación óptica de medida es el interferómetro Mach-Zehnder, ya que permite una sencilla implementación en fibra óptica. La figura 2.34 muestra la Implementación en fibra óptica del interferómetro de Mach-Zehnder. [29]

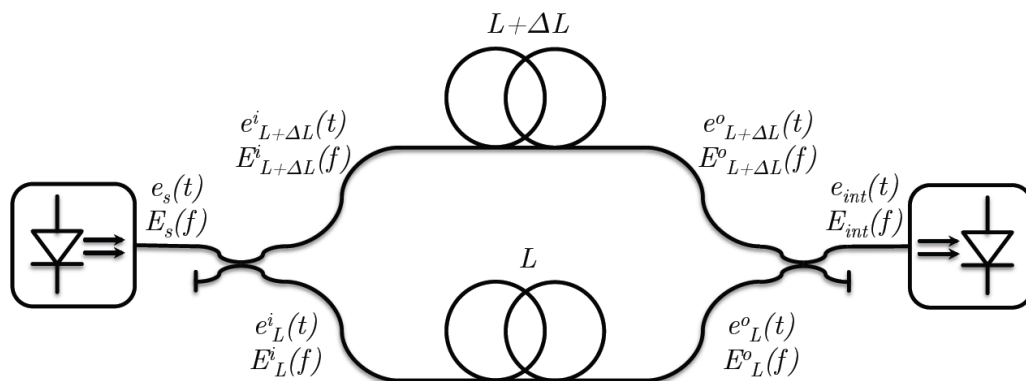


Figura 2.34. Implementación en fibra óptica del interferómetro de Mach-Zehnder. [29]

Este montaje consta de un acoplador direccional de entrada a 3dB que reparte una señal procedente de una fuente óptica (oscilador local) entre los dos brazos del interferómetro de forma que las señales a la entrada de cada brazo se encuentran en cuadratura. [29]

$$E_L^i + \Delta L = \sqrt{\frac{1}{2}} E_s \quad E_L^i = j \sqrt{\frac{1}{2}} E_s \quad (47)$$

Los brazos se construyen mediante sendas fibras ópticas siendo una de ellas ΔL metros más larga que la otra. Esta diferencia de longitud actúa como línea de retardo de la señal, de forma aparece un término dependiente de dicha diferencia de longitud. [29]

$$E_L^O + \Delta L = \sqrt{\frac{1}{2}} E_s e^{j\beta(L+\Delta L)} \quad (48)$$

$$E_L^O = j \sqrt{\frac{1}{2}} E_s e^{j\beta L} \quad (49)$$

Las señales procedentes de los brazos del interferómetro interfieren nuevamente en cuadratura al atravesar otro acoplador direccional a 3dB. [29]

$$E_{int} = \frac{1}{2} E_s (e^{j\beta(L+\Delta L)} - e^{j\beta L}) = E_s e^{j\beta(L+\frac{\Delta L}{2}+\frac{\pi}{2})} \text{sen} \left(\beta \frac{\Delta L}{2} \right) \quad (50)$$

El interferograma resultante es registrado por un fotodetector a la salida del interferómetro. La función de filtrado resultante es un peine cuyos lóbulos, debido al seno en la ecuación, están equiespaciados y su ancho es el FSR. [29]

$$FSR = \frac{c}{n\Delta L} \quad (51)$$

2.2.12 Reflectómetro óptico en el dominio del tiempo

El OTDR conocido como reflectómetro óptico en el dominio del tiempo, es muy versátil para muchas aplicaciones. Es importante seleccionar un OTDR con las especificaciones adecuadas para la aplicación necesaria. Con los recientes avances en la modularidad, algunos OTDRs, tales como la flexibilidad de la familia JDSU T-BERD/MTS que pueden configurarse para realizar pruebas en casi cualquier red de fibra óptica, monomodo o multimodo ya sea en recorridos cortos o largos. [26]

El uso de un OTDR se define ampliamente como un proceso de dos pasos:

1.- Adquisición: El OTDR capta los datos y muestra los resultados numéricamente o gráficamente. [26]

2.- Medición: Los técnicos analizan los datos y en base a los resultados, decidirá ya sea para almacenar, imprimir, o ir a la adquisición de la próxima fibra. [26]

La figura 2.35 muestra un modelo de OTDR de la familia JDSU T-BERD/MTS. [26]



Figura 2.35. Modelo de OTDR de la familia JDSU T-BERD/MTS. [26]

Adquisición de datos del OTDR

La mayoría de los OTDRs modernos pueden seleccionar automáticamente los parámetros de adquisición óptimos para una fibra en particular, mediante el envío de impulsos de prueba en un proceso conocido como auto-configuración. El uso de la función de configuración automática, los técnicos deberán de seleccionar la longitud de onda (o longitudes de onda) para probar, adquirir (valor medio) el tiempo, y los parámetros de la fibra (índice de refracción, por ejemplo, si no se han ingresado aún). [26]

A continuación, se describen los tres enfoques principales que se utilizan al configurar un OTDR:

1.- Los técnicos deben permitir que el OTDR se auto configure y aceptar la adquisición de los parámetros del OTDR seleccionado. [26]

2.- Los técnicos con más experiencia permiten que el OTDR se auto configure, luego analizar los resultados brevemente y cambiar uno o más parámetros de adquisición para optimizar la configuración de los requisitos de la prueba en particular. [26]

3.- Técnicos experimentados optan por no utilizar la función de configuración automática por completo e introducir los parámetros de adquisición basados en la experiencia y el conocimiento del enlace en pruebas. [26]

Típicamente, cuando se prueban los cables de varias fibras ópticas, una vez que se han seleccionado los parámetros de adquisición apropiados, ahí quedarán bloqueados. Estos mismos parámetros se utilizan para todos los tipos de fibra, esta característica acelera drásticamente el proceso de adquisición y establece la coherencia en los datos, lo cual es útil para el análisis o la comparación de las fibras. Las siguientes subsecciones discuten varios parámetros de adquisición y sus efectos sobre la traza resultante del OTDR. [26]

Nivel de inyección

El nivel de inyección se define como el nivel de potencia en el que el OTDR inyecta luz en la fibra que está bajo prueba. Cuanto mayor sea el nivel de inyección, mayor será el rango dinámico. Si el nivel de inyección es bajo, la traza de OTDR contendrá ruido, y la medición con exactitud se verá disminuida. Debido a las pobres condiciones en el inicio, lo que da como resultado niveles bajos o disminuidos de inyección. [27]

La presencia de suciedad en la cara de conectores y pigtails de baja calidad o cables de conexión (patch cords) dañados, son las principales causas de los bajos niveles de inyección. Es importante que todos los puntos de conexión físicos estén libres de polvo o suciedad en un sistema óptico. Con diámetros de núcleo de menos de 10 μm en sistemas de un solo modo, la presencia inclusive de una partícula de suciedad o polvo de 4 μm (aproximadamente del tamaño de la materia en partículas en el humo del cigarrillo) puede degradar gravemente los niveles de inyecciones. [27]

Los kits de limpieza están disponibles para sistemas por fibra óptica, tales kit o herramientas básicas son: la solución de limpieza isopropílico, papel Joseph, esterilizaciones de aire comprimido, y toallitas impregnada; todas listas para su uso.

Los métodos más avanzados utilizan productos de limpieza de cassette y sistemas de limpieza integrados, como el sistema JDSU CleanBlast. [27]

El acoplamiento de conectores sucios al conector OTDR puede rayar al conector OTDR, y de manera permanente degradar las condiciones iniciales. Algunos OTDRs, tales como la familia T-BERD/MTS, muestran el nivel de inyección medido durante la adquisición en tiempo real o justo antes de promediar. Para determinar la calidad relativa del nivel de inyección, el OTDR debe mirar distancias cortas. Al observar la retro dispersión que no es más que regresar el pulso de lanzamiento y compararlo con el valor esperado. [27]

A veces es posible que el nivel de inyección mostrado sea inaceptable, cuando en la realidad es aceptable, lo que puede ocurrir si hay un atenuador o un divisor en el sistema cerca del OTDR. En este caso, el nivel de retro dispersión será menor de lo esperado según lo medido por el medidor de nivel de inyección. [27]

Longitud de onda

El comportamiento de un sistema óptico está directamente relacionado con su longitud de onda de transmisión. La fibra óptica presenta diferentes características de pérdidas en diferentes longitudes de onda. Además, los valores de pérdida de empalme también difieren en diferentes longitudes de onda. [27]

En general, se deberá probar la fibra usando la misma longitud de onda utilizada para la transmisión. Por lo tanto, 850 y/o 1300 nm de longitud de onda se utilizan para los sistemas multimodo y 1310 y/o 1550nm longitud de onda se utilizan para sistemas monomodo. Al probar una sola longitud de onda, considerar los siguientes parámetros:

1.- Para obtener un rango dinámico dado, utilizando una longitud de onda de 1550 nm verá distancias más largas debajo de la misma fibra de una longitud de onda de 1310 nm debido a la menor atenuación en la fibra. [27]

a.- 0.35 dB/Km a 1310 nm significa que aproximadamente 1 dB de la señal se pierde cada 3 Km. [27]

b.- 0.2 dB/Km a 1550 nm significa que aproximadamente 1 dB de la señal se pierde cada 5 Km. [27]

2.- La fibra monomodo tiene un modo de campo de diámetro mayor en 1550 que en 1310 nm y en 1625 que en 1550 nm. Campos de modo más grandes son menos sensibles a la desviación lateral durante el empalme, pero que son más sensibles a las pérdidas sufridas por flexión durante la instalación o en el proceso de cableado. [27]

a.- 1550 nm es más sensible a los dobleces en la fibra que 1310 nm, conocido como macro flexión, tal como se muestra en la figura 2.36. [27]

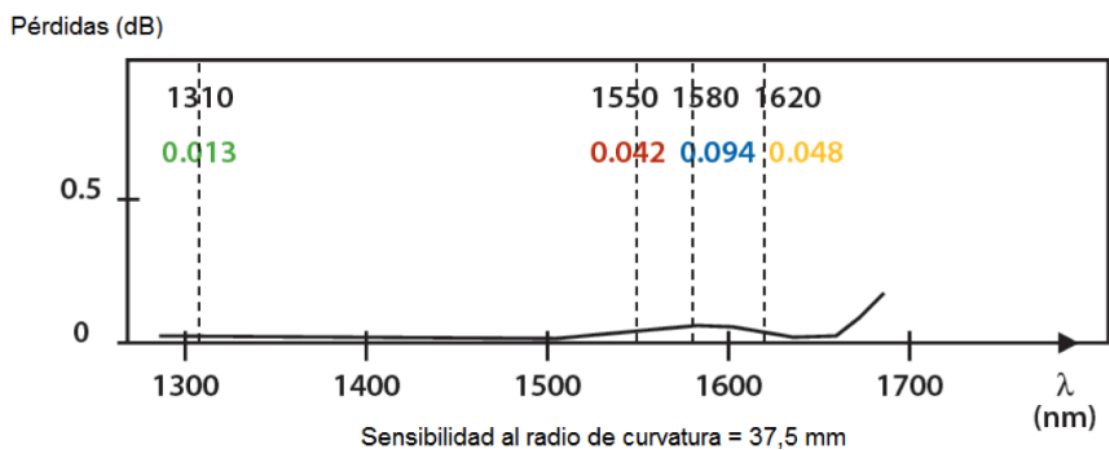


Figura 2.36. Efectos de la macro flexión sobre una fibra. [27]

b.- 1310 nm generalmente medirá empalme y pérdidas en los conectores superiores a 1550 nm. Estos resultados provienen de un estudio de más de 250 empalmes donde se muestran los valores de 1310 nm para ser normalmente superior en 0.02 dB sobre los valores de 1550 nm para la dispersión desplazada de fibra, tal como se muestra en la figura 2.37. [27]

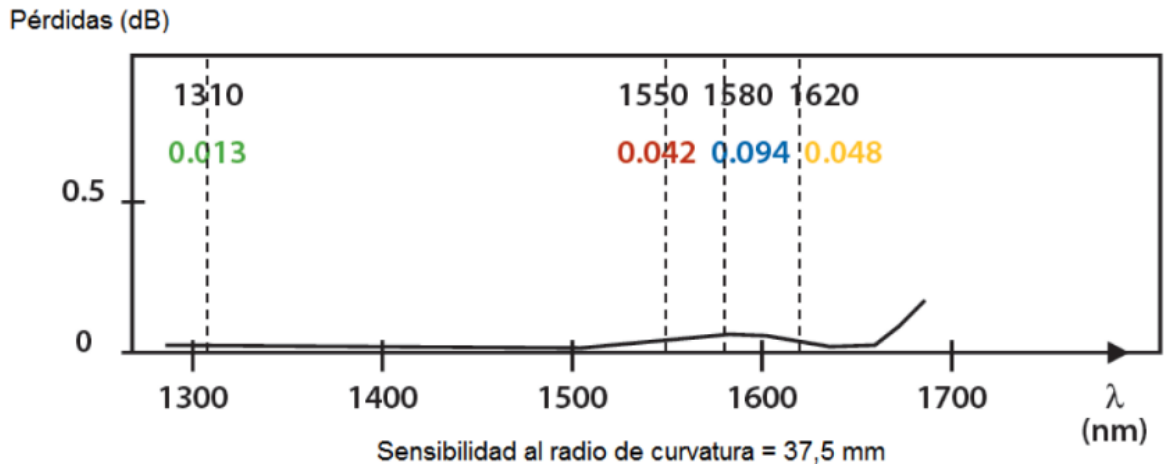


Figura 2.37. Efectos de la dispersión desplazada. [27]

Ancho de pulso

La duración de la anchura del pulso del OTDR controla la cantidad de luz que se inyecta en una fibra. Cuanto más tiempo la anchura de impulso, mayor será la cantidad de energía de la luz inyectada. La energía más ligera inyectada, mayor es la cantidad de luz, que se retro dispersa o reflejada por la fibra hasta el del OTDR. [27]

Anchos de pulso largos se utilizan para poder ver a grandes distancias por un cable de fibra. Anchos de pulso largos también producen zonas más largas en la onda del trazado del OTDR donde las mediciones no son posibles, conocido como la zona muerta del OTDR. Por otro lado, los anchos de pulsos cortos se inyectan los niveles más bajos de luz, sino que también reducen la zona muerta del OTDR. [27]

En la figura 2.38 se muestras varias mediciones realizadas en un OTDR con diferentes anchos de pulso. [27]

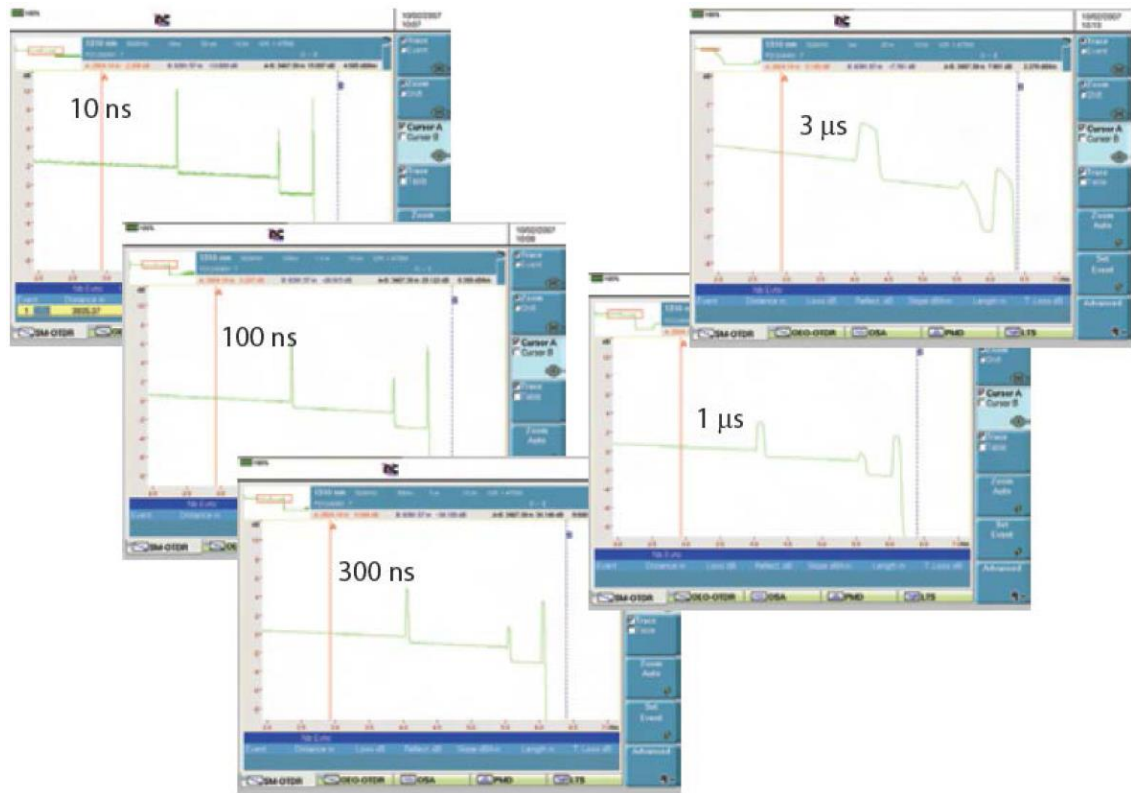


Figura 2.38. Medición de la fibra utilizando diferentes anchos de pulso. [27]

La duración de la anchura de impulso se da generalmente en nanosegundos, pero también puede ser estimada en metros de acuerdo con la siguiente fórmula:

$$D = \frac{cT}{2n} \quad (52)$$

Dónde:

c = es la velocidad de la luz en el vacío ($2,99792458 \times 10^8$ m/s)

T = es la duración del pulso en ns

n = es el índice de refracción

Alcance

El rango de un OTDR se refiere a la distancia máxima desde el cual el OTDR puede adquirir muestras de datos. Cuanto más largo sea el rango, aún más la OTDR disparará pulsos por la fibra. El rango se ajusta generalmente al doble de la distancia hasta el

extremo de la fibra. Si el rango se establece correctamente, la onda de la traza puede contener artefactos de medición, como fantasmas. [28]

Promediando

El detector OTDR funciona en los niveles de potencia óptica extremadamente bajas (tan bajas como 100 fotones por metro de la fibra). Promediar, es el proceso por el cual cada punto de adquisición se muestrea repetidamente, y se promedian los resultados a fin de mejorar la relación señal/ruido (SNR). [28]

Seleccionando el tiempo de adquisición o número de promedios dados, los técnicos de control sabrán el proceso de promediar en el OTDR. Mientras más largo sea el tiempo o cuanto mayor sea el número de promedios, la señal de la onda se mostrará en condiciones de ruido aleatorio. [28]

La relación entre el tiempo de adquisición (número de promedios) y la cantidad de mejora de la SNR se expresa por la siguiente ecuación:

$$\Delta SNR = 5 \log \sqrt{N} \quad (53)$$

Donde N es la relación de las dos medias.

Por ejemplo, con una adquisición promedio de 3 minutos mejorará el rango dinámico en 1.2 dB en comparación si utilizamos una adquisición promedio de 1 minuto. [28]

El promedio mejorará la relación de señal ruido (SNR) al aumentar el número de adquisiciones, pero también aumenta el tiempo que se necesita para un promedio del trazado. Sin embargo, de acuerdo con la ecuación, más allá de cierto tiempo de adquisición, ninguna ventaja se obtiene debido a que la señal permanece sola. En teoría, multiplicando el tiempo medio de adquisición por 4 proporcionará un aumento de 1.5 dB en la gama dinámica. [28]

2.2.13. Mediciones del OTDR

La mayoría de los OTDR modernos realizan mediciones automáticas, con muy poca participación de los técnicos. [29]

Interpretación de eventos

En general existen dos tipos de eventos: reflexivo y no reflexivo. [29]

a.- Eventos reflectantes

Los eventos reflectantes ocurren donde surge discontinuidad en la fibra, causando un cambio brusco en el índice de refracción. Los eventos reflectantes pueden ocurrir en las pasas, las uniones de conectores, empalmes mecánicos, o el final de fibra. Para eventos reflectantes, la pérdida en los conectores es típicamente alrededor de 0.5 dB. Para los empalmes mecánicos, sin embargo, la pérdida varía en 0.1 dB a 0.2 dB. [29]

La figura 2.39 muestra el evento reflectante debido a un cruce del conector. [29]

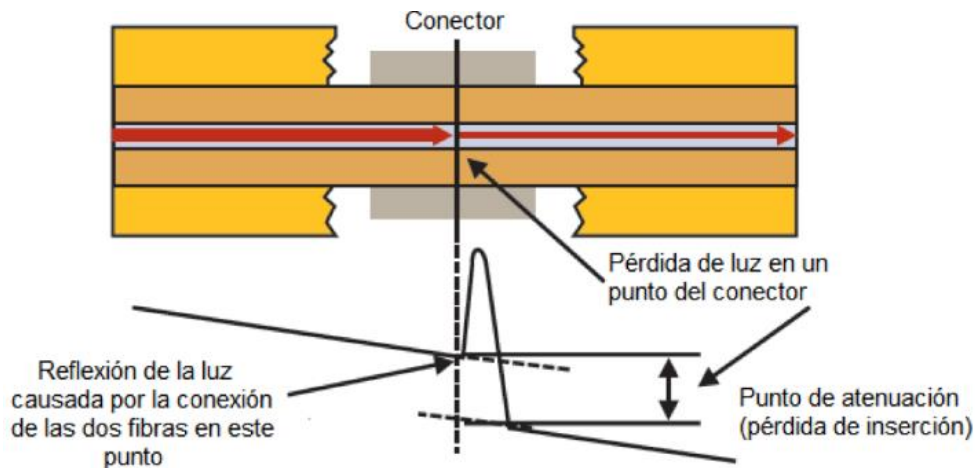


Figura 2.39. Evento reflectante debido a un cruce del conector. [29]

Si se producen dos eventos reflectantes muy juntos, el OTDR puede tener problemas de la medición de la pérdida de cada evento. En este caso, muestra la pérdida de las pruebas combinadas, que normalmente se produce cuando se mide una longitud de fibra corta, como un puente de fibra. La figura 2.40 muestra el evento de reflexión en dos uniones de conectores muy cercanos. [29]

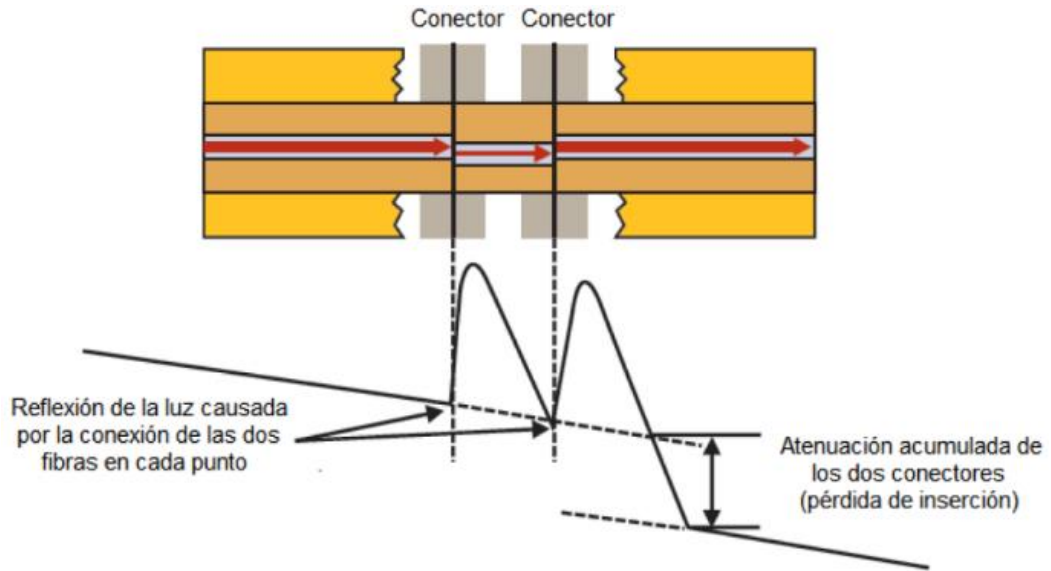


Figura 2.40. Evento de reflexión en dos uniones de conectores muy cercanos. [29]

En el caso de un extremo de la fibra, el evento de reflexión caerá en el ruido y evita tomar la medición de la atenuación. La figura 2.41 muestra el evento de reflexión extremo. [29]

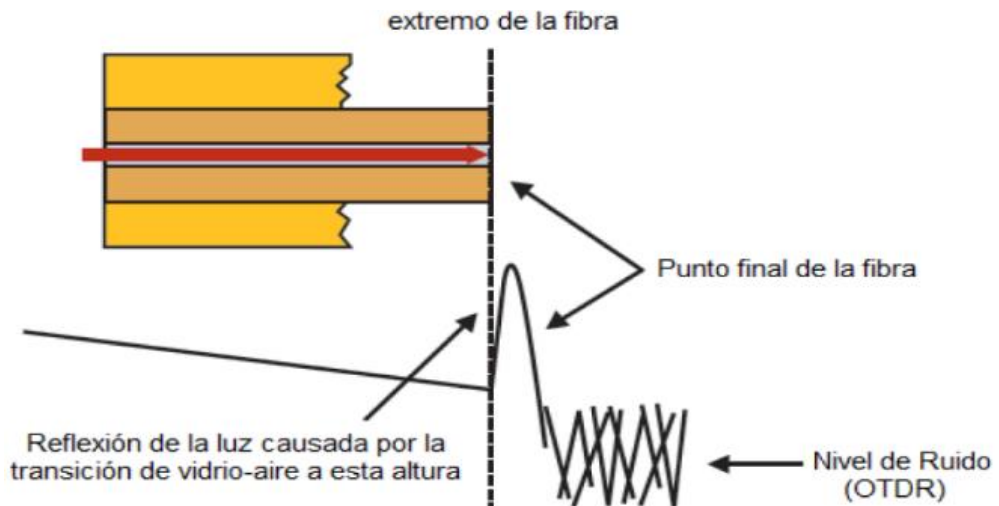


Figura 2.41. Evento de reflexión en un extremo de la fibra. [29]

b.- Eventos no reflectantes

Los eventos no reflectantes producen discontinuidades en donde está ausente la fibra y se producen generalmente mediante empalmes de fusión o pérdidas por flexión, tales como macro curvaturas. Valores típicos de pérdida están en el intervalo de 0.02 a 0.1

dB, y dependerá del equipo de corte y empalme y del técnico/operador. En la figura 2.42 se muestra el evento de no reflexión. [29]

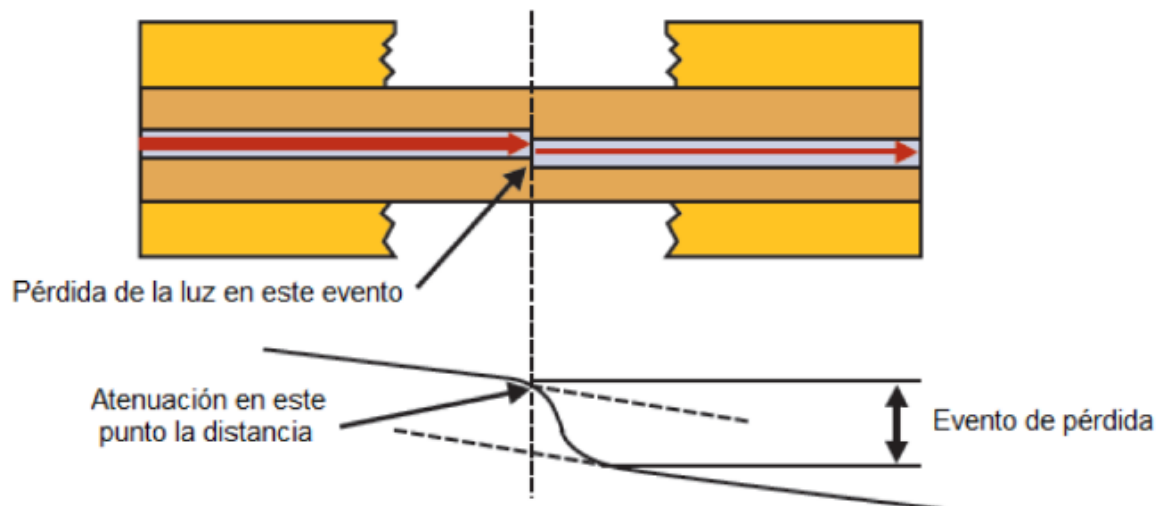


Figura 2.42. Evento de no reflexión. [29]

Para los eventos no reflectantes, la pérdida de evento puede aparecer como un aumento de eventos, que muestra un paso más en el trazado del OTDR. [29]

Mediciones en el OTDR

Un OTDR puede realizar las siguientes mediciones:

- a.- Para cada evento: distancia de ubicación, la pérdida y reflectancia. [30]
- b.- Para cada sección de la fibra se considera: longitud de sección, la pérdida de sección (en dB), la tasa de pérdida de sección (en dB/Km), y la pérdida de retorno óptico (ORL) de la sección. [30]
- c.- Para el sistema completo terminado: longitud del enlace, pérdida de enlace total (en dB) y ORL del enlace. [30]

Métodos de medición

El OTDR permite a los técnicos realizar mediciones en el tramo de fibra de varias maneras: funciones de medición automático (completo), semiautomático y manuales. Los técnicos también pueden usar una combinación de estos métodos. [30]

a.- Función de medición full automática

Usando la función totalmente automática, el OTDR detecta y mide todos los eventos, secciones y final de la fibra automática, utilizando un algoritmo de detección interna. [30]

b.- Función de medición semiautomática

La selección de la función semiautomática, las medidas del OTDR comunica un evento en cada ubicación (distancia) con un marcador. Los marcadores pueden ser colocados ya sea de forma automática o manualmente. La función semiautomática, es de gran interés durante el período de aceptación (después del empalme), los técnicos deben caracterizar completamente todos los eventos a lo largo del período con el fin de establecer datos de referencia. Dado que la detección automática no detecta e informa un evento no reflexivo con pérdida cero, se coloca un marcador en esa posición por lo que el análisis semiautomático informará la pérdida cero. [30]

c.- Función de medición manual

Para el análisis aún más detallado o por condiciones especiales, los técnicos controlan por completo la función de medición de forma manual. En este caso, los técnicos deben colocar dos o más cursores en la fibra con el fin de controlar la forma en que el OTDR mide el evento. [30]

2.2.14 Software de proceso matemático

Es aquel software que se utiliza para realizar, apoyar o ilustrar problemas matemáticos; este tipo de aplicaciones son un sub-campo de la computación científica, por ello también se le conoce como sistemas de cálculo numérico (SCN). Existen grupos y proyectos dedicados al estudio y difusión de software matemático libre, los cuales han aportado productos que facilitan el trabajo con estas herramientas:

Sistemas de Cálculo Numérico (SCN):

Un SCN contiene un motor o núcleo de análisis numérico que se usa para simular procesos matemáticos complejos aplicados a procesos del mundo real; entre las herramientas informáticas más conocidas se destacan. La tabla 2.2 muestra un cuadro

comparativo entre sistemas de cálculo numérico con un enfoque en el desarrollador, la licencia, sistema operativo y la interfaz. [31]

Tabla 2.2. Cuadro comparativo entre sistemas de cálculo numérico

SOFTWARE	DESARROLLADOR	LICENCIA	SISTEMA OPERATIVO	INTERFAZ
MATLAB	Mathworks	Software propietario	Linux, Unix, Windows, Mac	Interfaz gráfica de usuario
MAPLE	Maplesoft	Software propietario	Windows, Linux, Unix	Interfaz gráfica de usuario
CSPRO	SERPRO S.A.	Dominio público	Windows	-
X-12-ARIMA	Oficina del Censo de los Estados Unidos	Dominio público	Windows, Linux, Unix	Interfaz gráfica de usuario
DATAPLOT	Instituto Nacional de Estándares y Tecnología	Dominio público	Windows, Linux, Unix	Línea de comandos
AD MODEL BUILDER	ADMB Project	Licencia BSD	Windows, Linux	-
GRETL	The Gretl Team	GNU General Public License	Linux, Unix, Windows, Mac	Interfaz gráfica de usuario
FREEMAT	Samit Basu	GNU General Public License	Unix, Linux, Windows, Mac	Interfaz gráfica de usuario
GNU OCTAVE	John W. Eaton	GNU General Public License	Unix, Linux, Windows, Mac	Interfaz gráfica de usuario
SCILAB	Scilab Consortium	Código abierto	Windows, Linux, Unix	Interfaz gráfica de usuario
SECAN	Software educativo para la visualización de algunos métodos numéricos	GNU General Public License	Linux, Windows	-
LABVIEW	National Instruments	Software propietario	Unix, Linux, Windows, Mac	Interfaz gráfica de usuario

Fuente. La Investigación

En la lista de la tabla 2.2, los más utilizados en el mercado computacional son:

MATLAB:

Matlab (Matrix Laboratory) es un sistema interactivo, para su funcionamiento este software utiliza una matriz que no requiere de dimensionamiento. Fue especialmente diseñado para resolver problemas numéricos en fracciones de tiempo. Realiza cálculos matemáticos, analíticos y numéricos, permitiendo resolver sistemas de ecuaciones, matrices, representar funciones. La figura 2.43 muestra la interfaz de MATLAB. [32]

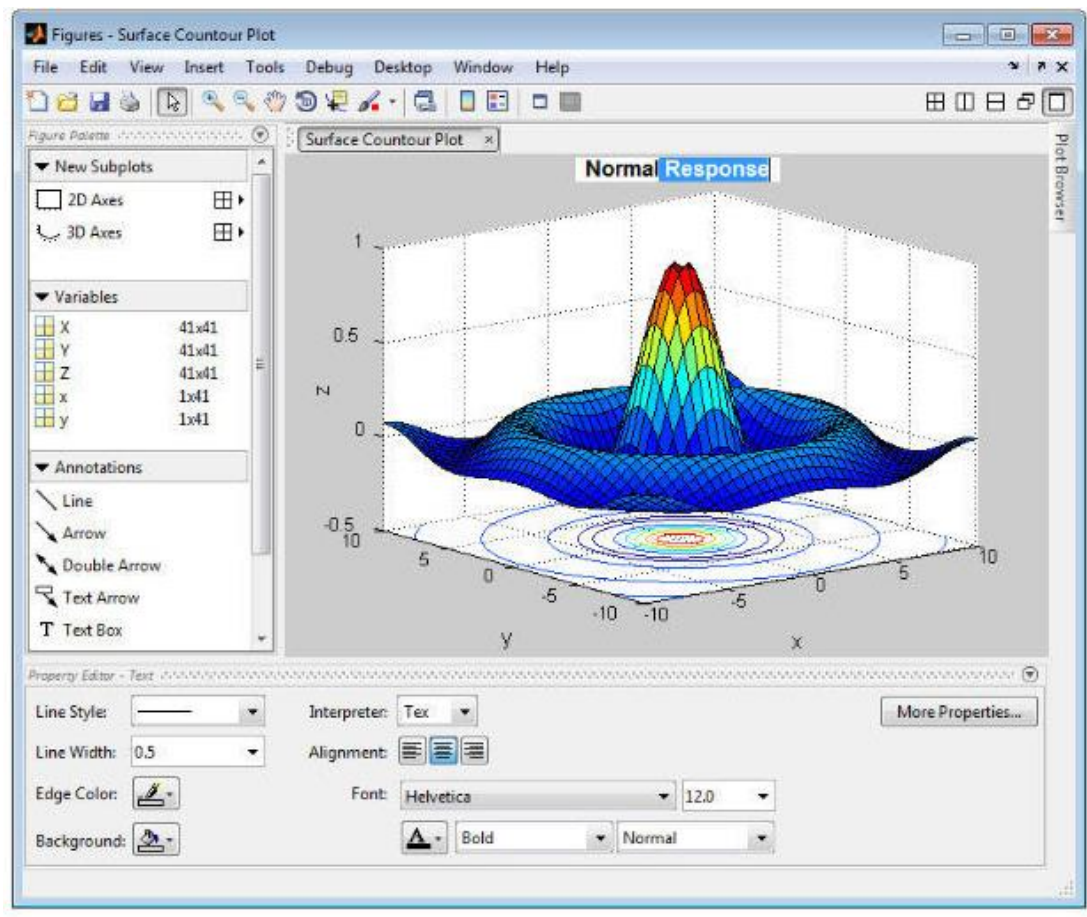


Figura 2.43. Interfaz de MATLAB. [32]

Descripción:

Cuenta con una serie de soluciones específicas denominadas TOOLBOXES, las cuales sirven para resolver clases particulares de problemas como:

- Identificación de sistemas y diseño de sistemas de control.
- Procesamiento de señales, redes neuronales y visión artificial.

Incluye también complementos como Simulink, con el cual se pueden hacer ensayos virtuales de sistemas, diagramas de Bode, Nyquist, etc. Se suele usar en ingenierías y carreras científicas. La figura 2.44 muestra la interfaz de SIMULINK. [32]

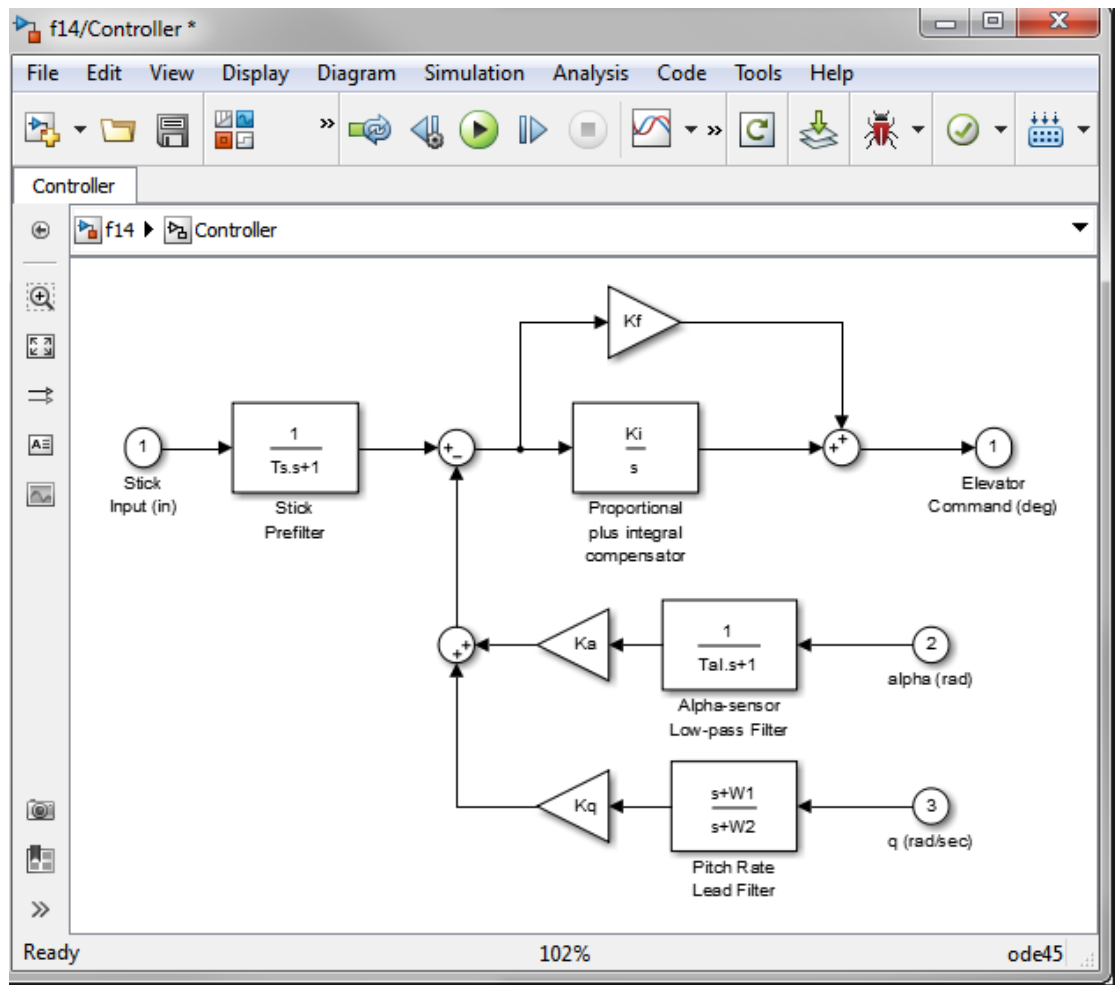


Figura 2.44. Interfaz de SIMULINK. [32]

Características:

- Software matemático incluyendo señales y procesamiento de imágenes.
- Se puede crear scripts que implementen algoritmos y funciones nuevas.
- Lenguaje de programación propio (lenguaje M). Los ficheros de script y funciones **.m** permiten ampliar y adaptar el lenguaje.
- No necesita implementar cálculo matricial, ya que la implementación de las matrices se las realiza como elemento del lenguaje, con lo cual se reduce notablemente el código

LABVIEW:

LabView (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de medición, adquisición de datos, procesamiento digital de señales, procesado matemático, entre otros. La figura 2.45 muestra la interfaz de LABVIEW. [33]

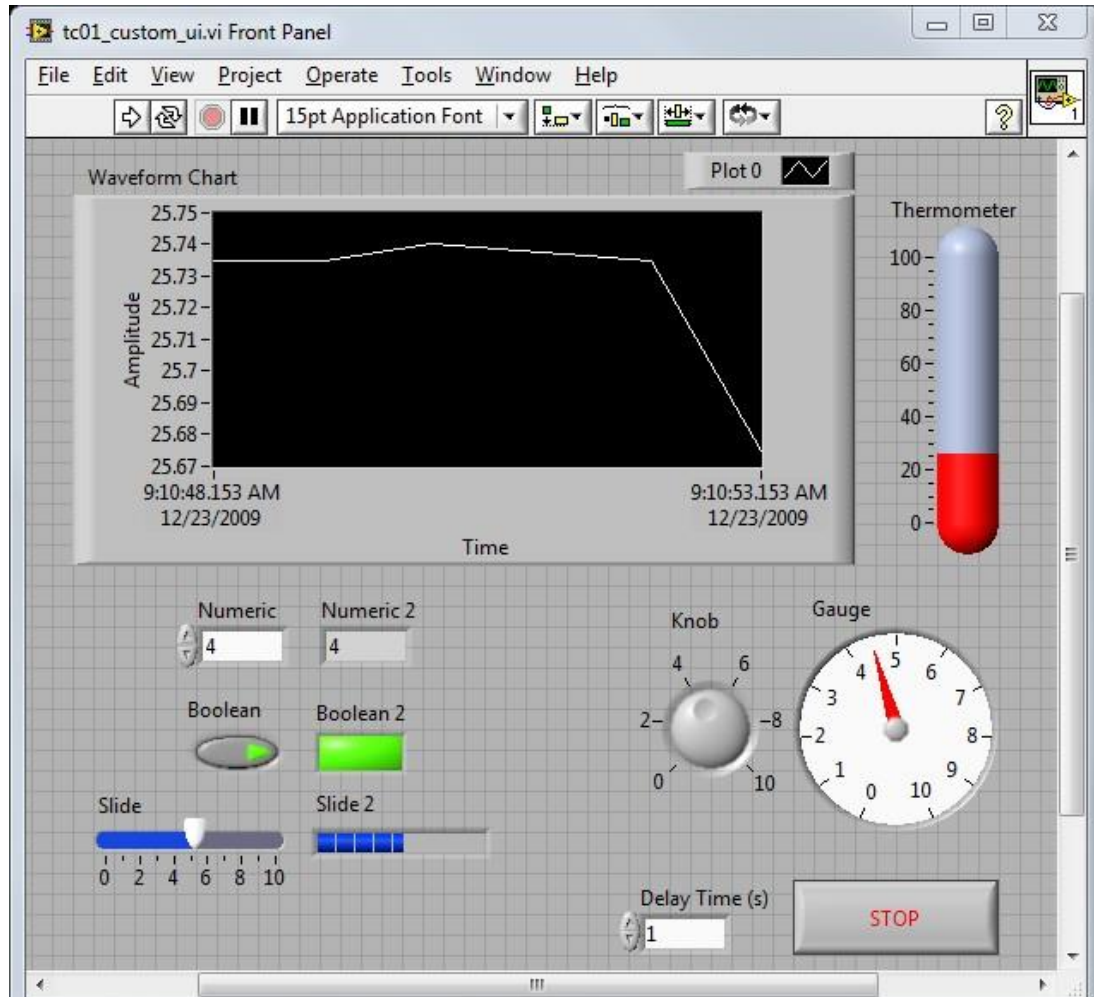


Figura 2.45. Interfaz de LABVIEW. [33]

Este programa permite una fácil integración con distintos tipos de hardware como tarjetas de medición, adquisición de imágenes y procesamiento de datos. [33]

Descripción:

Para la programación LABVIEW consta de dos paneles, un panel frontal y un panel de programación o también llamado diagrama de bloques. A su vez, es un entorno de

desarrollo integrado que se especializa en la informática industrial y científica. Lo que hace de LABVIEW un programa distinto a otros, es su entorno de programación totalmente gráfico llamado Lenguaje G. [33]

Características:

- Adquisición de datos, comunicación y control, automatización industria, visión artificial, robótica y redes.
- Los programas desarrollados con LABVIEW se llaman Instrumentos Virtuales, o Vis; ya que su aspecto y funcionamiento imita a instrumentos físicos.
- Facilidad de uso, válido para programadores profesionales y no profesionales.
- Se combina con todo tipo de software y hardware, tanto del propio fabricante como de otros fabricantes: tarjetas de adquisición de datos, visión, etc.

SCILAB:

Es un paquete de software científico Open Source para cálculo numérico, dentro de un entorno fácil de usar. Se puede programar algoritmos, representar funciones. La figura 2.46 muestra la interfaz de SCILAB. [34]

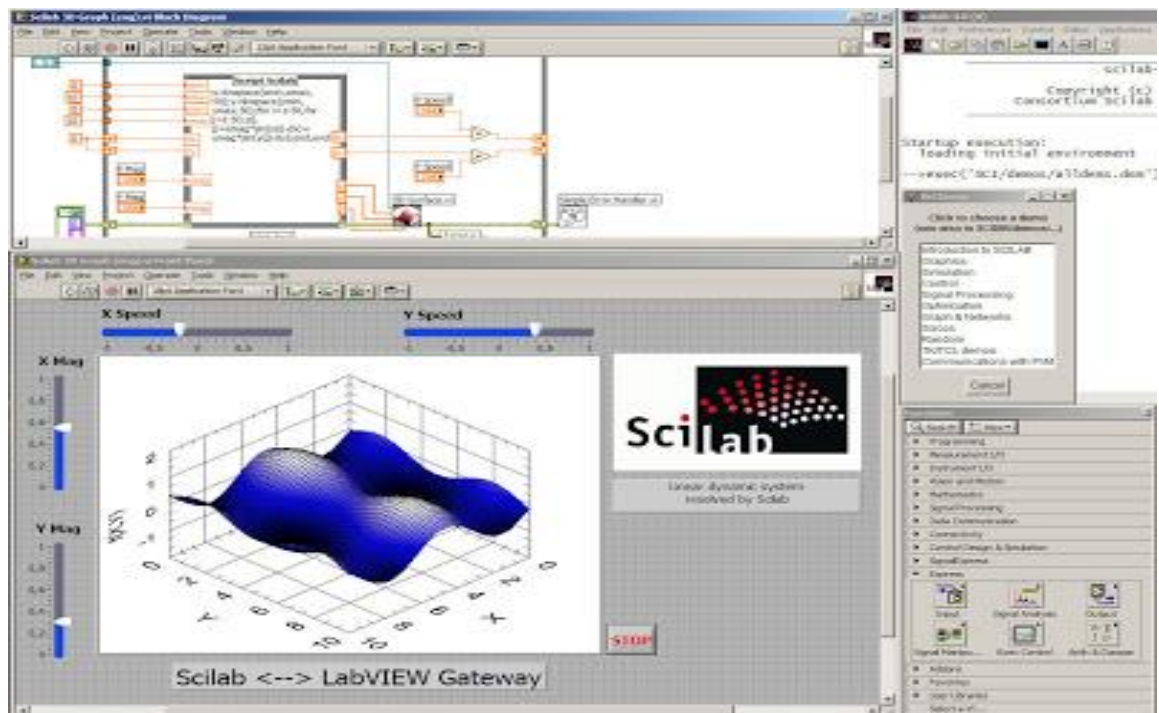


Figura 2.46. Interfaz de SCILAB. [34]

Descripción:

SCILAB incluye cientos de funciones matemáticas con la posibilidad de añadir programas desde diferentes lenguajes (C, Fortran...). [34]

Características:

- Puede incorporar nuevos tipos de datos y operaciones, incluye además una gran familia de toolboxes.
- Es un software gratuito de código abierto para programación numérica que proporciona un entorno para aplicaciones de ingeniería y científicas muy similar a MATLAB.
- SCILAB cubre una amplia gama de áreas, como la aeroespacial, industria energética, química, biología, medicina, finanzas, etc.
- Además, SCILAB contiene Xcos, un editor gráfico para el diseño de modelos de sistemas dinámicos híbridos, e incluso funcionalidades GUI, desde donde crear interfaces gráficas para el usuario.

GNU OCTAVE:

Es un lenguaje Open Source de alto nivel, orientado a la computación numérica. QtOctave es un front-end para GNU Octave, que sirve para simplificar el trabajo, implementando menús contextuales para las tareas más comunes, evitando así que se tengan que programar manualmente. La figura 2.47 muestra la interfaz de GNU Octave. [35]

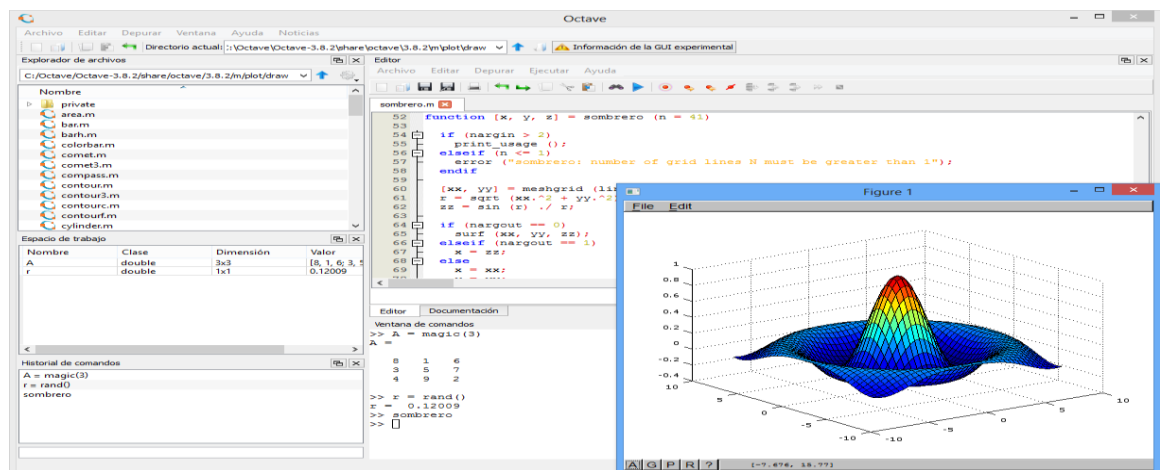


Figura 2.47. Interfaz de GNU Octave. [35]

Descripción:

Proporciona un entorno de comandos en línea adecuado para resolver problemas lineales y no lineales numéricamente, así como cualquier tipo de operación numérica. [35]

Características:

- Su tipo de datos fundamental es la matriz; integra compatibilidad con números complejos e incluye una potente y extensiva librería de funciones matemáticas.
- GNU Octave es un programa de libre distribución bajo los términos de la licencia GNU General Public License (GPL).
- Existen versiones para Unix-Linux y Window.
- Es usado normalmente a través de una línea de comandos interactiva.

MAPLE:

Resuelve sistemas de ecuaciones complicados mediante cálculos numéricos e iteraciones; realiza operaciones con matrices, programa algoritmos personalizados, dibuja gráficas y funciones de dos o más variables, resuelve ecuaciones diferenciales, etc. La figura 2.48 muestra la interfaz de Maple. [36]

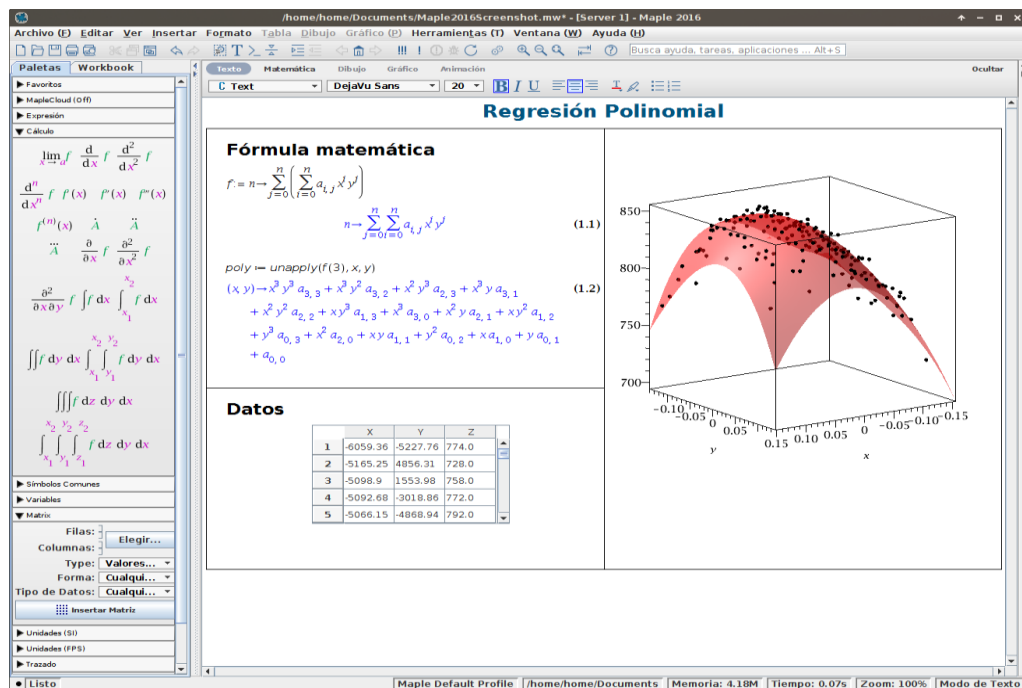


Figura 2.48. Interfaz de MAPLE. [36]

Descripción:

Orientado a la resolución de problemas matemáticos, capaz de realizar cálculos simbólicos, algebraicos y de álgebra computacional. [36]

Características:

- Maple se basa en un pequeño núcleo escrito en C.
- Las expresiones simbólicas son almacenadas en memoria como grafos dirigidos sin ciclos.
- La mayoría de funcionalidades son proporcionadas por librerías: unas escritas en lenguaje Maple, con acceso a su código fuente; pero también hace uso de otras librerías bien conocidas como las NAG, ATLAS o GMP.
- Tres tipos de licencia: Maple Professional, Maple Academic y Maple Student.
- Posee una librería de cálculo simbólico de Maple para integrarla en MATLAB.

2.3. PROPUESTA DE SOLUCIÓN

Se diseñó una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica que permite visualizar distintos tipos de enlaces de fibra óptica y la identificación de sus respectivos componentes; con lo cual se demuestra el funcionamiento de un OTDR tratando de caracterizar los principales problemas en las instalaciones con fibra óptica.

CAPÍTULO III

METODOLOGÍA

3.1. MODALIDAD DE LA INVESTIGACIÓN

La investigación también será de modo bibliográfica - documental para fundamentar el tema, estipular el marco teórico y constituir una estructura metodológica del proyecto de investigación; al mismo tiempo se utilizarán fuentes de información primaria y secundaria como repositorios digitales, papers, libros, revistas, etc.; de tal forma que se sustente el informe final del trabajo de graduación.

Se realizará la Investigación de Campo porque se tomará información de medidores existentes para realizar la comparación y el análisis correspondiente con el sistema propuesto.

3.2. POBLACIÓN Y MUESTRA

En el presente proyecto de investigación no se contó con una población y lógicamente con ninguna muestra, puesto que se realizó una investigación aplicada. Además, utilizó una metodología prototipo (prototipo), debido a que es un proyecto de investigación aplicado experimental.

3.3. RECOLECCIÓN DE INFORMACIÓN

En el presente proyecto se recolectó información por medio de documentos y el estudio de campo; además del levantamiento de información técnica, empleando fichas de

resumen de contenidos bibliográficos en conjunto a guías de observación como instrumentos para recopilar información según el avance de la investigación.

3.4. PROCESAMIENTO Y ANÁLISIS DE DATOS

La información formó parte de un procesamiento y análisis; con lo cual se obtuvo datos y resultados óptimos, efectivos y sistemáticos, mismos que se subrayan de la siguiente manera:

- Se analizó la información recopilada.
- Se depuró la información.
- Se procesó la información.
- Se optimizó algoritmos e información.
- Se diagramó los casos de uso.
- Se esquematizó y elaboró arquitecturas.
- Se diseñó e implementó la interfaz gráfica para simulación de un OTDR.
- Se realizó prototipos y pruebas piloto.
- Se evaluó e intervino errores.
- Se presentó resultados.

3.5. DESARROLLO DEL PROYECTO

En el desarrollo e implementación de una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica; se procederá de la siguiente manera:

- Analizar en investigaciones, tesis, papers y revistas acerca de la operatividad de la tecnología actual utilizada para la identificación de trazas reflectométricas en enlaces de fibra óptica.
- Realizar un análisis de las recomendaciones de la UIT-T para fibras ópticas en virtud al desarrollo del OTDR.
- Seleccionar un lenguaje de programación y herramienta matemática que permita crear una interfaz gráfica amigable al usuario.
- Elaborar la arquitectura de la interfaz gráfica de usuario (GUI) que simule un OTDR aplicado en sistemas de comunicaciones por fibra óptica.

- Construir algoritmos optimizados para cada proceso que integra un reflectómetro óptico en el dominio del tiempo.
- Validar prototipos en consecución a la metodología de diseño, basados en la adaptación de los algoritmos construidos.
- Integrar los módulos de los parámetros de simulación (diodo láser, longitud de onda, tipo de empalme, tipo de fibra óptica y tipo de conector), parámetros de funcionamiento (resonador, birrefringencia, EDFA) y parámetros métricos (atenuación del enlace, índice de refracción, atenuación de empalme, atenuación de conectores, reflectancia y pérdida de retorno óptico ORL).
- Realizar pruebas de la interfaz gráfica de usuario (GUI); verificar el funcionamiento y corregir errores.
- Elaborar la memoria técnica y documentación de la interfaz gráfica de usuario (GUI) que simule un OTDR aplicado en sistemas de comunicaciones por fibra óptica.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1. INTRODUCCION

El simulador que propone el proyecto de investigación brinda la posibilidad de estudiar diferentes parámetros dentro de las redes ópticas, posibilitando la variación de sus características técnicas reales en el mercado y poder descubrir su comportamiento para que la empresa SINFOTECNIA S.A pueda aplicar a sus planificaciones y gestionar su diseño, incluso puedan saber cuáles son los mejores parámetros aplicables a los dispositivos que actúan en su red; evitando gastos económicos y pérdidas de equipamiento.

El simulador induce a la empresa SINFOTECNIA S.A en la gestión de redes ópticas anticipando el diseño e implementación de las mismas, para descubrir cómo se comportaría una instalación óptica que solicite un cliente para determinado lugar y con determinado equipamiento; el aspecto calidad de redes sobresale con el uso del simulador, dando a la empresa la facilidad de buscar que materiales y equipos conviene para saber cómo actuará la red óptica y así de antemano conocer gastos y utilidades planificando de mejor manera la implantación de un sistema de comunicación por fibra óptica eficaz y eficiente.

La empresa SINFOTECNIA S.A contará con una herramienta informática de escritorio que le proporcione el “camino correcto” para la implementación de sus redes ópticas conociendo anticipadamente la estructura técnica de las mismas, así como su

actuación frente a tráfico real de información. Implementar una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica, es la mejor solución tecnológica a las exigencias internacionales de calidad e innovación a la planificación y gestión de redes ópticas, además; a la necesidad de ingeniería que solicitan las nuevas invenciones de software cuyo resultado potencian la ciencia y el conocimiento.

4.2. ANÁLISIS DE FACTIBILIDAD

Los siguientes numerales del proyecto de investigación, representan la factibilidad del desarrollo de la propuesta.

4.2.1 Factibilidad Operativa

Una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica es operativamente factible, debido a que consolida el estudio teórico y práctico de la Ingeniería en Electrónica y Comunicaciones, las telecomunicaciones y en especial las comunicaciones ópticas; ciencia tecnológica de alta utilidad en el perfil profesional del estudiante de ingeniería. Una óptima gestión del diseño e implementación de sistemas de comunicaciones por fibra óptica promueven calidad del servicio evitando gastos económicos en materiales y equipamiento.

4.2.2 Factibilidad Temporal

La factibilidad temporal en la implementación una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica está contemplada con el tiempo de aprobación estipulado por el Honorable Concejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial; aproximadamente seis meses.

4.2.3 Factibilidad Técnica

El desarrollo del proyecto es técnicamente factible, debido a que existe en la nube soporte técnico y algoritmos preestablecido para el desarrollo de interfaces gráficas en

varios lenguajes de programación orientado a objetos, los cuales serán acoplados y modificados a los requerimientos del proyecto de investigación.

4.2.4 Factibilidad Económica

Económicamente el desarrollo del proyecto es factible, debido al patrocinio y financiamiento de la empresa SINFOTECNIA S.A; además, para el desarrollo de la interfaz gráfica se utilizará software versión estudiantil y por tanto no representa magnánimos gastos al investigador.

4.2.5 Factibilidad Bibliográfica

Bibliográficamente es factible ya que la información necesaria para el desarrollo y posterior implementación de la interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica; se puede encontrar en libros, revistas científicas, papers, tesis de pregrado y postgrado, investigaciones y publicaciones técnicas relacionadas al tema en el internet.

4.2.6 Proyección a Futuro

El proyecto de investigación propone un método de gestión y diseño de sistemas de comunicaciones por fibra óptica con la simulación de un OTDR a través de una interfaz gráfica mejorando esporádicamente el primer prototipo. Esto podría hacerse adjuntando al algoritmo, la emulación de redes GPON; una base de datos con diversos fotodetectores, fotoreceptores, fibras ópticas y conectores empleados en el mercado nacional e internacional.

4.3. FUNDAMENTACIÓN CIENTÍFICO-TÉCNICA

Los numerales siguientes del proyecto de investigación representan los contextos de ingeniería, argumentos técnicos útiles para la implementación de una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica.

4.3.1 Recomendaciones de la UIT-T para fibras ópticas

La Unión Internacional de Telecomunicaciones (UIT) define una serie de recomendaciones donde establecen los tipos de fibras ópticas que presentan distintas

características según su estructura y propagación, estos estándares son proporcionados por la misma institución para el manejo público de esta información.

La información proporcionada por las recomendaciones de la UIT corresponde a las características y detalles de los tipos de fibras ópticas establecidos, dichos detalles comprenden propiedades físicas y de propagación.

A continuación, la tabla 4.1 presenta las recomendaciones consideradas en la elaboración del modelo de la GUI simulador SOTDR y las características relevantes para su realización.

Tabla 4.1. Resumen de estándares. [37], [38]

REC	Tipo de fibra	Revestimiento (μm)	Centro (μm)	Coeficiente de dispersión cromática		Dispersión por modo de polarización (ps/√Km)	Coeficiente de atenuación (dB/Km)	
				$\lambda_{0min} - \lambda_{0máx}$			Máximo en	
G.651.1	Multimodo	125	50	$\lambda_{0min} - \lambda_{0máx}$	1295-1340	-	Máximo en 850 nm	3.5
				$S_{0máx}$ para $1295 \leq \frac{\lambda_0}{nm} \leq 1310$	$\leq 0.105 ps/nm^2 \cdot Km$		Máximo en 1300 nm	1.0
				$S_{0máx}$ para $1310 \leq \frac{\lambda_0}{nm} \leq 1340$	$\leq 375 \cdot (1590 - \lambda_0) \cdot 10^{-6} ps/nm^2 \cdot Km$			
G.652 A	Monomodo	125	0.6	$\lambda_{0min} - \lambda_{0máx}$	1300-1324 nm	0.5	Máximo en 1310 nm	0.5
				$S_{0máx}$	$0.092 ps/nm^2 Km$		Máximo en 1550 nm	0.4
G.652 B	Monomodo	125	0.6	$\lambda_{0min} - \lambda_{0máx}$	1300-1324 nm	0.20	Máximo en 1310 nm	0.4
				$S_{0máx}$	$0.092 ps/nm^2 Km$		Máximo en 1550 nm	0.35
							Máximo en 1625 nm	0.4
G.652 C	Monomodo	125	0.6	$\lambda_{0min} - \lambda_{0máx}$	1300-1324 nm	0.5	Máximo de 1310 nm a 1625 nm	0.4
				$S_{0máx}$	$0.092 ps/nm^2 Km$		Máximo en 1383 nm	0.4
							Máximo en 1550 nm	0.3
G.652 D	Monomodo	125	0.6	$\lambda_{0min} - \lambda_{0máx}$	1300-1324 nm	0.2	Máximo de 1310 nm a 1625 nm	0.4
				$S_{0máx}$	$0.092 ps/nm^2 Km$		Máximo en 1383 nm	0.4
							Máximo en 1550 nm	0.3
G.653 A	Monomodo	125	0.8	$\lambda_{0min} - \lambda_{0máx}$	1500 – 1600 nm	0.5	Máximo en 1550 nm	0.35
				$S_{0máx}$	$0.085 ps/nm^2 Km$			
G.653 B	Monomodo	125	0.6	$D_{min}(\lambda): 1460 - 1525 nm$	$0.085(\lambda - 1525) - 3.5$	0.2	Máximo en 1550 nm	0.35
				$D_{min}(\lambda): 1525 - 1625 nm$	$3.5/75(\lambda - 1600)$			
				$D_{máx}(\lambda): 1460 - 1525 nm$	$3.5/75(\lambda - 1500)$			
				$D_{máx}(\lambda): 1525 - 1625 nm$	$0.085(\lambda - 1575) + 3.5$			
				$D_{1550máx}$	$20 ps/nm Km$			

G.654 A	Monomodo	125	0.6	$S_{1550m\acute{a}x}$	$0.070 \text{ ps/nm}^2 \text{ Km}$	0.5	Máximo en 1550 nm	0.22
G.654 B	Monomodo	125	0.6	$D_{1550m\acute{a}x}$	22 ps/nm Km	0.20	Máximo en 1550 nm	0.22
				$S_{1550m\acute{a}x}$	$0.070 \text{ ps/nm}^2 \text{ Km}$			
G.654 C	Monomodo	125	0.8	$D_{1550m\acute{a}x}$	20 ps/nm Km	0.20	Máximo en 1550 nm	0.22
				$S_{1550m\acute{a}x}$	$0.070 \text{ ps/nm}^2 \text{ Km}$			
G.654 D	Monomodo	125	0.8	$D_{1550m\acute{a}x}$	23 ps/nm Km	0.20	Máximo en 1550 nm	0.20
				$S_{1550m\acute{a}x}$	$0.070 \text{ ps/nm}^2 \text{ Km}$			
G.655	Monomodo	125	0.8	$D_{1550m\acute{a}x}$	22 ps/nm Km	0.20	Máximo en 1550 nm	0.22
				$S_{1550m\acute{a}x}$	$0.070 \text{ ps/nm}^2 \text{ Km}$			
G.656	Monomodo	125	0.8	$D_{min}(\lambda): 1460 - 1525 \text{ nm}$	$2.60/90(\lambda - 1460) + 1$	0.20	Máximo en 1460nm	0.4
				$D_{min}(\lambda): 1525 - 1625 \text{ nm}$	$0.98/75(\lambda - 1550) + 3.60$		Máximo en 1550 nm	0.35
				$D_{m\acute{a}x}(\lambda): 1460 - 1525 \text{ nm}$	$4.68/90(\lambda - 1460) + 4.60$		Máximo en 1625 nm	0.4
				$D_{m\acute{a}x}(\lambda): 1525 - 1625 \text{ nm}$	$4.72/75(\lambda - 1550) + 9.28$			
G.657	Monomodo	125	0.5	$\lambda_{0min} - \lambda_{0m\acute{a}x}$	1300 - 1324 nm	0.20	Máximo de 1310 nm a 1625	0.40
				$S_{0m\acute{a}x}$	$0.092 \text{ ps/nm}^2 \text{ Km}$		Máximo en 1383	0.40
							Máximo en 1550	0.30

Fuente: La Investigación

Según la ITU-T, el coeficiente de dispersión cromática $D(\lambda)$, G.651.1, se encuentra entre los límites de los parámetros de la curva de dispersión, la cual se encuentra en función a las longitudes de onda especificados en cada tipo de fibra clasificadas por estándares. Es calculado a partir de valores ideales, en los cuales la dispersión es nula en las longitudes de onda máxima, $\lambda_{0m\acute{a}x}$ y mínima, λ_{0min} , a considerar, y del coeficiente máximo de la pendiente, $S_{0m\acute{a}x}$, como se aprecia a continuación:

$$\frac{\lambda S_{0m\acute{a}x}}{4} \left[1 - \left(\frac{\lambda_{0m\acute{a}x}}{\lambda} \right)^4 \right] \leq D(\lambda) \leq \lambda \frac{S_{0m\acute{a}x}}{4} \left[1 - \left(\frac{\lambda_{0min}}{\lambda} \right)^4 \right] \quad (54)$$

4.3.2 Nomenclatura

Para la implementación de una interfaz gráfica para simulación de un OTDR aplicado en sistemas de comunicaciones por fibra óptica, se optó por comprimir el tema al acrónimo **S.O.T.D.R.** (Simulator Optical Time Domain Reflectometer – Simulador de Reflectómetro Óptico en el Dominio del Tiempo).

4.3.3 Selección de Software para el SOTDR

Para simular procesos matemáticos complejos aplicados a procesos del mundo real existen diferentes herramientas informáticas ya sea de código abierto (BSD) o código privado. Luego de hacer el análisis de algunos Sistemas de Cálculo Numérico, entre ellos, Matlab, Labview, SCILAB etc. El proyecto se desarrolló en el entorno de Matlab se optó por este lenguaje por sus problemas numéricos en fracciones de tiempo, ya que realiza cálculos matemáticos, analíticos y numéricos, permitiendo resolver sistemas de ecuaciones, matrices, representar funciones el simulador

Requerimiento para el SOTDR

Para que el programa pueda funcionar correctamente en un ordenador se necesita las siguientes características mínimas:

- Windows 7 o superior.
- Intel Core i5 2,6 Ghz o superior.
- Memoria RAM, 1GB o superior.
- Matlab 2014b

4.3.4 Requerimientos de SOTDR

Realizado el estudio teórico de los fundamentos y principales elementos de los sistemas de comunicación por fibra óptica, se procede al desarrollo del simulador realizado en MATLAB del OTDR. Encargándose éste de proporcionar la variación de la potencia óptica en función de la distancia y de los distintos elementos que componen el enlace de fibra óptica (conectores, empalmes, cable de fibra óptica, transmisor y receptor). El desarrollo del SOTDR se realiza íntegramente en interfaces .fig y funciones .m. A continuación, se muestra un diagrama de bloques del mismo, pero se debe tener en cuenta que el valor de cada bloque depende única y exclusivamente de los valores seleccionados por el usuario al ejecutar la interfaz gráfica GUI.

En la simulación realizada sólo se tiene en cuenta el tramo de ida en cuanto al balance de potencias, es decir, no es un OTDR real. Para que lo fuera se debería multiplicar todas las variaciones de la potencia por dos, considerando el tramo de ida y vuelta. En la figura 4.1 se muestra el esquema del OTDR implementado en MATLAB.

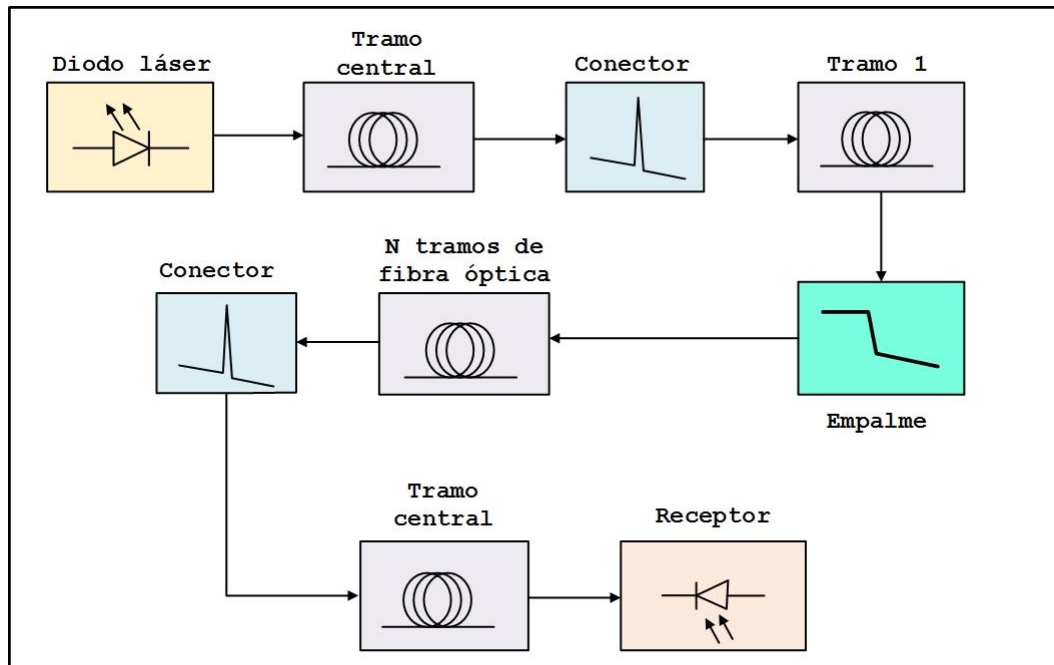


Figura 4.1. Esquema del OTDR implementado en MATLAB
Fuente: La Investigación

Como se puede apreciar en la figura anterior se inyecta la señal al enlace de fibra óptica por medio de un diodo láser o por un diodo LED a voluntad del usuario del simulador OTDR.

Cada bloque del sistema hace referencia a las características, tales como el tipo de fibra o empalme que depende de la elección del usuario y que se describirán más adelante. También se puede apreciar, que tanto para comenzar como para finalizar el enlace de fibra óptica, se utiliza un conector de fibra óptica a disposición del usuario y el simulador. Los tramos de fibra óptica se distribuirán, a elección del usuario que será el encargado de elegir tanto la longitud del enlace como la longitud del carrete (longitud de fábrica).

Con el fin de que la simulación realizada se parezca lo más posible a un OTDR real se ha utilizado una herramienta ya diseñada disponible en Mathworks, se trata de dos cursores desplazables horizontalmente con el fin de medir las diferencias de distancia y potencia entre ambos, para comprobar por ejemplo la atenuación de un conector o empalme.

A continuación, se mostrará lo anterior expuesto con un sencillo ejemplo. Se definirá dos vectores, uno llamado distancia y otro llamado potencia, realizando un sencillo

tramo de fibra óptica con un pequeño tramo al principio, un empalme y un tramo más largo al final. Tras esto se dibujarán estos dos vectores, añadiendo por último el cursor. [37], [38]

Este cursor dibujará en pantalla dos líneas rojas desplazables. Cada línea roja tiene asociado una caja amarilla de texto en la que se indica el valor de distancia y potencia del punto en la que está situada. Además de esto, en la parte inferior de la gráfica se indicará la diferencia tanto en distancia como en potencia de los cursores. [37], [38]

A la función original, se le han realizado un par de modificaciones sencillas para adaptarse mejor a la GUI desarrollada. En lugar de poner “x” o “y”, se ha sustituido por “distancia” y “potencia”. A continuación, se muestra tanto el código desarrollado como varias gráficas que muestran la forma en la que trabajan los cursores. [37], [38]

```

distancia=[0 0.3 0.5 0.51 1 2 3 4 5 6];
potencia=[0 -0.06 -0.1 -0.35 -0.45 -0.65 -0.85 -1.05 -1.25 -1.45];
plot (distancia, potencia, 'LineWidth', 2);
xlabel('Distancia (Km)');
ylabel('Potencia (dBm)');
dualcursor

```

La figura 4.2 muestra el resultado del ejemplo con los cursores añadidos. [37]

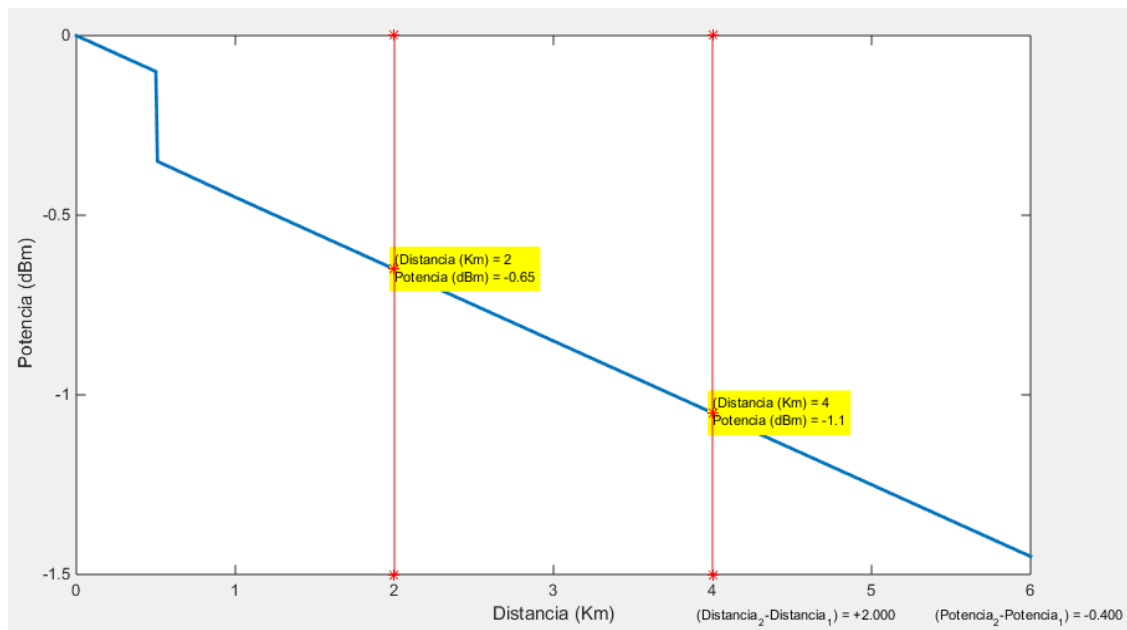


Figura 4.2. Resultado del ejemplo con los cursores añadidos. [37]

Fuente: La Investigación

Como se puede apreciar en la figura 4.2, se ha obtenido la representación de los dos vectores antes citados y aparecen los dos cursores con los que, a modo de ejemplo, se medirá la atenuación de la fibra en los tramos. Para una mejor visualización a continuación aparecen dos imágenes, una con los datos que aparecen en la parte posterior con la diferencia entre los dos cursores y otra con la zona entre los dos cursores. La figura 4.3 muestra el resultado de la medida de la atenuación en un Km. [37], [38]

$$(Distancia_{cursor2} - Distancia_{cursor1}) = +200 \quad (Potencia_{cursor2} - Potencia_{cursor1}) = -0.400$$

Figura 4.3. Resultado de la medida de la atenuación en un Km. [37]

Fuente: La Investigación

La figura 4.4 muestra una ampliación de la zona a medir. [37], [38]

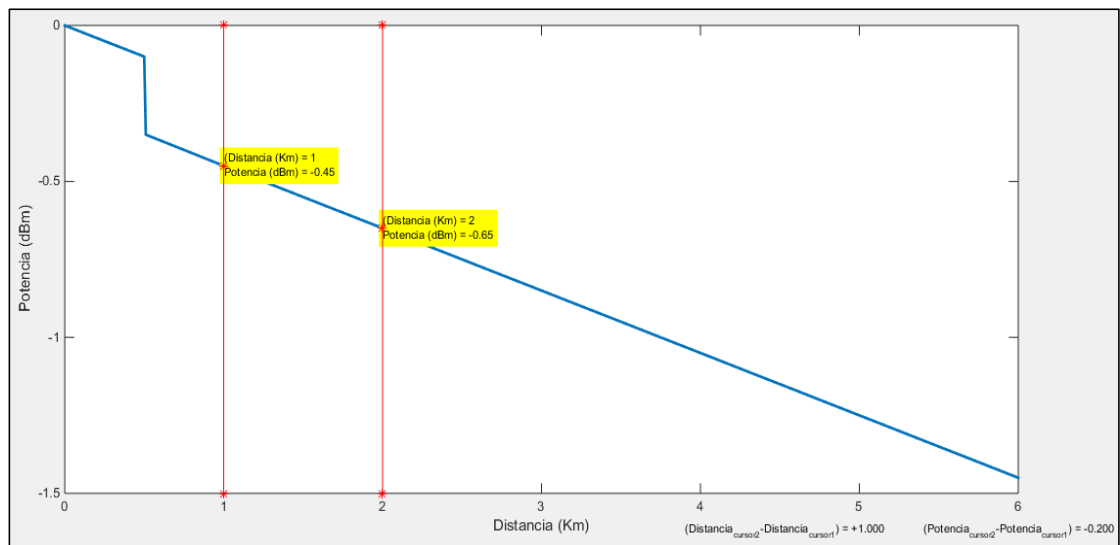


Figura 4.4. Ampliación de la zona a medir. [37]

Fuente: La Investigación

Como se observa en la primera imagen (o realizando un cálculo en la segunda), la atenuación de la fibra óptica en un Km es de 0.2 dB. [37], [38]

4.3.5 Diseño de la Interfaz Gráfica de Usuario (GUI) de SOTDR

El simulador SOTDR está desarrollado bajo la plataforma de MATLAB 2014b a través de su herramienta de Interfaz Gráfica de Usuario (GUI) de GUIDE, la cual, permite crear una comunicación interactiva y amigable para el usuario. Está enfocada en la simulación de las trazas reflectométricas semejantes al de un equipo OTDR, para

obtener valores de atenuación y potencia en enlaces de fibra óptica con parámetros reales a una instalación de comunicación, considerando también el fenómeno crítico en el uso de fibra óptica, la dispersión que ocurre en la transmisión de datos. La atenuación de la señal a lo largo del enlace es muy importante en el diseño de un enlace de fibra óptica, debido a que dependiendo de la cantidad de potencia que logre llegar al receptor óptico, será correctamente interpretada la señal en el receptor a la salida del enlace. Con la ayuda del simulador, el usuario puede escoger el receptor con la sensibilidad necesaria para poder observar la señal atenuada que llega. [39]

El otro parámetro que se ha tomado en cuenta es la dispersión, la cual permite dimensionar el ancho de banda máximo, así como también, la tasa de transferencia máxima de los datos que teóricamente el medio soporta. [39]

Diagrama general

En la figura 4.5 se muestra un diagrama de bloques que representa de manera general el funcionamiento del simulador, en este diagrama están representados los datos de entrada, que son los escogidos por el usuario para la simulación, los datos de salida, que son la respuesta del simulador y el bloque simulador que representa las operaciones que realiza MATLAB en base a los algoritmos de programación. Los datos de entrada son los datos del enlace que el usuario ingresa o escoge en la interfaz, dependiendo de estos datos será el resultado que se obtendrá al hacer la simulación. Los datos de salida son los datos que muestra la interfaz luego de analizar la información y simular el enlace óptico.

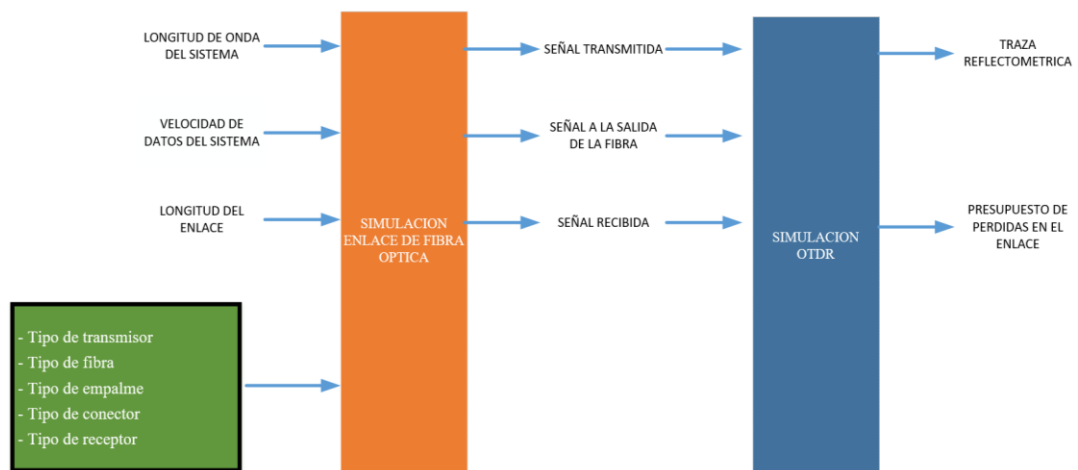


Figura 4.5. Diagrama de bloques simulador SOTDR
Fuente: Investigador

Los bloques simulación enlace de fibra óptica y simulación OTDR representan el modelo matemático, las ecuaciones y los algoritmos usados para emular un sistema de comunicación por fibra óptica y la capacidad principal de un equipo OTDR comercial, con sus trazas reflectométricas.

Diseño de los algoritmos del simulador SOTDR

La interfaz se ha diseñado de forma que el usuario tenga una GUI de ambiente amigable y de fácil manejo, sin la necesidad de tener conocimientos eruditos de los sistemas de comunicaciones por fibra óptica. El simulador SOTDR está desarrollado por medio de scripts de MATLAB con el enmascaramiento de la GUI, cuya interfaz interactúa con el usuario. Una vez configurada la información, la GUI ejecutada tiene como propósito ser una herramienta de escritorio que permita verificar el comportamiento de los enlaces de fibra óptica a través de una gráfica de señales reflectométricas, emulando en gran medida la capacidad de un equipo OTDR profesional. La figura 4.6 muestra la interacción entre las herramientas de MATLAB.

[39]

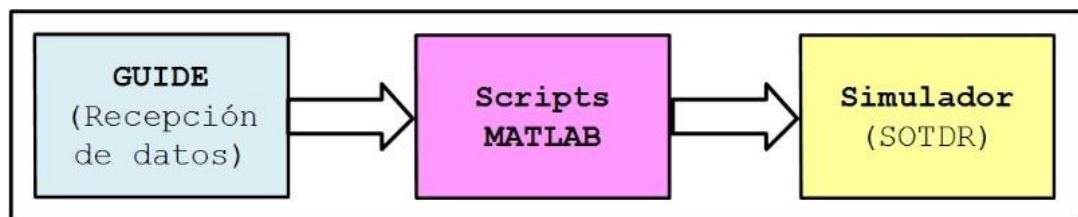


Figura 4.6. Interacción entre herramientas de MATLAB

Fuente. La Investigación

Algoritmos del simulador SOTDR

En este numeral del proyecto de investigación se explicará el algoritmo del simulador SOTDR. El script Enlace_Optico.m al igual que SOTDR.m son las funciones principales del proyecto, al igual que la GUI Enlace_Optico.fig y SOTDR.fig, reciben la información de las interfaces y ejecuta a través de la compilación de los algoritmos; los cálculos referentes al enlace de fibra óptica que esté configurado. El diagrama de flujo en la figura 4.7 detalla la secuencia de los procesos que se ejecutan en las interfaces principales.

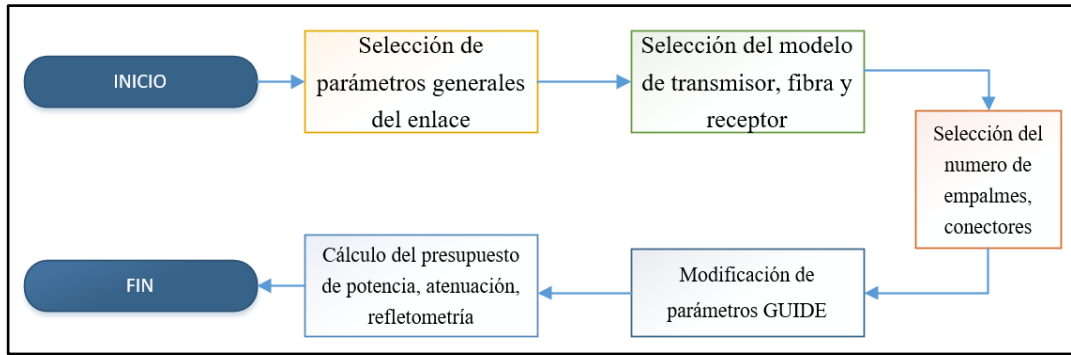


Figura 4.7. Algoritmo de la interfaz principal
Fuente: Investigador

Algoritmo para la selección de componentes

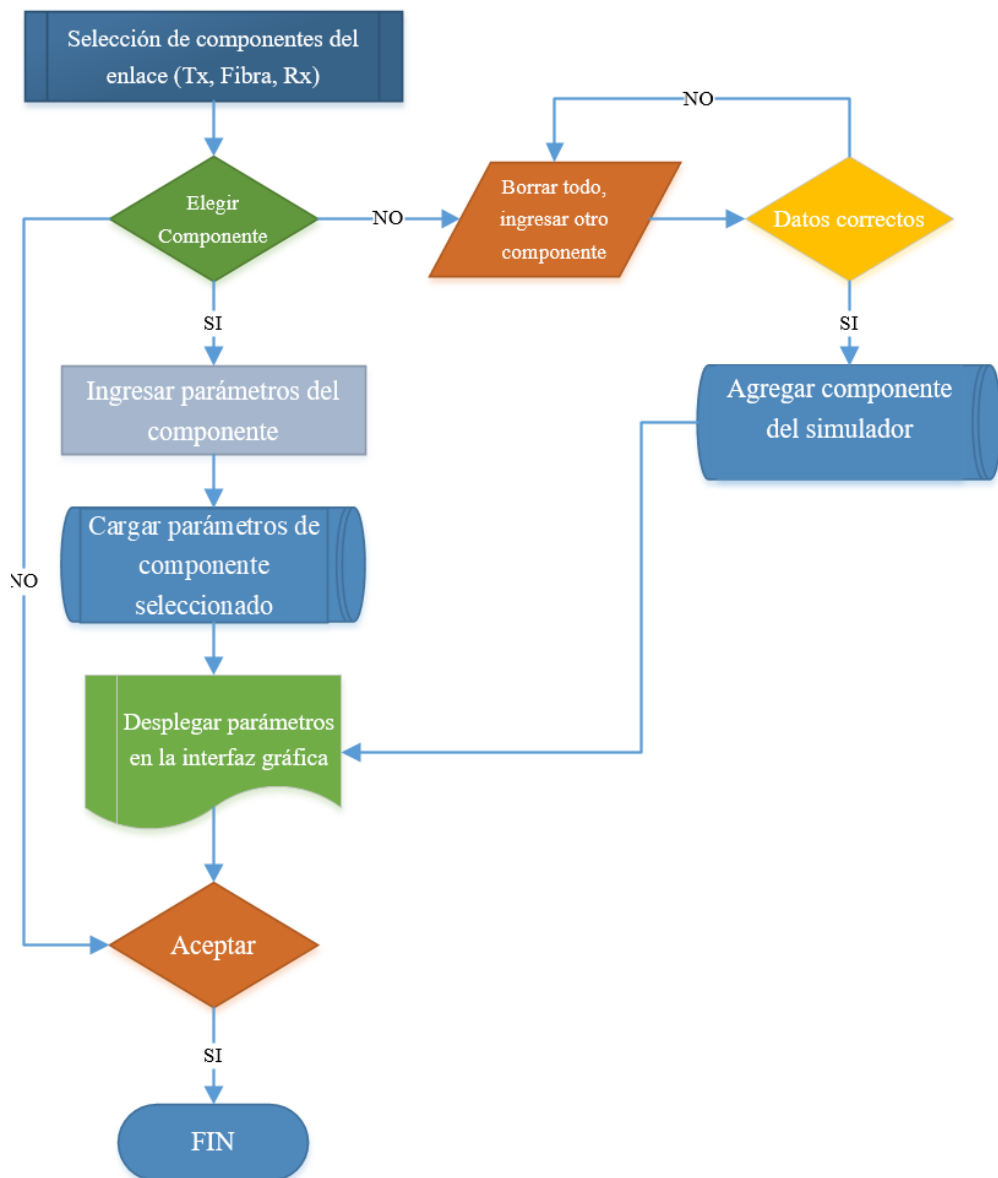


Figura 4.8. Algoritmo elección de componentes
Fuente: Investigador

La lógica que detalla la figura 4.8., es para ingresar parámetros de los componentes del enlace de fibra óptica existen tres scripts en .M de los .FIG del GUI: Transmisor.m, Fibra_Optica.m y Receptor.m; los cuales leen datos de las librerías, dentro de la misma carpeta del simulador SOTDR, estas son: Transmisor.fig, Fibra_Optica.fig y Receptor.fig; respectivamente.

Algoritmo para codificar la simulación en GUIDE

El siguiente diagrama de flujo explica la programación seguida por el simulador SOTDR en el proceso de realizar la simulación desde el script Enlace_Optico.m y SOTDR.m con los datos de los componentes y parámetros escogidos por el usuario. La figura 4.9 muestra el algoritmo para codificar la simulación en GUIDE.

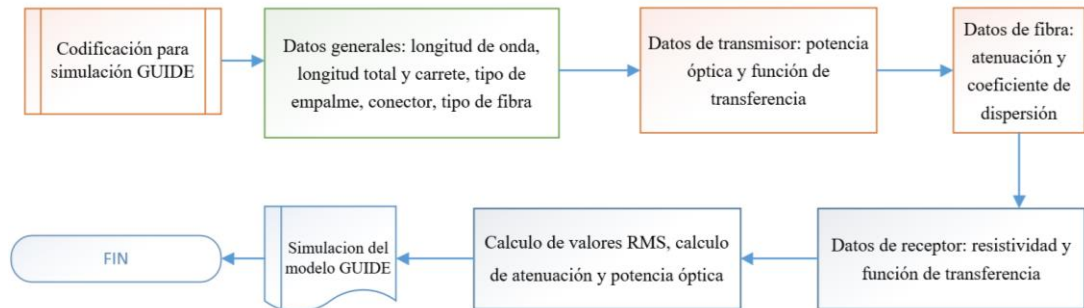


Figura 4.9. Algoritmo para codificar la simulación en GUIDE
Fuente: Investigador

Algoritmo del presupuesto de potencia y atenuación

El presupuesto de potencia es calculado en el script Presupuesto_Potencia.m y SOTDR.m, el cálculo se realiza en la etapa final del simulador. El script Enlace_Optico.m le concede la información de empalme y parámetros de componentes por medio de variables globales a los scripts antes mencionados. La principal función de este proceso, mostrado en la figura 4.10, es graficar el presupuesto de potencia en función de la longitud del enlace escogida e ilustrar al usuario si se cumplen los resultados de potencia y atenuación a través de una traza reflectométrica.

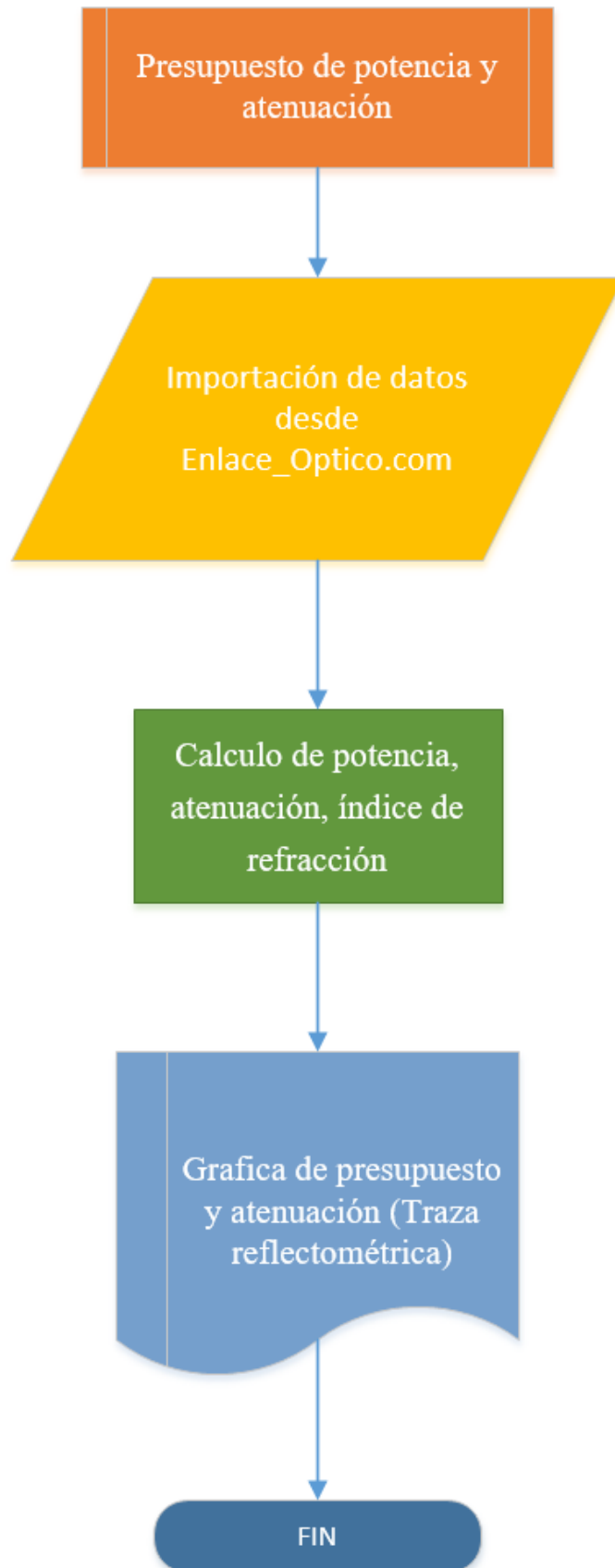


Figura 4.10. Algoritmo del presupuesto de potencia y atenuación
Fuente: Investigador

Interfaz gráfica para simulación del OTDR

El simulador SOTDR consta de ocho interfaces gráficas de usuario en primer orden se encuentra la interfaz PORTADA, que es como su nombre lo indica la portada del simulador SOTDR, en segundo orden se encuentra la interfaz ENLACE ÓPTICO, que es la interfaz donde se configuran los parámetros para la simulación del sistema de comunicación por fibra óptica; en tercer lugar están las interfaces TRANSMISOR, FIBRA ÓPTICA y RECEPTOR donde se configura cada componente del sistema de comunicación; en cuarto lugar se localiza la interfaz simulador SOTDR en donde se mostrará los resultados de la simulación del enlace óptico junto con los resultados de atenuación y potencia, además de la traza reflectométrica del enlace; en quinta ubicación está la interfaz PRESUPUESTO DE POTENCIA con una gráfica de la potencia del transmisor versus la potencia del receptor. Por último y no menos importante, se ubica la interfaz AYUDA, donde se expresa un resumen de la funcionalidad del simulador SOTDR, además del copyright del mismo. En la figura 4.11 se muestra la interfaz PORTADA del simulador SOTDR.



Figura 4.11. Interfaz PORTADA del simulador SOTDR

Fuente: Investigador

En la figura 4.12 se muestra la pantalla principal del simulador SOTDR, esta interfaz está dividida en tres procesos principales que deben cumplirse antes de realizar la simulación:

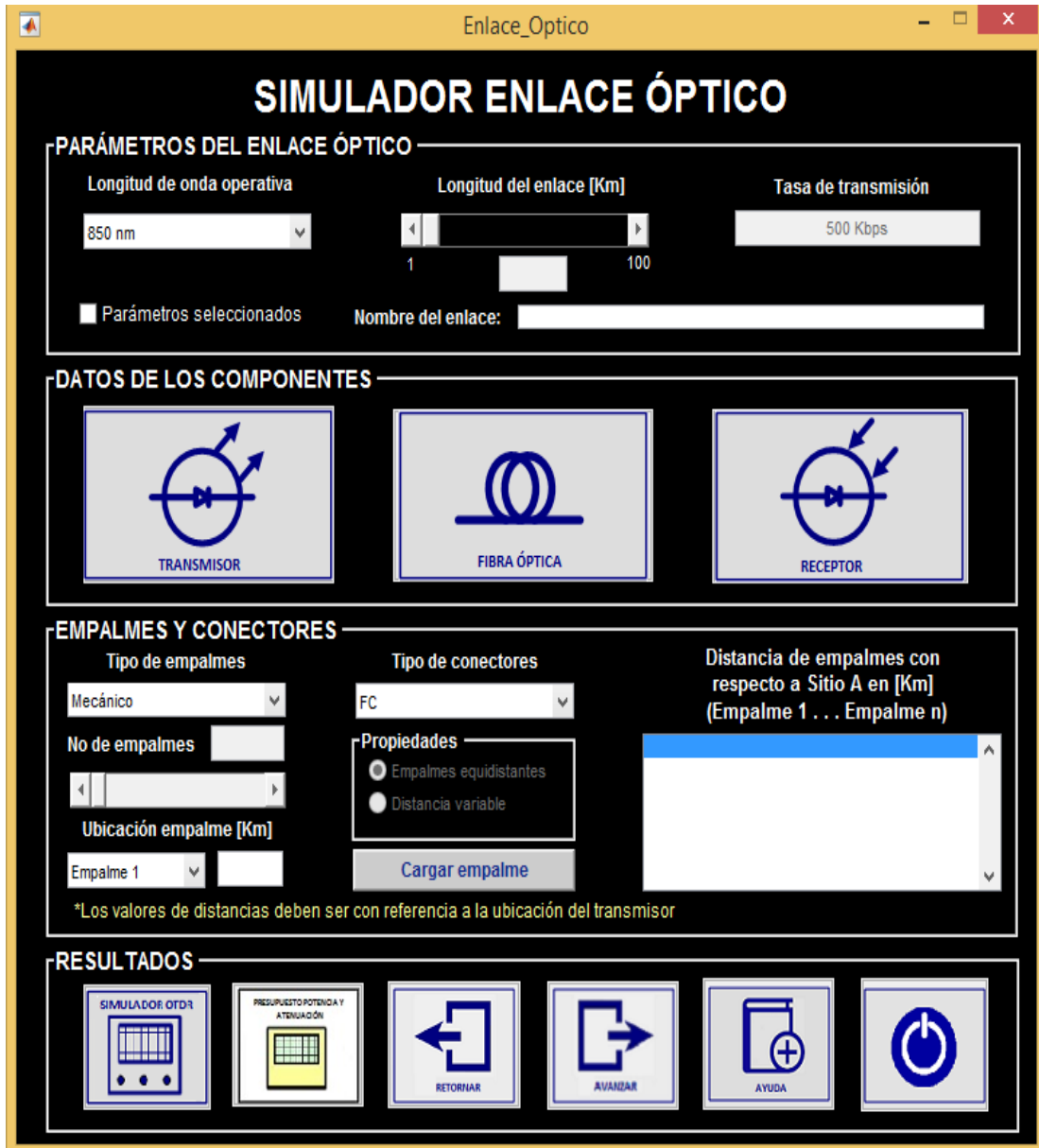


Figura 4.12. Interfaz ENLACE ÓPTICO del simulador SOTDR
Fuente: Investigador

- 1.- Parámetros del enlace de fibra óptica.
- 2.- Datos de los componentes ópticos.
- 3.- Empalmes y conectores del enlace de fibra óptica.

Estos tres procesos deben seguirse en el orden secuencial correcto antes de efectuar la respectiva simulación y obtener los datos presupuestados de potencia y atenuación para la visualización de la traza reflectométrica. Es importante seguir el orden de compilación de cada interfaz que agrupa la GUI Enlace_Optico para llegar al proceso

de simulación OTDR, a pesar de que el simulador manipula mensajes de error y advertencia para guiar todos los procesos.

En la figura 4.13 se muestra la interfaz SIMULADOR SOTDR.

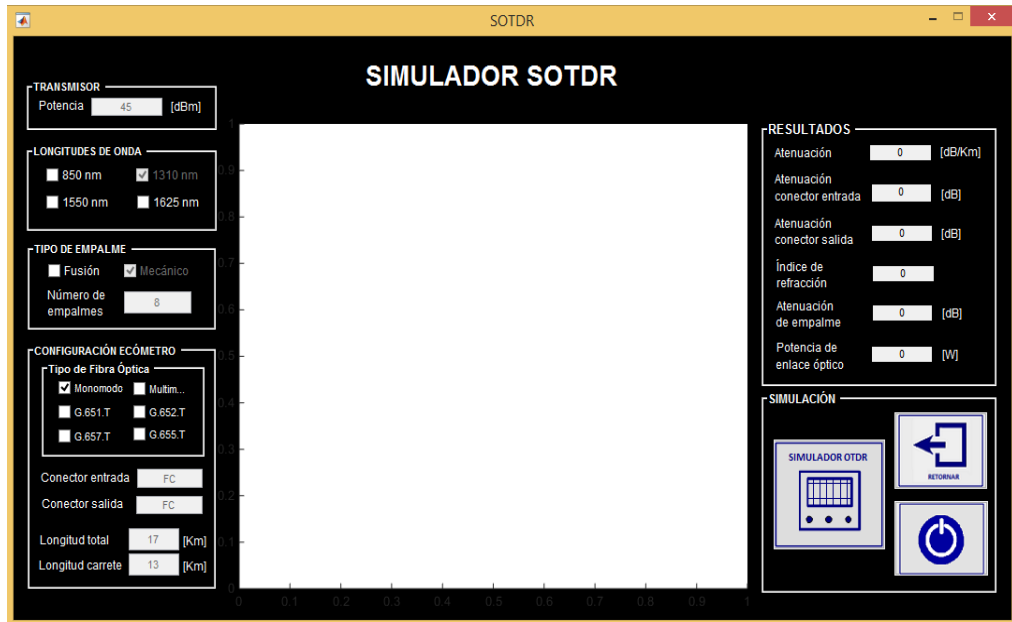


Figura 4.13. Interfaz SIMULADOR SOTDR
Fuente: Investigador

En la figura 4.14 se muestra la interfaz AYUDA del simulador SOTDR.



Figura 4.14. Interfaz AYUDA del simulador SOTDR
Fuente: Investigador

4.3.6 SOTDR

Para el desarrollo del simulador SOTDR se va a describir cada uno de los componentes o módulos implementados, partiendo en un orden procedimental, en virtud a la configuración de un enlace de fibra óptica.

Parámetros del enlace óptico

Los parámetros del enlace óptico están distribuidos en la longitud de onda operativa (nm), la longitud del enlace (Km) y la tasa de transmisión (Kbps o Mbps); así como el nombre del enlace; toda esta configuración es cargada al enlace por medio del checkbox Parámetros seleccionados. La figura 4.15 muestra los parámetros del enlace óptico.

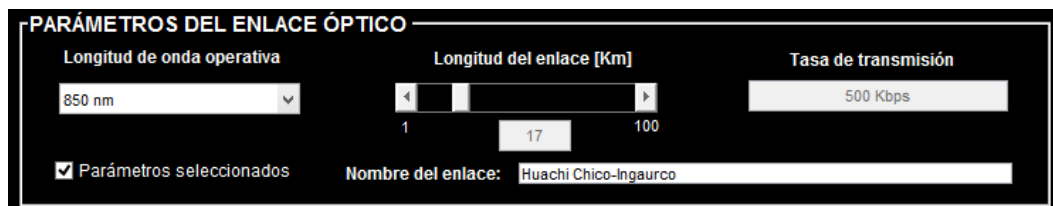


Figura 4.15. Parámetros del enlace óptico

Fuente: Investigador

En la configuración de la longitud de onda operativa, los valores de selección son 850 nm, 1310 nm, 1550 nm y 1625 nm. La longitud del enlace está distribuida en una distancia desde 1 Km hasta 100 Km; por último, la tasa de transmisión está comprendida en valores desde: 500 Kbps, 1 Mbps, 3Mbps, 5 Mbps, 51.84 Mbps, 155.52 Mbps, 622.08 Mbps, 1244.16 Mbps, 2588.32 Mbps, 9953.28 Mbps, 39814.32 Mbps.

Datos de los componentes

Los datos de los componentes se configuran en los botones Transmisor, Fibra Óptica y Receptor, de la GUI Enlace_Optico.fig hecha en MATLAB GUIDE. En la figura 4.16 se muestra los botones de configuración de los datos de componentes.

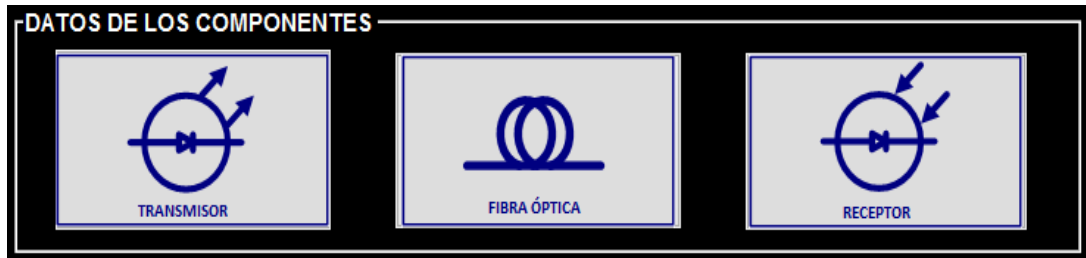


Figura 4.16. Datos de los componentes
Fuente: Investigador

Transmisor

Para la configuración de la GUI Transmisor.fig, es necesario tomar en cuenta los parámetros del sitio A, el sitio donde se encuentra el transmisor; el tipo de transmisor (Láser o Led), el fabricante o modelo del transmisor, el Bit Rate (Kbps o Mbps), la potencia del transmisor (dBm), la modulación (NRZ, RZ y Manchester), el tiempo de subida (ns) y el ancho espectral (nm). Toda la configuración del transmisor se la puede cargar al enlace o borrar con los respectivos botones de la interfaz. La figura 4.17 muestra la GUI del Transmisor.



Figura 4.17. GUI Transmisor
Fuente: Investigador

Fibra Óptica

La configuración de la GUI Fibra_Optica.fig está determinada por el tipo de fibra óptica (Monomodo y Multimodo), el estándar de fibra óptica; entre el cual destaca: UIT-T G.651, UIT-T G.652A, UIT-T G.652B, UIT-T G.652C, UIT-T G.652D, UIT-T G.653, UIT-T G.654, UIT-T G.655, UIT-T G.656, UIT-T G.657 y Sin estándar. La figura 4.18 muestra la GUI de la fibra óptica para su configuración.

DATOS DE LA FIBRA ÓPTICA		
Estándar de fibra óptica	Tipo de fibra óptica	Dispersión cromática material [ns/nm.Km]
UIT-T G.652B	Monomodo	
Atenuación [dB/Km]	Dispersión cromática [ns/nm.Km]	Dispersión cromática guía de onda [ns/nm.Km]
0.56	3.2	
PDM [ps/sqrt(Km)]	Ancho de banda [Mhz*Km]	Cargar
15	26	Borrar
Factor de acoplamiento entre modos	Longitud de fábrica de la fibra óptica [Km]	
	4.5	

Figura 4.18. GUI Fibra Óptica
Fuente: Investigador

Además de la configuración del tipo de fibra óptica y el estándar; existe la atenuación del transmisor (dB/Km), la dispersión cromática (ns/nm.Km), la dispersión cromática del material (ns/nm.Km), la dispersión cromática de la guía de onda (ns/nm.Km), el PDM (ps/sqrt.Km), el ancho de banda (Mhz.Km), el factor de acoplamiento entre modos y la longitud de fábrica de la fibra óptica (bobina en Km).

Receptor

Para la configuración de la GUI Receptor.fig es necesario introducir valores en el sitio B (ubicación del receptor), el tipo de receptor (PIN, diodo avalancha y fotodiodo), el fabricante o modelo del receptor, el margen de seguridad (dBm), la sensibilidad (dBm) y el tiempo de subida del receptor (ns). La figura 4.19 muestra la GUI del Receptor.



Figura 4.19. GUI Receptor
Fuente: Investigador

Empalmes y conectores

La configuración de los empalmes y conectores está enfocada como puntos principales en el tipo de empalme donde se puede seleccionar entre empalme mecánico o empalme de fusión; los tipos de conectores están disponibles a satisfacción del enlace en FC, LC, MT-Array, SC, SC-Duplex, ST y BFOC. La figura 4.20 muestra la sección empalmes y conectores de la GUI Enlace_Optico.fig.

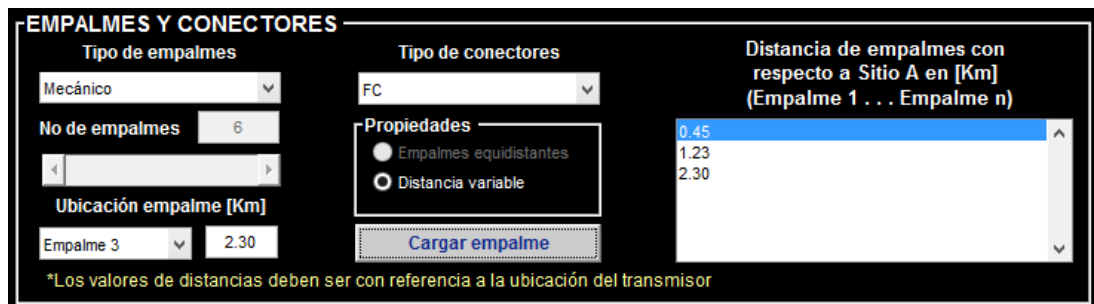


Figura 4.20. Empalmes y Conectores
Fuente: Investigador

Adicional al tipo de empalme y conector; las propiedades de los empalmes pueden establecerse en el número de empalmes, que va desde uno a diez empalmes debido a que si un enlace supera esta cantidad de empalmes ya sea por fusión o mecánico, la atenuación total del enlace es excesiva al nivel de impedir la factibilidad del mismo,

limitando la correcta propagación del haz de luz del láser o led empleado como transmisor.

También, se puede seleccionar la propiedad del empalme ya sea para empalmes equidistantes o empalmes que se encuentran a una distancia variable; una vez seleccionada la propiedad, se puede establecer la ubicación equidistante o variable para cada empalme.

El botón cargar empalme permite subir los datos configurados del empalme o empalmes al enlace óptico, al mismo tiempo que el tipo de conector.

Resultados

La sección de resultados de la GUI Enlace_Optico.fig está compuesta por seis botones entre los cuales destacan las propiedades de acceder a la GUI del simulador SOTDR, al GUI del presupuesto de potencia y atenuación y al GUI de Ayuda.

Además, existen los botones de retornar a la pantalla de portada, avanzar a la siguiente pantalla GUI en este caso SOTDR y lógicamente el botón de Salir del simulador. La figura 4.21 muestra la sección resultados de la GUI Enlace_Optico.fig.

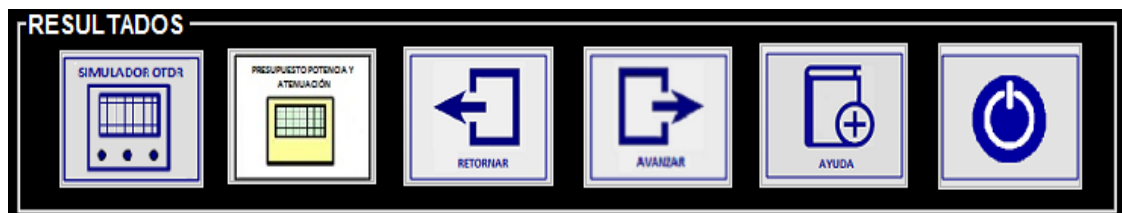


Figura 4.21. Resultados
Fuente: Investigador

Simulador OTDR

La figura 4.22 muestra el botón simulador OTDR con el que se puede acceder al GUI SOTDR.fig, el mismo que representa la funcionalidad técnica del proyecto de investigación.



Figura 4.22. Botón Simulador OTDR
Fuente: Investigador

La figura 4.23 muestra la GUI SOTDR, la misma que está constituida por los parámetros rescatados de las configuraciones en cada una de las respectivas GUI, transmisor, receptor, fibra óptica y enlace óptico.

Entre las secciones que contiene el GUI SOTDR está la potencia del transmisor (dBm), la longitud de onda (nm), el tipo de empalme (fusión o mecánico) con el número de empalmes, el tipo de fibra seleccionado con su respectivo estándar de la UIT-T, el conector de entrada y salida; además de la longitud total del enlace (Km), así como la longitud de fábrica de la fibra óptica en el carrete o bobina (Km).

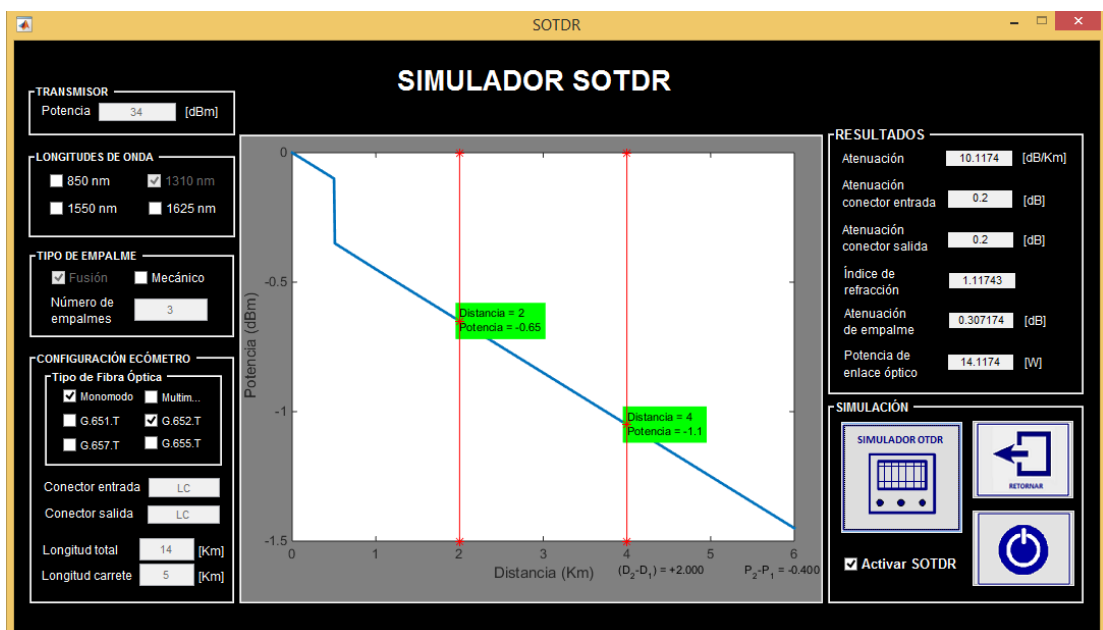


Figura 4.23. GUI SOTDR
Fuente: Investigador

En la parte lateral derecha la GUI SOTDR se establece la sección de resultados, los mismos que están distribuidos en atenuación total del enlace (dB/Km), la atenuación

del conector de entrada (dB), la atenuación del conector de salida (dB), el valor del índice de refracción, la atenuación del o los empalmes (dB) y la potencia del enlace óptico (W). Los valores de estos resultados están calculados dentro del algoritmo del Callback del botón simulador OTDR, el mismo que permite graficar la traza reflectométrica en un componente axes del GUIDE de MATLAB y al mismo tiempo plasmar los datos calculados en el editbox correspondiente de cada variable resultado. El botón retornar permite regresar a la GUI Enlace_Optico.fig y el botón Salir permite habilitar un cuadro de diálogo como muestra la figura 4.24.

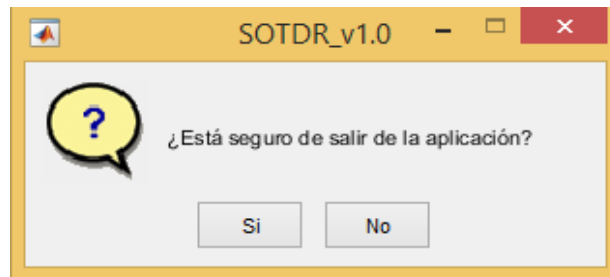


Figura 4.24. Cuadro de diálogo botón Salir
Fuente: Investigador

El cuadro de diálogo del botón salir permite al usuario establecer la decisión de si abandona el simulador SOTDR o se mantiene en el mismo para la configuración y emulación de nuevos enlaces de fibra óptica.

Presupuesto de Potencia y Atenuación

La figura 4.25 muestra el botón Presupuesto Potencia y Atenuación con el que se puede acceder al GUI Presupuesto_Potencia.fig, esta interfaz ayuda al usuario a visualizar de manera gráfica la caída en la potencia óptica a lo largo del enlace, como también líneas constantes que corresponden a la Potencia óptica de salida, la sensibilidad del receptor, el límite fijado por el margen de seguridad y la potencia óptica que se recibe en el receptor.



Figura 4.25. Botón Presupuesto Potencia y Atenuación
Fuente: Investigador

La figura 4.26 muestra la GUI Presupuesto_Potencia.fig, esta interfaz está compuesta como elementos principales de un axes, un listbox y un push button; el listbox contiene la ubicación de los respectivos empalmes configurados en la GUI Enlace_Optico ya sea por fusión o mecánico, el botón cargar y regresar habilitan los valores recibidos de la GUI antecesora y el axes muestra de forma gráfica los resultados de potencia, sensibilidad y atenuación.

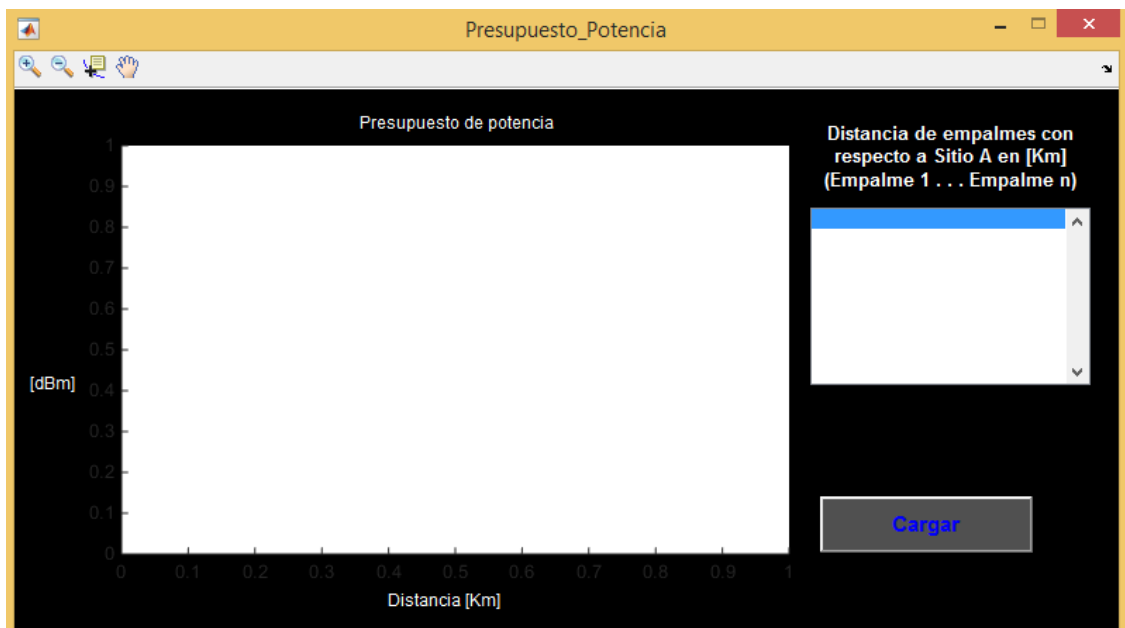


Figura 4.26. GUI Presupuesto_Potencia
Fuente: Investigador

4.3.7 Configuración simulador SOTDR

La respectiva configuración establece los parámetros de cada componente para la emulación de un enlace óptico y su respectiva traza reflectométrica.

Configuración de longitud de onda operativa

La longitud de onda operativa para fibra monomodo y multimodo según el estándar UIT-T, está comprendida en los siguientes valores mostrados en la tabla 4.2.

Tabla 4.2. Longitud de onda operativa para fibra monomodo y multimodo.

Longitud de onda (nm)	Estándar (UIT-T)	Fibra óptica operativa (UIT-T)
850	UIT-T G.651	Multimodo
1310	UIT-T G.652A UIT-T G.652B UIT-T G.652C UIT-T G.652D UIT-T G.657	Monomodo Monomodo Monomodo Monomodo Monomodo
1550	UIT-T G.652A UIT-T G.652B UIT-T G.652C UIT-T G.652D UIT-T G.653 UIT-T G.654 UIT-T G.655 UIT-T G.656 UIT-T G.657	Monomodo Monomodo Monomodo Monomodo Monomodo Monomodo Monomodo Monomodo Monomodo
1625	UIT-T G.652B UIT-T G.652C UIT-T G.652D UIT-T G.656 UIT-T G.657	Monomodo Monomodo Monomodo Monomodo Monomodo

Fuente. La Investigación

Configuración de la fibra óptica

Para la configuración de la fibra óptica se fijaron los siguientes parámetros para los cálculos de presupuesto de potencia, ancho de banda y velocidad de transmisión:

- Estándar de fibra.

- Atenuación de la fibra (dB/Km)
- Dispersión por modo de polarización o PDM (ps/ $\sqrt{\text{Km}}$)
- Factor de acoplamiento entre modos
- Tipo de fibra
- Ancho de banda (MHz)
- Dispersión cromática (ns/nm*Km)
- Dispersión cromática del material (monomodo) (ns/nm*Km)
- Dispersión cromática de la guía de onda (monomodo) (ns/nm*Km)
- Longitud de fábrica de la fibra (Km)

La tabla 4.3 muestra el coeficiente de atenuación de la fibra óptica según los estándares de la UIT-T.

Tabla 4.3. Coeficiente de atenuación de la fibra óptica.

Estándar UIT-T	Tipo de fibra óptica	Longitud de onda operativa (nm)	Coefficiente de atenuación (dB/Km)
UIT-T G.651	Multimodo	850	3.5
UIT-T G.652A	Monomodo	1310	0.5
		1550	0.4
UIT-T G.652B	Monomodo	1310	0.4
		1550	0.35
		1625	0.4
UIT-T G.652C	Monomodo	1310	0.4
		1550	0.3
		1625	0.4
UIT-T G.652D	Monomodo	1310	0.4
		1550	0.3
		1625	0.4
UIT-T G.653	Monomodo	1550	0.35
UIT-T G.654	Monomodo	1550	0.215
UIT-T G.655	Monomodo	1550	0.22
UIT-T G.656	Monomodo	1550	0.35
		1625	0.4
UIT-T G.657	Monomodo	1310	0.4
		1550	0.3
		1625	0.4
Sin Estándar	Multimodo	850	3.5
	Monomodo	1310	0.42
		1550	0.31
		1625	0.4

Fuente. La Investigación

Configuración del empalme

El valor del empalme ya sea por fusión o mecánico, tanto en los sistemas reales como en este simulador, no siempre es fijo por lo que se le añade a su valor común un cierto componente aleatorio que le hace variar de una simulación a otra.

En caso del empalme por fusión: $\text{atenuacion_empalme}=0.1 + 0.05 * \text{rand};$

Y en el caso del empalme mecánico: $\text{atenuacion_empalme}=0.35 + 0.25 * \text{rand};$

Configuración del conector

El valor predeterminado de los conectores está estipulado en un rango nominal para cada tipo de conector, en el simulador SOTDR se utiliza los conectores más frecuentes empleados en el Ecuador. Los diferentes tipos de conectores seleccionables y su respectiva atenuación se pueden apreciar en la tabla 4.4.

Tabla 4.4. Conectores y su atenuación.

Conector	Atenuación (dB)
FC	0.25
LC	0.20
MT-Array	0.20
SC	0.25
SC-Duplex	0.15
ST	0.20
BFOC	0.20

Fuente. La Investigación

4.4 PRUEBAS Y RESULTADOS

Para la comprensión de SOTDR, se simularán tres enlaces de fibra óptica con el fin de poder apreciar lo que ocurre con las distintas fibras y sus atenuaciones, así como los diferentes tipos de empalmes y conectores. Para ello se emularán tres enlaces, con las siguientes características:

Simulación Uno: Enlace con pocos tramos y de corta duración para apreciar los conectores y empalmes, así como las distintas potencias de entrada. Todo esto se realizará variando la longitud del enlace.

Simulación Dos: Enlace de larga longitud con muchos tramos de fibra óptica utilizando el zoom para apreciar correctamente los resultados.

Simulación Tres: Enlace de larga longitud con fibra de alta atenuación para apreciar la disminución de la relación señal a ruido.

Simulación Uno

Primera parte

La tabla 4.5 muestra los datos de ingreso en el simulador SOTDR para su respectivo análisis.

Tabla 4.5. Datos principales para la simulación uno

Sitio A	Sitio B	Potencia entrada (dBm)	Longitud de onda (nm)	Tipo de fibra	Conector entrada	Conector salida	Empalme	Longitud fábrica (Km)	Longitud total (Km)
Izamba	Ingahurco	0	1310	Monomodo	LC	LC	Fusión	2.1	4.5

Fuente: Investigador

Antes de pasar a hablar de los resultados se puede observar que hay tres tramos de fibra óptica. Dos tramos de 2.1 Km y un tercer tramo de 0.3 Km (además de los enlaces con las centrales). La figura 4.27 muestra el resultado de la simulación uno.

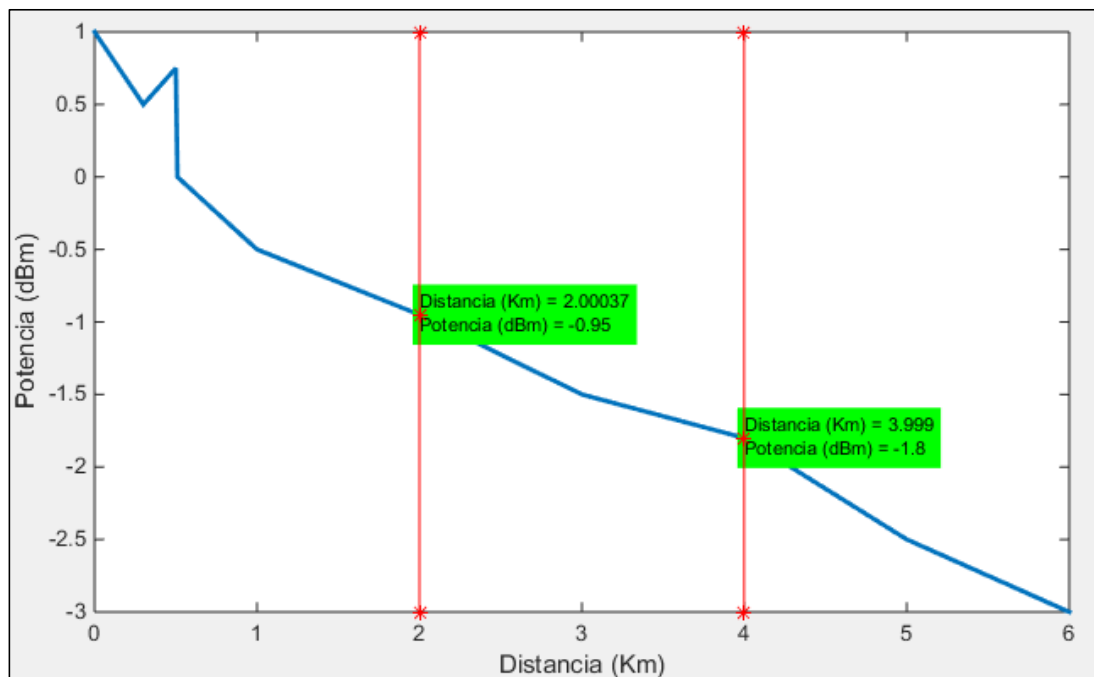


Figura 4.27. Resultado de la simulación uno en SOTDR

Fuente: Investigador

Para visualizar las características del enlace óptico se puede centrar el punto de atención en la parte derecha de la GUI SOTDR, donde aparece el cuadro resumen con distintos parámetros de interés del enlace emulado. A continuación, la figura 4.28 muestra dichos resultados.

RESULTADOS		
Atenuación	0.35	[dB/Km]
Atenuación conector entrada	0.25	[dB]
Atenuación conector salida	0.25	[dB]
Índice de refracción	1.11743	
Atenuación de empalme	0.108623	[dB]
Potencia de enlace óptico	0.85	[W]

Figura 4.28. Cuadro resumen resultados de interés
Fuente: Investigador

En primer lugar, se va a comprobar que la atenuación de la fibra es 0.35 dB/Km. Como se puede apreciar en la figura 4.29. También se puede apreciar que al realizar zoom sobre una zona concreta se puede observar con mayor precisión el ruido adherido a la señal, pero en este caso prácticamente no varía la potencia.

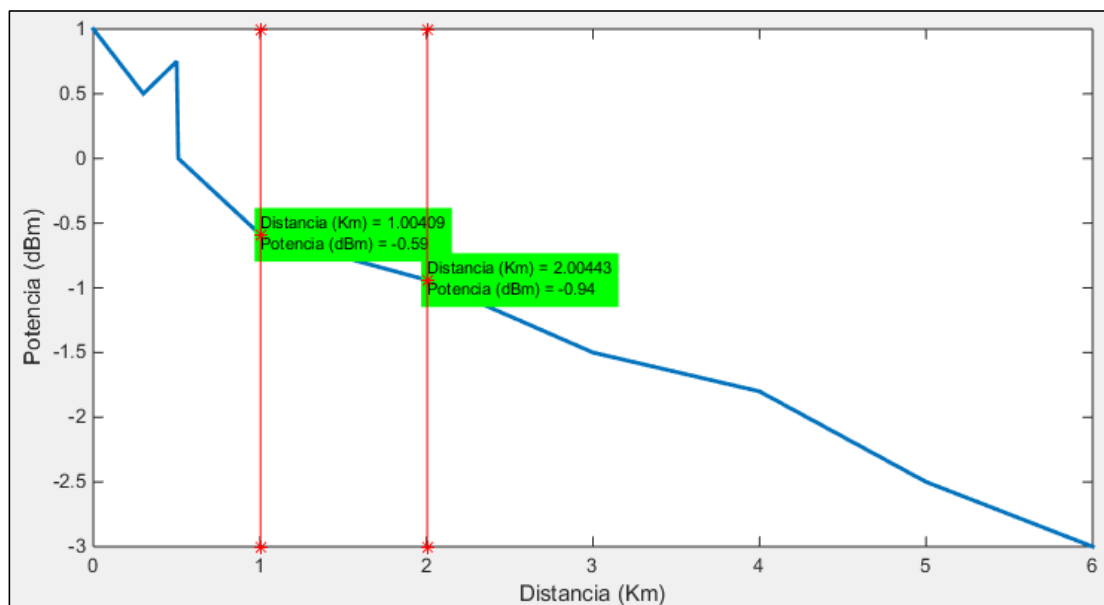


Figura 4.29. Verificación de la atenuación por medio de los cursores

Fuente: Investigador

La figura 4.30 muestra la comprobación de la atenuación con la diferencia entre cursores.

$$(D_{cursor2} - D_{cursor1}) = +1000 \quad (P_{cursor2} - P_{cursor1}) = -0.350$$

Figura 4.30. Segunda verificación diferencia entre cursores

Fuente: Investigador

En segundo lugar, se comprobará que la atenuación de los conectores es de 0.25 dB. Dicha atenuación se cumple para los dos casos, pero con el fin de simplificar el resultado se mostrará en la figura 4.31, únicamente las correspondientes al conector de entrada.

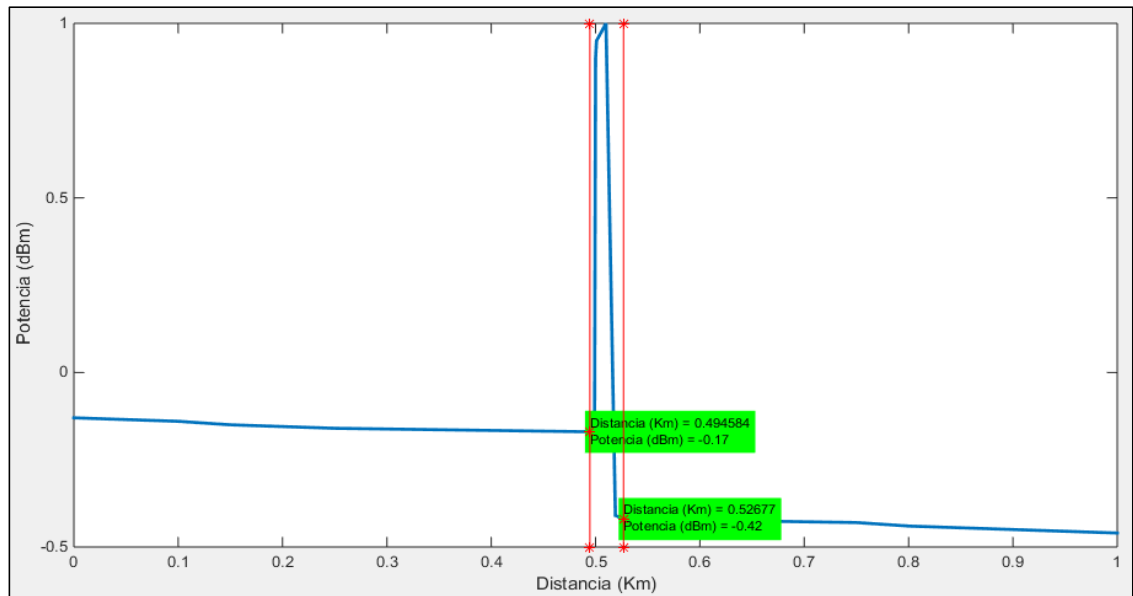


Figura 4.31. Medida de la atenuación del conector de entrada

Fuente: Investigador

La figura 4.32 muestra el resultado de la atenuación del conector de entrada por el método de diferencia entre cursores.

$$(D_{cursor2} - D_{cursor1}) = +0.032 \quad (P_{cursor2} - P_{cursor1}) = -0.250$$

Figura 4.32. Diferencia entre cursores conector de entrada

Fuente: Investigador

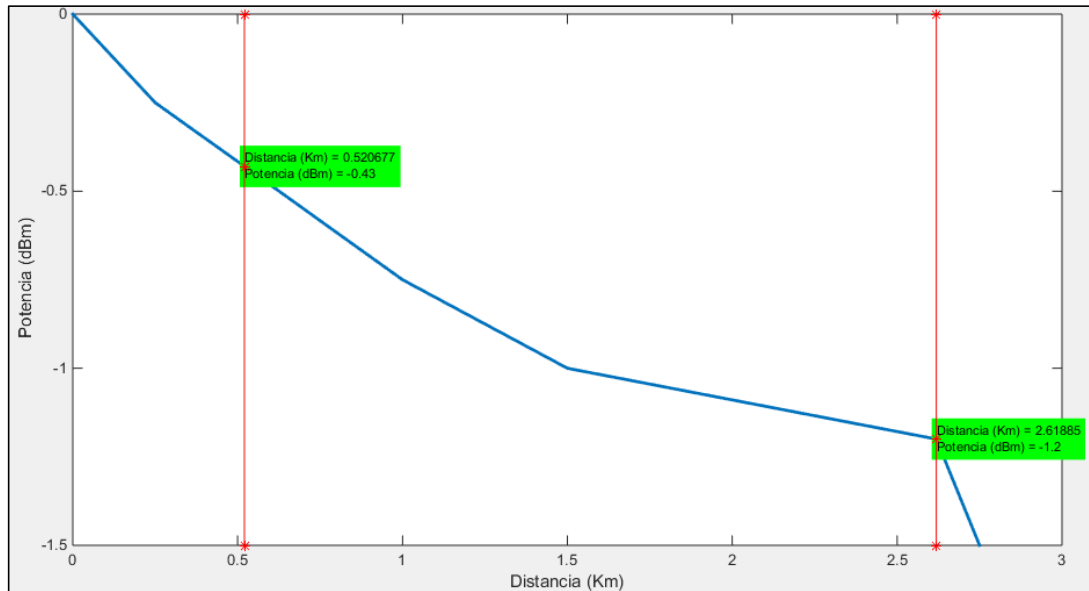


Figura 4.33. Comprobación de la longitud de fábrica
Fuente: Investigador

Se comprueba que la longitud de fábrica sea de 2.1 Km, ya que la longitud total del enlace se puede apreciar que es de 4.5 Km, se suman a los tramos de entrada y salida de la central telefónica (0.5 Km cada uno), que dando una longitud de 5.5 Km, aproximadamente. La anterior figura 4.33 mostró la comprobación de la longitud de fábrica. A su vez, la figura 4.34 muestra la longitud de fábrica por el método de diferencia entre cursores.

$$(D_{cursor2} - D_{cursor1}) = +2.098 \quad (P_{cursor2} - P_{cursor1}) = -0.770$$

Figura 4.34. Comprobación de la longitud de fábrica
Fuente: Investigador

Segunda Parte

A continuación, manteniendo parte de los parámetros de entrada, se procede a simular un enlace de fibra óptica en tercera ventana con conectores y empalmes de peor calidad que en el caso anterior. La tabla 4.6 muestra los datos de ingreso en el simulador SOTDR para su respectivo análisis.

Tabla 4.6. Datos principales para la simulación uno, segunda parte

Sitio A	Sitio B	Potencia entrada (dBm)	Longitud de onda (nm)	Tipo de fibra	Conector entrada	Conector salida	Empalme	Longitud fábrica (Km)	Longitud total (Km)
Izamba	Ingahurco	-10	1550	Monomodo	ST	ST	Mecánico	2.1	4.5

Fuente: Investigador

La figura 4.35 muestra el resultado de la simulación uno en su segunda parte.

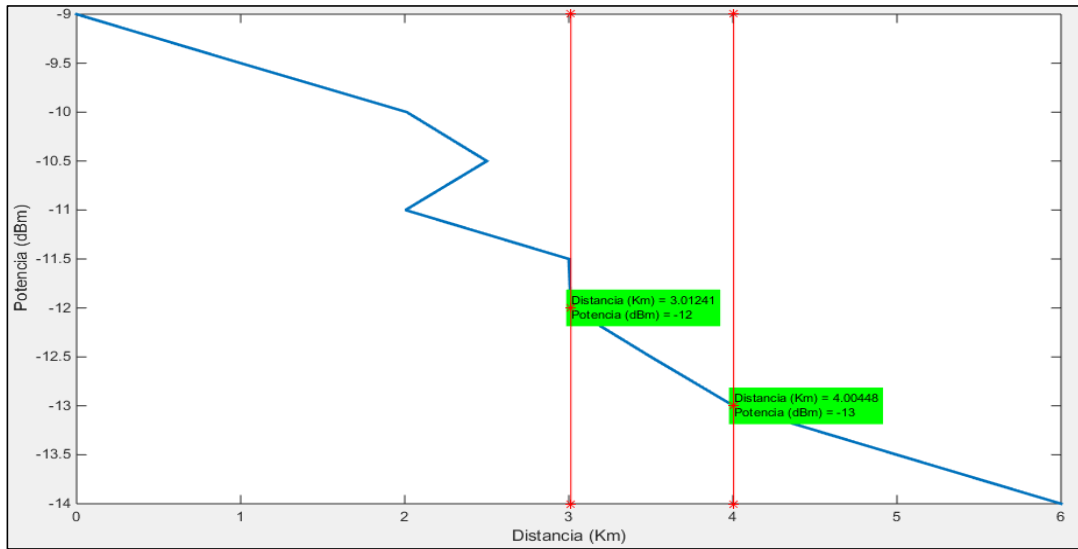


Figura 4.35. Resultado de la simulación uno, segunda parte
Fuente. La Investigación

La figura 4.36 muestra el resultado de la atenuación de la fibra por el método de diferencia entre cursores.

$$(Distancia_{cursor2} - Distancia_{cursor1}) = +0.99 \quad (Potencia_{cursor2} - Potencia_{cursor1}) = -0.19$$

Figura 4.36. Atenuación de la fibra diferencia de cursores
Fuente: Investigador

Como se puede apreciar la atenuación de la fibra óptica en este caso es de 0.19 dB/Km, por lo que no varía demasiado respecto al caso anterior. Las tres grandes diferencias de esta simulación son:

- El cambio de tipo de empalme, pasando de una atenuación de 0.11 dB a una atenuación de 0.416 dB.
- El aumento claramente apreciable de la relación señal a ruido debido al cambio de la potencia de entrada.
- El aumento de la atenuación del conector de entrada, al haber cambiado el tipo del mismo. Se ha pasado de una atenuación de 0.25 a una atenuación de 1.5 dB, dado que el conector ST es un conector que introduce pérdidas muy elevadas.

Simulación Dos

La tabla 4.7 muestra los datos de ingreso en el simulador SOTDR para su respectivo análisis.

Tabla 4.7. Datos principales para la simulación dos

Sitio A	Sitio B	Potencia entrada (dBm)	Longitud de onda (nm)	Tipo de fibra	Conector entrada	Conector salida	Empalme	Longitud fábrica (Km)	Longitud total (Km)
12 de Noviembre	Huachi grande	-10	1550	Monomodo	BFOC	BFOC	Mecánico	1	22

Fuente: Investigador

Para ver como el número de empalmes y conectores, así como, su atenuación afecta a la potencia de salida se realizará un sencillo balance de potencias. El resultado final debe ser unos -30 dBm de potencia de salida como se puede observar en la gráfica anterior. La figura 4.37 muestra el resultado de la simulación dos. En esta simulación los conectores BFOC inyectan una atenuación de 0.20 dB que en conjunto con el coeficiente de atenuación de la fibra óptica monomodo de la ventana de 1550 nm, es de 0.4 dB para el estándar UIT-T G.652 A; entre los dos componentes la atenuación es de 0.24 dB y se suman al cálculo de la atenuación total del sistema.

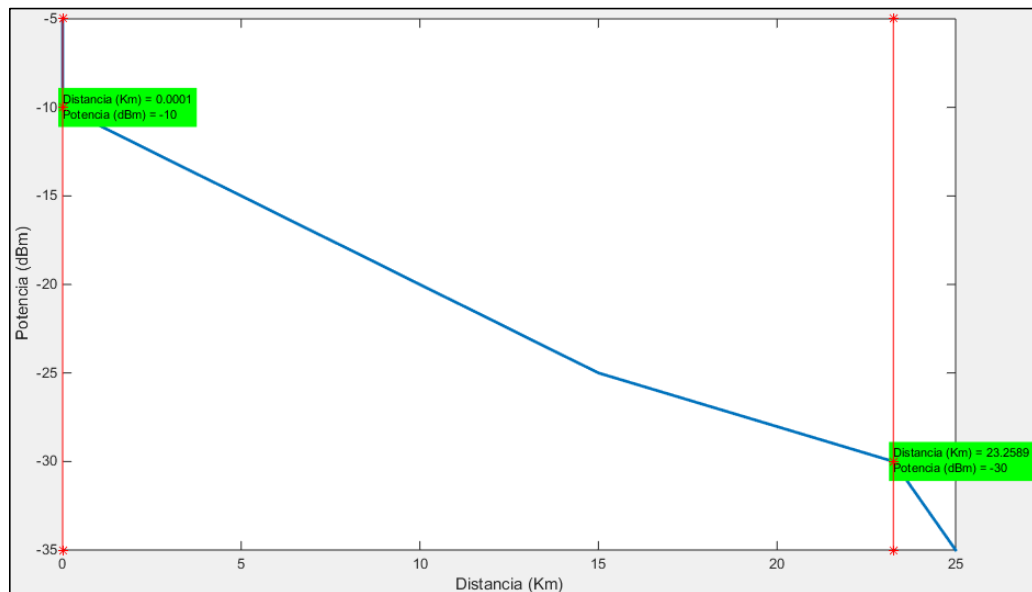


Figura 4.37. Resultado de la simulación dos

Fuente: Investigador

Balance de potencias:

$$P_{salida}(dBm) = P_{entrada}(dBm) - n_c \alpha_c - n_e \alpha_e - L(Km) \alpha_{atenuación} n_{fibra} \quad (55)$$

$$P_{salida}(dBm) = -10(dBm) - 1.5 * 2 - 0.579 * 21 - (22 + 0.5 + 0.5) * 0.2dB/Km$$

$$P_{salida}(dBm) = -29.75923$$

Efectivamente el resultado es $-29.759 \cong -30$ dBm

Se observa que ocurre cambiando el tipo de conector a uno con menor atenuación. También se cambiará el tipo de empalme y el número de los mismos, cambiando la longitud de fábrica. La tabla 4.8 muestra los datos de ingreso en el simulador SOTDR para su respectivo análisis.

Tabla 4.8. Datos principales para la simulación dos, segunda parte

Sitio A	Sitio B	Potencia entrada (dBm)	Longitud de onda (nm)	Tipo de fibra	Conector entrada	Conector salida	Empalme	Longitud fábrica (Km)	Longitud total (Km)
12 de Noviembre	Huachi grande	-10	1550	Monomodo	SC	SC	Fusión	22	22

Fuente: Investigador

La figura 4.38 muestra el resultado de la simulación dos en su segunda parte.

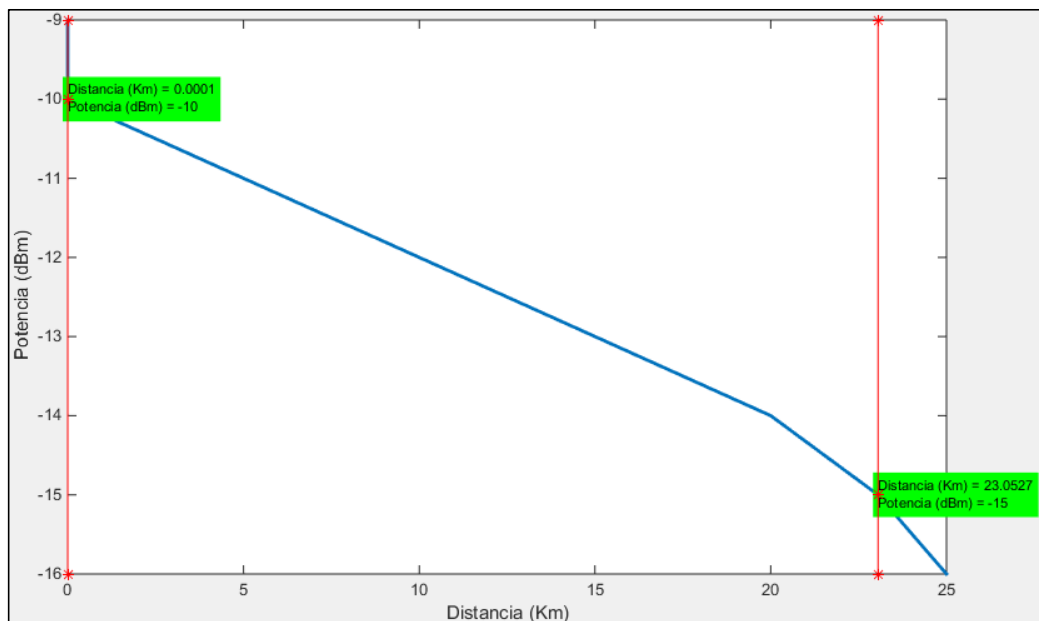


Figura 4.38. Resultado de la simulación dos, segunda parte

Fuente: Investigador

Si se realiza el mismo balance de potencias, para ver la gran diferencia entre ambos casos:

$$P_{salida}(dBm) = P_{entrada}(dBm) - n_c \alpha_c - n_e \alpha_e - L(Km) \alpha_{atenuación\ n_{fibra}} \quad (56)$$

$$P_{salida}(dBm) = -10(dBm) - 0.25 * 2 - 0.14 * 1 - (22 + 0.5 + 0.5) * 0.2dB/Km$$

El resultado ahora es -15.04 dBm de potencia de salida; es decir, una diferencia de 15 dB, 31.6 veces más potencia a la salida de diferencia entre los dos casos.

Simulación Tres

La tabla 4.9 muestra los datos de ingreso en el simulador SOTDR para su respectivo análisis.

Tabla 4.9. Datos principales para la simulación tres

Sitio A	Sitio B	Potencia entrada (dBm)	Longitud de onda (nm)	Tipo de fibra	Conector entrada	Conector salida	Empalme	Longitud fábrica (Km)	Longitud total (Km)
Ingahurco	Huachi Grande	0	850	Multimodo	LC	BFOC	Mecánico	3.5	22

Fuente: Investigador

A diferencia de los casos anteriores, la fibra utilizada tiene una atenuación de 2.9 dB/Km mucho más elevada que en las simulaciones anteriores. Se puede apreciar, que a medida que la distancia se incrementa el ruido cada vez afecta más a la señal.

También se puede apreciar que cuando la potencia sobrepasa el valor de -70 dBm aparece un suelo de ruido, ya que supera los niveles de potencia admisibles para el simulador SOTDR.

A continuación, se va a cambiar el tipo de fibra óptica a una con menor atenuación pero con un enlace de gran tamaño para apreciar también la variación de la relación señal a ruido.

La figura 4.39 muestra el resultado de la simulación tres.

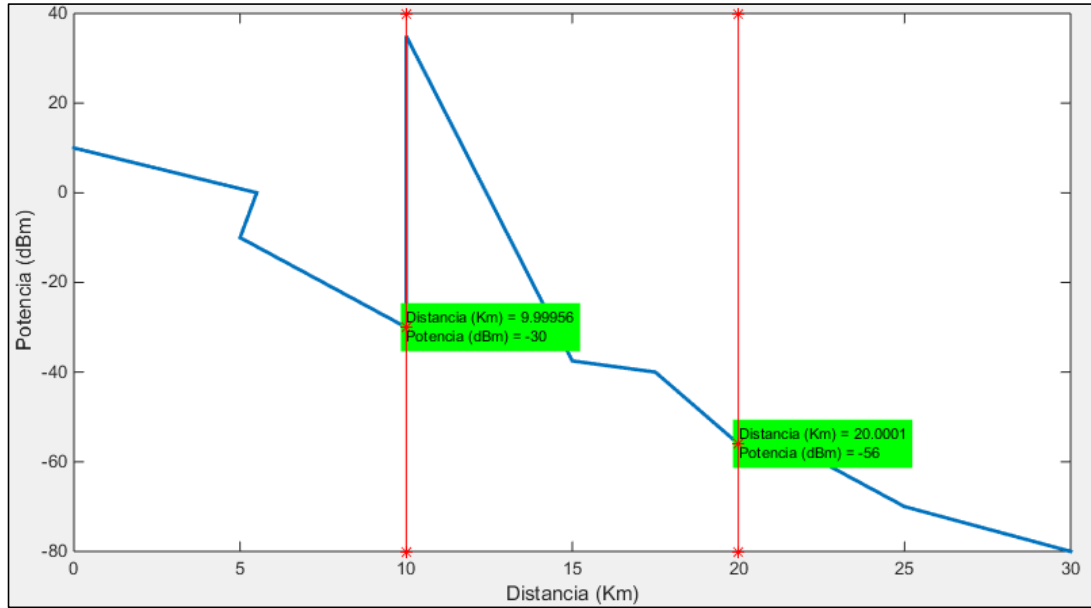


Figura 4.39. Resultado de la simulación tres
Fuente: Investigador

La tabla 4.10 muestra los datos de ingreso en el simulador SOTDR para su respectivo análisis.

Tabla 4.10. Datos principales para la simulación tres, segunda parte

Sitio A	Sitio B	Potencia entrada (dBm)	Longitud de onda (nm)	Tipo de fibra	Conector entrada	Conector salida	Empalme	Longitud fábrica (Km)	Longitud total (Km)
Ingahurco	Huachi Grande	0.2	850	Multimodo	MT-Array	BFOC	Mecánico	3.5	25

Fuente: Investigador

Con este tipo de fibra también se obtiene una relación señal a ruido muy mala llegando a la central de destino. En comparación con el resto de simulaciones, estos dos últimos enlaces son de muy baja calidad en lo que respecta a relación señal a ruido, no obstante, si los enlaces no fueran de tan larga distancia podrían considerarse enlaces adecuados o si se amplificase periódicamente la señal mediante un EDFA cada cierta distancia.

La figura 4.40 muestra el resultado de la simulación tres en su segunda parte.

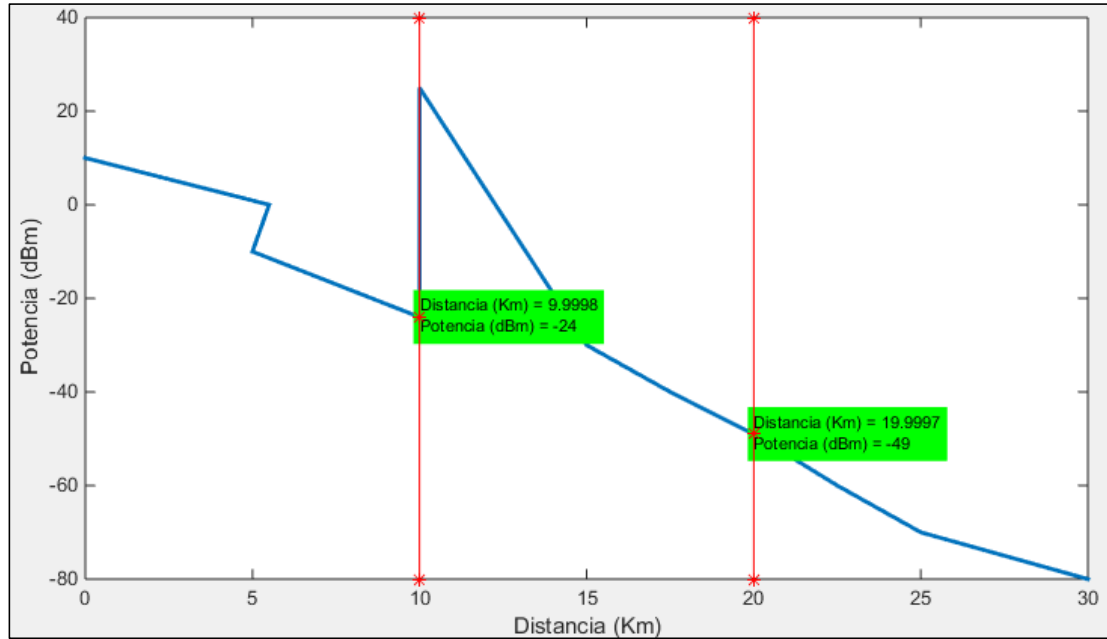


Figura 4.40. Resultado de la simulación tres, segunda parte

Fuente: Investigador

4.5. PRESUPUESTO DEL PROYECTO

4.5.1 Costo de Hardware y software

Tabla 4.11. Tabla de presupuesto.

Nº	DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
1	Computador portátil Asus Intel Core i7	c/u	1	900,00	900,00
2	Licencia Matlab 2014b	c/u	1	950,00	950,00
3	Flash Memory MAXELL de 8 Gb	c/u	1	13,00	13,00
4	Anillados	c/u	3	0,80	2,40
5	Empastados	c/u	2	13,50	27,00
6	Suministros de oficina	c/u	1	250,00	250,00
7	Copias	c/u	200	0,02	4,00
8	Impresiones	c/u	1000	0,05	50,00
9	Internet	hora	400	0,60	240,00
10	Transporte urbano	coste	80	0,30	24,00
	SUB-TOTAL	-----	-----	-----	2460.40
	Imprevistos (5%).	-----	-----	-----	123,02
	TOTAL				2583.42

Fuente: Investigador

*El presupuesto total de este proyecto asciende a la cantidad de **Dos mil quinientos ochenta y tres con 42/100 DÓLARES**. Dinero financiado por el Investigador y la empresa SINFOTECNIA S.A.*

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- En el análisis de pérdidas en la fibra óptica, se constató que una conexión fija permanente como el caso de empalme por fusión introduce menos pérdidas en un enlace (0.3 dB) que un conector (0.75 dB). Puesto que el empalme por fusión presenta un proceso de elaboración con menos impurezas en los extremos de la fibra óptica, mientras que siempre existirá una separación de aire que ocasione mayores eventos reflexivos de la señal óptica en un conector.
- La herramienta de simulación denominada SOTDR posee diferentes tipos de anchos de pulso, los cuales se utilizan en función de la distancia del enlace para obtener buenos resultados. Para distancias pequeñas se utilizan anchos de pulsos pequeños que permiten mayor resolución, sin embargo, para distancias grandes se requiere anchos de pulso grandes en los cuales el ruido no afecte la medición. Por último, para encontrar el mejor ancho de pulso se utilizaron como referencia los valores definidos por fabricante.
- El programa MATLAB y su componente GUIDE fue una de las herramientas más apropiadas para la simulación en el ámbito de las comunicaciones, tanto por la capacidad de programación numérica como por las facilidades de desarrollo gráfico que permite. El diseño lógico de SOTDR ha sido implementado satisfactoriamente, facilitando las tareas de configuración, cálculo y visualización de los datos obtenidos mediante un analizador de la traza reflectométrica; para inducir en un ambiente de simulación los enlaces que conforman las comunicaciones por fibra óptica, al nivel de emular la realidad operativa de una instalación.

- Al seleccionar fuentes ópticas con mayor ancho espectral en fibras con mayor dispersión, disminuye el límite máximo de tasa de transmisión de datos soportada por el enlace; es decir, para lograr altas tasas de transmisión, es necesario escoger fuentes de precisión y fibras de baja dispersión; corroborando las ventajas de usar un láser en lugar de un led, al usar el láser se logró transmitir a altas tasas, esto se debe a su ancho espectral de 50 veces menos, en el peor de los casos; y a su tiempo de subida 10 veces menor que el led.
- El método de medida que se ha empleado en el diseño, la reflectometría óptica en el dominio del tiempo, OTDR; ha podido comprobar mediante simulación que tanto la estructura del sistema propuesto como el algoritmo de procesamiento de señal ofrecen resultados excelentes al trabajar con valores de atenuación de los componentes pasivos y niveles de potencia de los emisores tomados de equipos reales, logrando un resultado fiable con el cual se puede guiar para establecer enlaces de comunicaciones por fibra óptica en la realidad.
- La elección de los correctos elementos a la hora de diseñar un enlace óptico (empalmes, conectores, tipos de fibra), es tanto importante como necesario para obtener unos niveles de potencia y atenuación adecuados a la salida. Al igual que en el resto de sistemas de comunicaciones, se ha comprobado, lo necesario que es mantener una relación señal a ruido adecuada, con el fin de recibir a la salida del enlace un nivel de potencia por encima de un umbral, indispensable para el correcto funcionamiento del sistema.

5.2. RECOMENDACIONES

- El simulador SOTDR en sus posteriores versiones puede agregar una intención de universalidad en su proceso de desarrollo para adquirir la posibilidad de añadir nuevos parámetros de configuración como la inclusión de instrumentos de medida, o tarjetas adquisidoras de datos. Por tal razón, se recomienda poner en práctica la teoría desarrollada en este proyecto de titulación, con el fin de establecer modelos didácticos que servirían para una mejor comprensión del funcionamiento de sistemas de comunicaciones por fibra óptica.
- Con el fin de diseñar un enlace de comunicaciones por fibra óptica se deben considerar la atenuación y la capacidad de transmisión requerida para garantizar que la atenuación no afecte el tráfico de información se realiza un balance de

potencias, en este se resta de la potencia del transmisor todas las atenuaciones que se producen en un enlace óptico y se verifica que la potencia de llegada al punto de destino sea mayor a la sensibilidad del receptor.

- Es importante recordar que los empalmes, conectores y la atenuación que presenta la fibra escogida son pérdidas de potencia que se producen a lo largo del enlace, estas deben de considerarse y analizarse previamente a escoger el transmisor y receptor para así poder tener una idea amplia acerca de los parámetros exactos que estos deben cumplir para que la señal llegue a su destino.
- El presente trabajo permitirá tener una visión más amplia de los sistemas de comunicación por fibra óptica, pues el simulador didáctico implementado accederá al usuario a asumir un contacto virtual con esta tecnología aproximándose a un plan de gestión de redes ópticas.
- Organizar la carpeta del programa de simulación por tipo de archivo, de tal forma que, al guardarse los documentos del proyecto a diseñar, en la raíz de la partición principal del disco duro del computador (C, o D); estos se visualicen en orden consecutivo. Además, el uso del simulador SOTDR es coherente bajo versiones de MATLAB iguales o superiores a la usada en el diseño y desarrollo del mismo.
- Se recomienda el uso del simulador SOTDR para redes de comunicación por fibra óptica, bajo versiones de MATLAB iguales o superiores a la usada en el diseño y desarrollo del mismo.

BIBLIOGRAFÍA

- [1] VILLACIS GARCIA, Verónica A., “Análisis de conectividad por fibra óptica en la Corporación Nacional de Telecomunicaciones (CNT E.P.) Cotopaxi para las centrales Corazón – Moraspungo - Provincia de Cotopaxi San Luis Provincia de Bolívar”, Universidad de Guayaquil, Ecuador, Guayaquil, Tesis, Año 2015. [En línea]. Disponible en: <http://repositorio.ug.edu.ec/bitstream/redug/6986/1/AL%20FIN%20TESIS%20-%20copia.pdf>. Último acceso: 12/04/2017
- [2] GÓMEZ PAUTA, Wilson L., “Desarrollo de aplicación de software para automatizar el diseño de enlaces ópticos y su aplicación en un enlace para la central Minas-San Francisco”, Universidad del Azuay, Ecuador, Cuenca, Tesis, Año 2015. [En línea]. Disponible en: <http://dspace.uazuay.edu.ec/bitstream/datos/4858/1/11300.pdf>. Último acceso: 23/04/2017
- [3] DA CUNHA COSTA, Ulisses W., “Análise da influência do monitoramento OTDR em WDM-PON: Um Estudo Baseado em Simulação”, Universidade Federal Do Pará, Belém, Tesis, Año 2013. [En línea]. Disponible en: http://www.repositorio.ufpa.br:8080/jspui/bitstream/2011/4970/1/Dissertacao_AnaliseInfluenciaMonitoramento.pdf. Último acceso: 17/05/2017
- [4] ROMERO CORTÉS, Luis, “Reflectometría óptica en el dominio de la frecuencia (OFDR) para la caracterización de componentes y dispositivos ópticos”, Universidad de Sevilla, España, Sevilla, Tesis, Año 2012. [En línea]. Disponible en: http://bibing.us.es/proyectos/abreproy/12002/fichero/Luis-Romero-Cortes_49083148k_PFC-Memoria.pdf. Último acceso: 23/05/2017
- [5] TONELLO, Bruno, “OTDR-based Monitoring in Passive Optical Networks”, Università degli Studi di Padova, Italia, Praga, Tesis, Año 2015. [En línea]. Disponible en: http://tesi.cab.unipd.it/47173/1/Tonello_Bruno_thesis.pdf. Último acceso: 24/05/2017

- [6] GOVIND, P., Agrawal, “Fiber-Optic Communication Systems”, Wiley Interscience, Academic Press, Año 2010, pp. 8-15.
- [7] GOVIND, P., Agrawal Clifford Headley, “Raman Amplification in Fiber Optical Communication Systems”, Elsevier Academic Press, Año 2011, pp. 7-11.
- [8] GOVIND, P., Agrawal, “Nonlinear Fiber Optics”, Wiley Interscience, Academic Press, Año 2010, pp. 21-25.
- [9] CHOMYCZ, Bob. “Instalaciones de fibra óptica: fundamentos, técnicas y aplicaciones”, Editorial McGraw-Hill, Interamericana de España, Año 2002, pp. 31-32.
- [10] AGUILERA, S., MENÉNDEZ, P. “Conceptos Fundamentales de Comunicaciones Ópticas”, Docencia del Departamento en E.T.S.I de Telecomunicación, Año 2009, pp. 37.
- [11] TOMASÍ, W. “Sistemas de comunicaciones electrónicas”, Editorial Pearson Educación, Año 2003, pp. 453-455.
- [12] CONECTRONICA. “Comunicaciones Ópticas”, Tecnología-electrónica, España, Año 2014. [En línea]. Disponible en: <http://www.conectronica.com/comun%opticas%2.html/>. Último acceso: 13/01/2018.
- [13] PEREDA, J. A. “Sistemas y redes ópticas de comunicaciones”, Editorial Pearson Educación, Año 2004, pp. 182-183.
- [14] BLAKE, R. “Sistemas electrónicos de comunicaciones”, Editorial Thomson, Ontario, Canadá, Año 2004, pp. 890 – 891.
- [15] BOQUERA, M. “Tutorial de Comunicaciones Ópticas”, Grupo de Comunicaciones Ópticas, España, Año 2012. [En línea]. Disponible en: <http://nemesis.gco-kernel.tel.uva.es/>. Último acceso: 11/01/2018.
- [16] SANTACRUZ, O. “Principios generales del sistema de fibra óptica”, España, Año 2008. [En línea]. Disponible en:

<http://www.profesores.frc.utn.edu.ar/PlantelExt/Resumen%20FO.pdf>. Último acceso: 14/01/2018.

[17] LÓPEZ, M. “Sistemas de Transmisión”, Perú, Año 2016. [En línea]. Disponible en: <http://sx-de-tx.spaces.com/page/pdf/FTTx?responseToken=9cb4d356>. Último acceso: 15/01/2018.

[18] PÉREZ, H. “La guía de las comunicaciones ópticas”, Canadá: Exfo, Año 2016. [En línea]. Disponible en: http://www.c3comunicaciones.es/Documentacion/Guia%20com_opticas.pdf. Último acceso: 16/01/2018.

[19] LATTANZI, M. “Redes de comunicaciones por fibra óptica”, Argentina, Año 2011. [En línea]. Disponible en: <http://www.cicomra.org.ar/cicomra2/tutorial%20graf-%20ieee.pdf>. Último acceso: 17/01/2018.

[20] GARCÍA, A. “Introducción y conceptos generales de las comunicaciones ópticas”, España, Año 2014. [En línea]. Disponible en: <http://www.ccapitalia.net/descarga/docs/copticas-introduccion-conceptos.pdf>. Último acceso: 17/01/2018.

[21] PALLO, Juan P. “Folleto Comunicaciones Ópticas”, Universidad Técnica de Ambato, FISEI, Ecuador, Ambato, Año 2013, pp. 34 – 47.

[22] PALADINES, C. “Diseño de una red GPON para la localidad de Vilcabamba”, Universidad Particular de Loja, Ecuador, Loja, Año 2014. [En línea]. Disponible en: <http://dspace.utpl.edu.ec/bitstream/8473/1/%20Palidenes-Cesar%al%20dspace.pdf>. Último acceso: 19/01/2018

[23] MARTÍN LÓPEZ, Sonia, “Generación de supercontinuo en fibras ópticas monomodo con fuentes de bombeo continuo”, Universidad Complutense de Madrid, España, Madrid, PhD Tesis, Año 2006. [En línea]. Disponible en: <http://digital.csic.es/bitstream/10261/11290/1/Generaci%C3%B3n%20de%20supercontinuo%20.pdf>. Último acceso: 27/05/2017

- [24] DERICKSON, Dennis, “Fiber Optic Test and Measurement”, Hewlett-Packard Professional Books, Año 2013, pp. 11-16.
- [25] FROGGATT, M., MOORE, Eric, “Interferometric measurement of dispersion in optical components”, Luna Technologies, Año 2016, pp. 5-10.
- [26] KIEFFER, J., AZAÑA, J., PARK, Y., “Real time optical frequency domain reflectometry”, Papper, Año 2014, pp. 3-5.
- [27] JDSU.COM, “Descripción y características de los OTDR JDSU”. [En línea]. Disponible en: <http://www.jdsu.com>. Último acceso: 14/01/2017
- [28] WOLFE, S., FROGGATT, M., SOLLER, J., “High resolution optical frequency domain reflectometry for characterization of components and assemblies”, Luna Technologies, Año 2016, pp. 3-4.
- [29] BANEY, Douglas, VANWIGGEREN, Gregory, MOTAMEDI, Ali, “Single-scan interferometric component analyzer”, Año 2003, pp. 5-6.
- [30] MURIEL, Miguel, CAPMANY, José, “A new transfer matrix formalism for the analysis of fiber ring resonators: Compound coupled structures for FDMA demultiplexing”, Scientific paper, Año 2015, pp. 10-29.
- [31] MATEMATICAS.US, “Descripción teórica de los software de proceso matemático y Sistemas de Cálculo Numérico (SCN)”. [En línea]. Disponible en: <http://www.matematicas.us.es/informacion-general-mp/software-matematico>. Último acceso: 12/07/2017
- [32] MATHWORKS.ORG, “Fundamentación y descripción de MATLAB”. [En línea]. Disponible en: <http://www.mathworks.org/matlab/documentation.html>. Último acceso: 12/07/2017
- [33] SOFTIN.INFO, “Antecedentes y argumentos teóricos de LABVIEW”. [En línea]. Disponible en: <http://softin.info/labview-long/>. Último acceso: 13/07/2017

[34] SCILAB.ORG, “Documentación y fundamentación contextual del programa SCILAB”. [En línea]. Disponible en: <http://www.scilab.org/products/scilab/documentation>. Último acceso: 13/07/2017

[35] GNU.ORG, “Fundamentos y características principales de GNU Octave”. [En línea]. Disponible en: <http://www.gnu.org/software/octave/documentation>. Último acceso: 14/07/2017

[36] MAPLESOFT.COM, “Descripción y fundamentación específica de MAPLE”. [En línea]. Disponible en: <http://www.maplesoft.com/maple/documentation>. Último acceso: 14/07/2017

[37] BLANCO, R. “Desarrollo de una GUI para el estudio de Sistemas de Comunicaciones Ópticas”, Universidad Politécnica de Madrid, España, Año 2012. [En línea]. Disponible en: http://oa.upm.es/21757/2/PFC_RODRIGO_BLANCO_MARTIN_BUENISIMA.pdf. Último acceso: 20/12/2017

[38] BRAVO, B., SÁNCHEZ, L. “Simulador de enlace de fibra óptica punto a punto usando interfaz gráfica de MATLAB”, Escuela Superior Politécnica del Litoral, Ecuador, Guayaquil, Año 2012. [En línea]. Disponible en: <https://www.dspace.espol.edu.ec/bitstream/123456789/21566/2/Simulador%20enlace%20de%20fibra%20C3%B3ptica%20punto%20a%20punto%20usando%20MatLab.pdf>. Último acceso: 21/12/2017

[39] MATHWORKS. “Desarrollo de aplicaciones GUI”, Toolbooks GUIDE, Año 2015. [En línea]. Disponible en: http://www.MathWorks.com/matlab%gui_guide.html. Último acceso: 14/01/2018.

ANEXOS

ÍNDICE DE ANEXOS

ANEXO A: Manual de Usuario	111
ANEXO B: Código de Programación	131

ANEXO A: Manual de Usuario

MANUAL DE USUARIO



SOTDR
UTA-FISEI-COPYRIGHT

ÍNDICE MANUAL DE USUARIO

1. INTRODUCCIÓN	113
2. OBJETIVOS	113
3. REQUISITOS	113
4. CONOCIMIENTOS PREVIOS	114
5. MENSAJES DE SEGURIDAD	114
6. SIMULADOR SOTDR	117

1. INTRODUCCIÓN

El presente manual está desarrollado con el propósito de proporcionarle al usuario una orientación del modo de utilizar el Simulador SOTDR. Cada opción es explicada de una forma práctica y con ilustraciones para su mejor comprensión. Para argumentar la operatividad del Simulador SOTDR, es necesario conocer su funcionalidad.

El Simulador SOTDR es un software de emulación para simular el diseño de un enlace de fibra óptica, y su respuesta en atenuación y potencia como comportamiento de los parámetros reales ingresados en los componentes transmisor, receptor, fibra óptica; el simulado ofrece al usuario al igual que el equipo OTDR profesional, la respectiva traza reflectométrica de la instalación, en la cual se puede apreciar la atenuación del enlace para para los tramos respectivos del tendido de fibra óptica. Permite reflejar métricas inherentes y de gran importancia para la gestión de las instalaciones de fibra óptica, entre los cuales destacan la atenuación de la fibra, la atenuación de los empalmes y conectores y la atenuación total del enlace; al igual que la potencia recibida en el receptor con su presupuesto, información muy importante en la implementación real de una red de comunicación óptica. El Simulador SOTDR tiene una estructura adaptativa de software y es intuitivo en su utilidad, de manera ergonómica presenta el diseño de sus interfaces.

2. OBJETIVOS

- Proporcionar al usuario la información necesaria para utilizar el simulador SOTDR.
- Describir instrucciones apropiadas para el uso, manipulación y operatividad del simulador SOTDR.

3. REQUISITOS

Para la utilización del simulador SOTDR se debe tener los siguientes requerimientos básicos:

- Sistema Operativo Windows 8.1.
- MATLAB 2014b o versiones posteriores.
- Espacio en el disco duro superior a 8 GB.
- Memoria RAM superior a 1GB.

4. CONOCIMIENTOS PREVIOS

Para el uso correcto y la comprensión eficiente de las instrucciones que actúan en el simulador SOTDR, es necesario poseer conocimientos básicos de los componentes y la programación orientada a objetos del software MATLAB GUIDE; además de la tecnología de sistemas de comunicaciones por fibra óptica, conocer fundamentos de los componentes de un enlace de fibra óptica, de las características y tipos de fibra óptica, características y fundamentos de receptores y transmisores ópticos, características y fundamentos de conectores y empalmes; proporcionan un mejor entendimiento de la operatividad del simulador SOTDR.

El simulador SOTDR es un software emulador que permite, una vez configurado, compilado y ejecutado; emular el comportamiento de un enlace de fibra óptica con el propósito de efectuar una gráfica semejante a una traza reflectométrica, en la que se reflejan la atenuación versus la distancia, con la posibilidad de manipular dos cursores para ver la atenuación que se presenta durante el trayecto de la red óptica, esto permite al operario del simulador establecer una gestión de la instalación óptica para la toma de decisiones.

5. MENSAJES DE SEGURIDAD

Como en todo buen programa, es fundamental que la ejecución del mismo sea clara e intuitiva y de alguna forma sepa guiar al usuario mediante la presencia de carteles informativos y ayudas cuando éste cometa un error o se olvide de rellenar un parámetro, por ejemplo. Es por ello que se mostrarán algunos de los mensajes de avisos y/o errores que pueden aparecer durante la ejecución del simulador SOTDR.

Cuando se utilice el simulador SOTDR, se debe prestar atención a los cuadros de diálogo o mensajes de avisos y/o errores como norma de seguridad en la configuración del enlace óptico, a fin de evitar resultados efímeros y equivocados. Los errores están ordenados por orden de posible aparición, es decir, por interfaces.

Es posible, no obstante, encontrar que varias de las interfaces no contienen avisos. Por favor tenga en cuenta los siguientes cuadros de diálogo:

1.- Transmisor

El primero de los mensajes de avisos se puede presentar en la primera de las interfaces de la simulación. Cuando no se selecciona ningún tipo de transmisor.

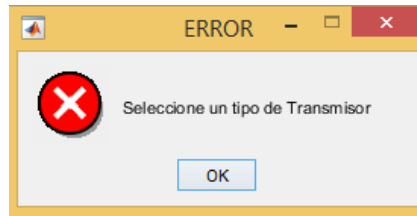


Figura A.1. Error: Seleccione un transmisor

Otro mensaje de error ocurre cuando no se selecciona un tipo de Bit Rate.

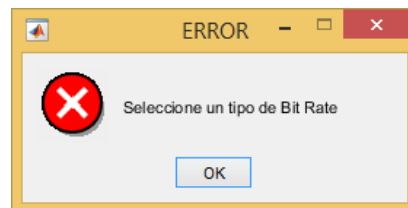


Figura A.2. Error: Seleccione un tipo de Bit Rate

Otro error ocurre cuando no se selecciona un tipo de modulación.

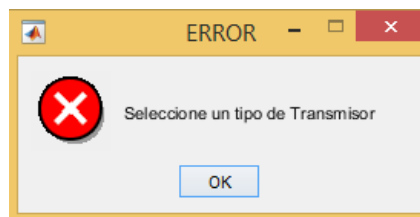


Figura A.3. Error: Seleccione un tipo de Modulación

Al digitar en algún campo numérico, información que no corresponda a un número.

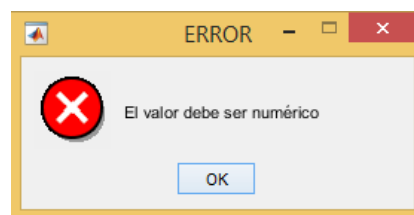


Figura A.4. Error: Seleccione un tipo de Modulación

2.- Fibra Óptica

El primero de los mensajes de avisos que se puede presentar en la interfaz fibra óptica de la simulación. Cuando se selecciona un estándar UIT-T que no corresponde a la longitud de onda seleccionada en la interfaz Enlace_Óptico.

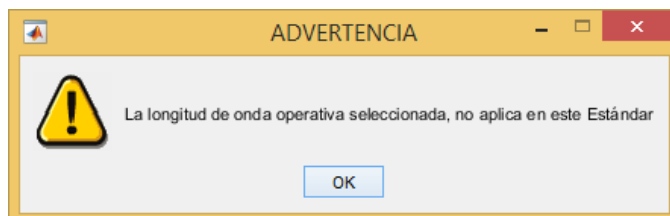


Figura A.5. Advertencia: La longitud de onda no corresponde al estándar UIT-T

El segundo de los mensajes de avisos que se puede presentar en la interfaz fibra óptica de la simulación. Cuando se selecciona un tipo de fibra óptica que no corresponde al estándar UIT-T seleccionado y su respectiva longitud de onda seleccionada en la interfaz Enlace_Óptico.

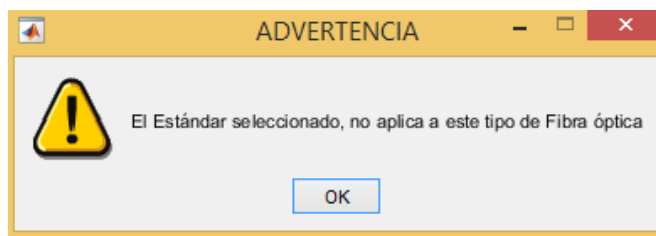


Figura A.6. Advertencia: La fibra óptica no corresponde al estándar UIT-T

3.- Receptor

El primero de los mensajes de avisos que se puede presentar en la interfaz receptor de la simulación. Cuando no se selecciona algún tipo de receptor.

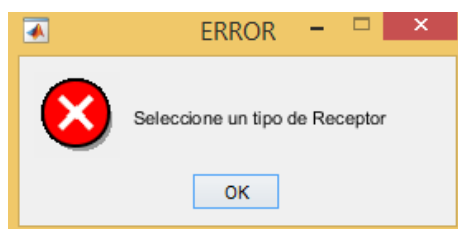


Figura A.7. Error: Seleccione un tipo de Receptor

4.- Enlace Óptico

El primero de los mensajes de avisos que se puede presentar es cuando se digita una distancia del empalme superior a la longitud total del enlace óptico.



Figura A.8. Advertencia: La ubicación del empalme es mayor a la del enlace

Un cuadro de diálogo que se presenta en la interfaz enlace óptico y en otras interfaces de la simulación. Cuando se presiona el botón de salir del simulador SOTDR.

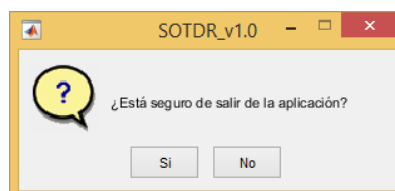


Figura A.9. Cuadro de Diálogo: Salir de SOTDR

6. SIMULADOR SOTDR

La pantalla de inicio de la GUI desarrollada, se inicializa con el archivo Portada.m, el cual siendo llamado desde la ventana de comandos de MATLAB o desde el editor de MATLAB, muestra la siguiente figura.



Figura A.10. Portada del simulador SOTDR

Una vez presionado el botón de inicio, se abrirá la interfaz Enlace Óptico. Esta GUI desarrollada se puede dividir en cuatro bloques: la parte de parámetros del enlace

óptico, una parte de datos de los componentes, parte de los conectores y una parte utilizada para la visualización de los resultados.



Figura A.11. Portada del simulador SOTDR

1.- Bloque Parámetros del Enlace Óptico

En la primera parte se escoge los parámetros del enlace óptico como es: La longitud de onda operativa como se muestra en la figura al dar clic el popuymenú.

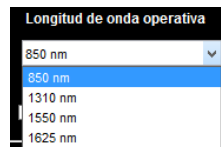


Figura A.12. Selección de longitud de onda operativa

En segundo lugar se establece la longitud del enlace por medio del slider, cuya distancia va de 1 a 100 Km como tentativa del prototipo.

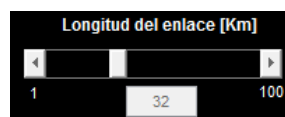


Figura A.13. Selección de longitud del enlace

La siguiente información es la tasa de transmisión, cuyo dato es heredado de la interfaz Transmisor, por tal razón el cuadro de texto se ejecuta de manera inhabilitada.

Figura A.14. Campo de la tasa de transmisión

Por último en este bloque se encuentra un checkbox y un cuadro de texto para el nombre del enlace; al activar el checkbox la configuración de este bloque se cargará.

Figura A.15. Parámetros seleccionados y Nombre del enlace

2.- Bloque Datos de los Componentes

En este bloque de la interfaz Enlace_Optico, se encuentran tres botones que permiten el acceso a las interfaces Transmisor, Fibra Óptica y Receptor para su respectiva configuración.



Figura A.16. Bloque datos de los componentes

2.1.- Transmisor

Al dar clic en el botón Transmisor, se abre la interfaz Transmisor para su respectiva configuración como se muestra en la figura.

Figura A.17. Interfaz Transmisor

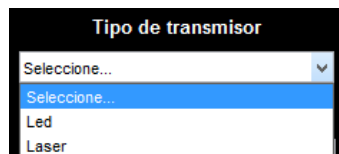
En la parte inicial de la interfaz Transmisor se configura el Sitio A, desde donde inicia la instalación de fibra óptica, en este caso puede ser una localidad, un barrio, una central telefónica, etc.



The image shows a configuration interface with two fields. The first field is labeled 'Sitio A' and is an empty text input box. The second field is labeled 'Tipo de transmisor' and is a dropdown menu with the text 'Seleccione...' and a downward arrow.

Figura A.18. Sitio A y tipo de transmisor

Al habilitar la selección del tipo de transmisor se puede escoger entre led y láser.



The image shows a dropdown menu titled 'Tipo de transmisor'. The menu is open, showing a list of options: 'Seleccione...' (highlighted in blue), 'Led', and 'Laser'.

Figura A.19. Tipo de transmisor

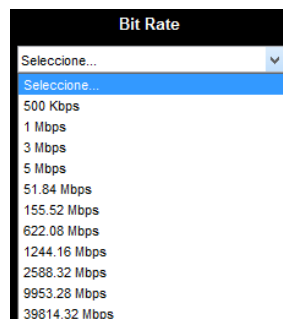
En la siguiente configuración se encuentra el Fabricante o modelo del transmisor, junto con el Bit Rate el cual puede ser seleccionado a través de un popuymenú.



The image shows a configuration interface with two fields. The first field is labeled 'Fabricante/modelo' and is an empty text input box. The second field is labeled 'Bit Rate' and is a dropdown menu with the text 'Seleccione...' and a downward arrow.

Figura A.20. Fabricante/modelo y Bit Rate

Al habilitar la selección del Bit Rate se puede escoger entre una variedad de velocidades de transmisión.



The image shows a dropdown menu titled 'Bit Rate'. The menu is open, showing a list of options: 'Seleccione...' (highlighted in blue), '500 Kbps', '1 Mbps', '3 Mbps', '5 Mbps', '51.84 Mbps', '155.52 Mbps', '622.08 Mbps', '1244.16 Mbps', '2588.32 Mbps', '9953.28 Mbps', and '39814.32 Mbps'.

Figura A.21. Bit Rate

En la siguiente configuración se encuentra la Potencia, junto con el tipo de Modulación el cual puede ser seleccionado a través de un popuymenú.

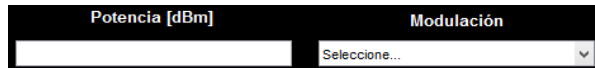


Figura A.22. Potencia y Modulación

Al habilitar la selección del tipo de Modulación se puede escoger entre NRZ, RZ y Manchester.

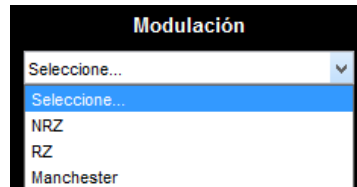


Figura A.23. Modulación

En la siguiente configuración se encuentra el Tiempo de subida del transmisor, junto con la Ventana de transmisión, la misma que es heredada de la interfaz Enlace_Óptico por esta razón se encuentra inhabilitada al momento de su ejecución.



Figura A.24. Tiempo de subida y Ventana de transmisión

En la última configuración se encuentra el Bit Rate ya establecido, junto con el ancho espectral del transmisor. El Ancho espectral es digitado según las especificaciones del fabricante del transmisor y su unidad es en nm.



Figura A.25. Tiempo de subida y Ventana de transmisión

Los botones de la interfaz Transmisor permiten, como su nombre lo indica cargar los datos a la interfaz principal la Enlace_Optico y el botón borrar a su vez permite eliminar todos los datos configurados para establecer otro tipo de transmisor.



Figura A.26. Botones Borrar y Cargar

2.2.- Fibra Óptica

Al dar clic en el botón Fibra Óptica, se abre la interfaz Fibra_Optica para su respectiva configuración como se muestra en la figura.

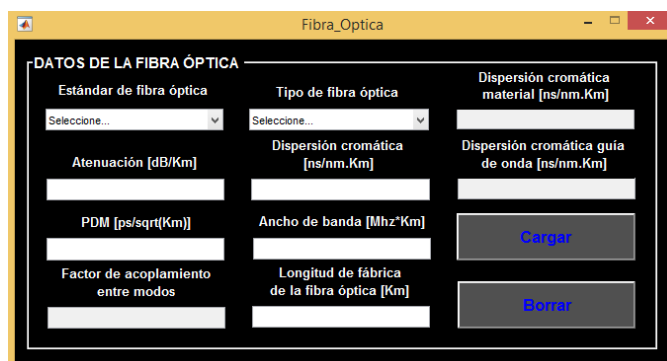


Figura A.27. Interfaz Fibra_Optica

En la parte inicial de la interfaz Fibra_Optica se configura el Estándar de fibra óptica, el tipo de fibra óptica y la dispersión cromática del material, en este caso la dispersión cromática es un parámetro inhabilitado debido a que no es un valor trascendental para la traza reflectométrica o el presupuesto de potencia.

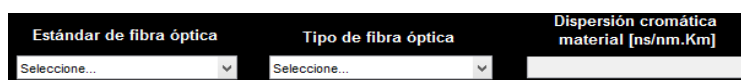


Figura A.28. Estándar y tipo de fibra óptica, dispersión cromática del material

Al habilitar el popuymenú del estándar de fibra óptica se puede escoger entre los estándares normalizados por la UIT-T para fibras monomodo y multimodo, como norma general en las instalaciones ópticas.

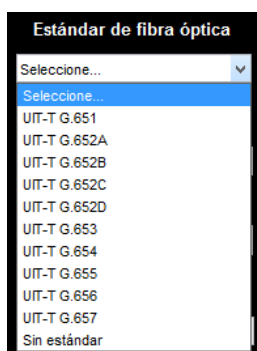


Figura A.29. Estándar de fibra óptica

Al habilitar el popupmenú del tipo de fibra óptica se puede escoger entre la fibra óptica monomodo y multimodo; se establecio este parámetro de esta manera en base a las recomendaciones UIT-T, que predominan ante los nombres de la fibra según las especificaciones de los fabricantes.

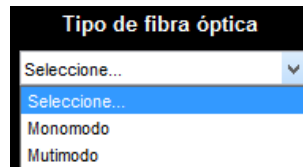


Figura A.30. Tipo de fibra óptica

En la siguiente parte de la interfaz Fibra_Optica se configura la atenuación de la fibra óptica, la dispersión cromática de la fibra óptica y la dispersión cromática de la guía de onda, para este último caso el prototipo no utiliza el valor de la dispersión cromática de guía de onda por tal razón lo mantiene inhabilitado en su ejecución, por no ser imperante en los datos respectivos a la traza reflectométrica.

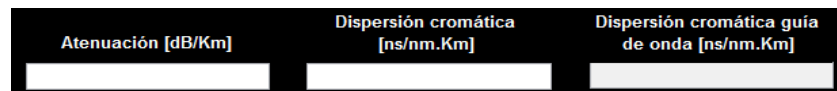


Figura A.31. Atenuación, Dispersión cromática y dispersión cromática de guía de onda

En la siguiente parte de la interfaz Fibra_Optica se configura el PDM y el ancho de banda en MHz por Km; los dos valores son numéricos y tienen que establecerse para el tipo de fibra monomodo o multimodo; de ellos el dato más importante es el ancho de banda el mismo que es heredado para los respectivos cálculos en las otras interfaces.



Figura A.32. PDM y Ancho de banda

En la última configuración de la interfaz Fibra_Optica se encuentra el factor de acoplamiento entre modos y la longitud de fábrica de la fibra óptica, de los cuales el dato más importante es la longitud de fábrica, información que es relevante para la longitud del enlace y para determinar cuantos carretes de fibra serán empleados en la

instalación óptica. El factor de acoplamiento está inhabilitado en el prototipo por ser un valor que muy pocos fabricantes proporcionan en las características de sus fibras.



Factor de acoplamiento entre modos

Longitud de fábrica de la fibra óptica [Km]

Figura A.33. Factor de acoplamiento entre modos y Longitud de fábrica de la fibra

Los botones de la interfaz Fibra_Optica permiten, como su nombre lo indica cargar los datos a la interfaz principal la Enlace_Optico y el botón borrar a su vez permite eliminar todos los datos configurados para establecer otro tipo de fibra óptica con su respectivo estándar y sus características.



Cargar

Borrar

Figura A.34. Botones Borrar y Cargar

2.3.- Receptor

Al dar clic en el botón Receptor, se abre la interfaz Receptor para su respectiva configuración como se muestra en la figura.



Receptor

DATOS DE RECEPTOR

Sitio B

Tipo de receptor

Fabricante/modelo

Margen de seguridad [dBm]

Sensibilidad [dBm]

Tiempo de subida [ns]

Borrar

Cargar

Figura A.35. Interfaz Receptor

En la parte inicial de la interfaz Receptor se configura el Sitio B o lugar donde finaliza la instalación de fibra óptica, además del tipo de receptor.

The image shows a horizontal interface with two main sections. On the left, there is a text input field labeled 'Sitio B'. On the right, there is a dropdown menu labeled 'Tipo de receptor' with the text 'Seleccione...' and a downward arrow.

Figura A.36. Sitio B y Tipo de receptor

Al habilitar el pop-up menú del tipo de receptor, se puede seleccionar entre un receptor PIN, un diodo de avalancha o un fotodiodo común.

The image shows a dropdown menu titled 'Tipo de receptor'. The menu is open, showing a list of options: 'Seleccione...' (highlighted in blue), 'PIN', 'Diodo avalancha', and 'Fotodiodo'.

Figura A.37. Tipo de receptor

En la parte siguiente de la interfaz Receptor se configura el fabricante o modelo del receptor y el margen de seguridad, este último dato es muy importante en las especificaciones del receptor porque permite determinar la factibilidad del enlace según su comparación con el presupuesto de potencia y la atenuación que llega al receptor. El valor del fabricante es tipo string y del margen de seguridad es un valor numérico.

The image shows two text input fields side-by-side. The left field is labeled 'Fabricante/modelo' and the right field is labeled 'Margen de seguridad [dBm]'.

Figura A.38. Fabricante/modelo y Margen de seguridad

En la parte final de la interfaz Receptor se configura la sensibilidad en dBm y el tiempo de subida en ns, información consecuente a la gestión de la instalación de fibra óptica.

The image shows two text input fields side-by-side. The left field is labeled 'Sensibilidad [dBm]' and the right field is labeled 'Tiempo de subida [ns]'.

Figura A.39. Sensibilidad y tiempo de subida

Los botones de la interfaz Receptor permiten, como su nombre lo indica cargar los datos a la interfaz principal la Enlace_Optico y el botón borrar a su vez permite eliminar todos los datos configurados para establecer otro tipo de configuración.

The image shows two buttons side-by-side. The left button is labeled 'Borrar' and the right button is labeled 'Cargar'.

Figura A.40. Botones Borrar y Cargar

3.- Bloque Empalmes y Conectores

En este bloque de la interfaz Enlace_Optico, se encuentran componentes para configurar los empalmes y conectores utilizados, de los cuales destacan el tipo de conector, tipo de empalme y número de empalmes con su respectiva distancia de ubicación en el enlace óptico

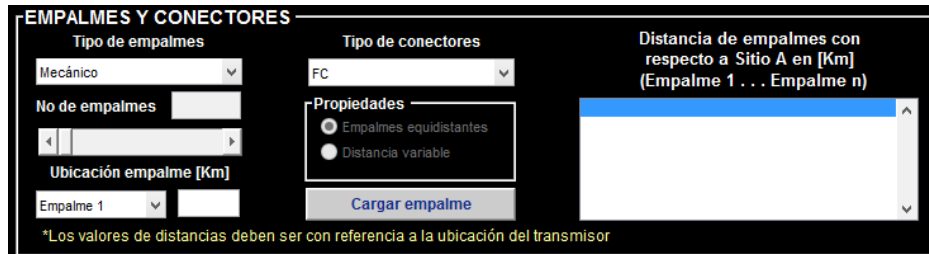


Figura A.41. Bloque Empalmes y Conectores

Al habilitar el pop-up menú del tipo de empalmes, se puede seleccionar entre empalme mecánico y empalme de fusión.

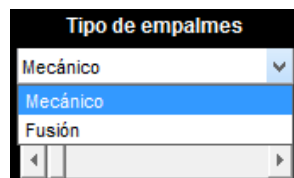


Figura A.42. Tipo empalmes

Al habilitar el pop-up menú del tipo de conectores, se puede seleccionar entre siete conectores de los más utilizados en el mercado ecuatoriano.



Figura A.43. Tipo de conectores

Al deslizar la barra espaciadora me permite configurar el número de empalmes utilizados en el enlace óptico simulado, el mismo que está configurado como máximo hasta diez, cantidad muy elevada y de poco uso en instalaciones reales, debido a que

el uso de demasiados empalmes provoca demasiada atenuación al enlace y toda la información transmitida se perdería por completo en el canal de comunicación.



Figura A.44. Número de empalmes

Las propiedades de los empalmes me permite configurar si los empalmes seleccionados son equidistantes o de distancia variable.

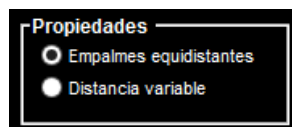


Figura A.45. Propiedades empalmes

La ubicación del empalme es muy importante para la configuración de la distancia en donde se encuentra el empalme; y según sea solicitado según distancia variable o equidistantes. Si es distancia variable, la distancia de cada empalme será variada, en cambio con empalmes equidistantes la distancia a la que se encuentran los empalmes son semejantes.



Figura A.46. Ubicación empalme

La figura muestra el botón cargar empalme con lo que se puede acoplar la información de cada empalme y de la instalación óptica para un pronto análisis.



Figura A.47. Cargar empalme

La figura muestra los parámetros establecidos en la configuración del número de empalmes y la distancia a la que se encuentra del sitio A. los valores configurados de distancia de los empalmes se puede reflejar en el listbox

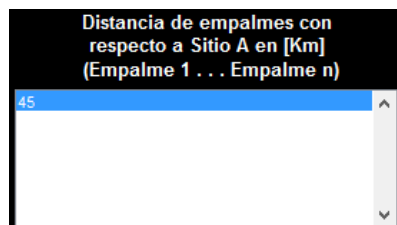


Figura A.48. Ubicación empalme

4.- Bloque Resultados

En este bloque de la interfaz Enlace_Optico, se caracteriza por tener seis botones con los cuales se accede a diferentes interfaces de cumplimientos específicos



Figura A.49. Bloque Resultados

Al dar clic en el botón del simulador SOTDR, se habilita una interfaz donde se refleja en forma predominante la traza reflectométrica necesaria para que el usuario gestione o no la instalación de fibra óptica.

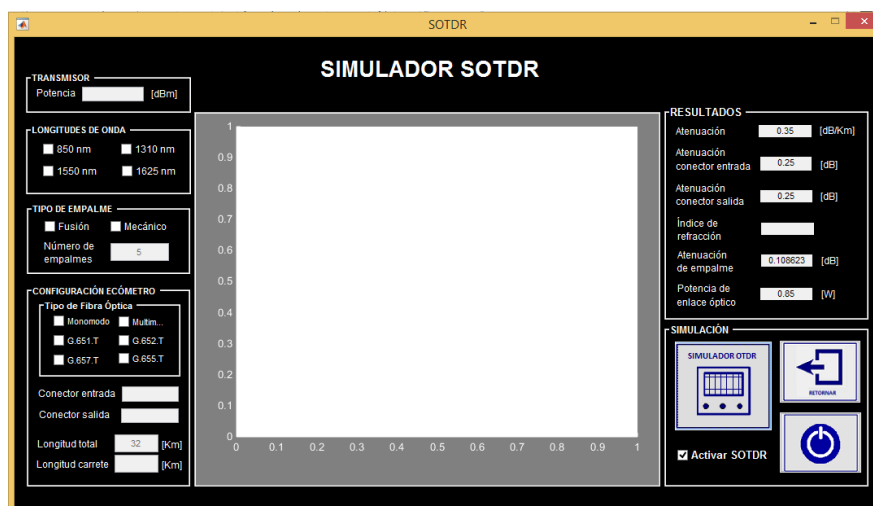


Figura A.50. Simulador SOTDR

Cada componente que integra esta interfaz SOTDR esta inhabilitada, debido a que son valores heredados y/o calculados de las anteriores configuraciones o interfaces; a excepción de los botones del bloque simulación de la interfaz SOTDR; que son

utilizados para obtener la gráfica de la traza reflectométrica; el botón retornar hace que la interface portada se ejecute de nuevo para empezar nuevamente desde cero. Y por último el botón salir, el mismo que habilita un cuadro de diálogo para asegurar o negar la salida del simulador SOTDR; como se muestra en la figura.



Figura A.51. Botones interfaz SOTDR.

Los botones que muestran la figura permiten acceder a cada interfaz respectivamente, las interfaces presupuesto y la interfaz SOTDR.



Figura A.52. Botones Bloque Resultados.

Al dar clic en el botón presupuesto de potencia y atenuación, se habilita la interfaz Presupuesto_Potencia, como se muestra en la figura. .



Figura A.53. Interfaz Presupuesto de Potencia

Los botones que muestran la figura permiten avanzar en el simulador a la interfaz SOTDR o retornar a una interfaz previa del software, en este caso la portada.

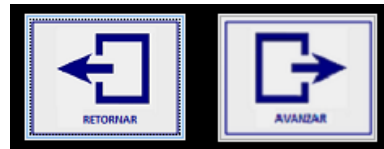


Figura A.54. Botones Retornar y Avanzar

Por otro lado, en el Bloque Resultados de la interfaz Enlace_Optico se dispone de los botones Ayuda y Salir respectivamente como se muestra en la figura.



Figura A.55. Botones Ayuda y Salir.

Al dar un clic en el botón Ayuda se ejecuta la interfaz Ayuda como muestra la figura, en la cual se detalla un extracto del resumen y la funcionalidad del prototipo, los autores y el respectivo copyright. La interfaz dispone de un botón para retornar a la interfaz Portada y empezar todo el proceso de simulación nuevamente, para otro caso de enlace óptico.



Figura A.56. Interfaz Ayuda

ANEXO B: Código de Programación

Anexo B.1: Interfaz gráfica de usuario Portada.fig

Los componentes de la GUI Portada.fig se muestran en la Figura B.1.



Figura B.1. Interfaz gráfica de usuario Portada.fig

Anexo B.2: Código de programación Portada.m

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

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

% Last Modified by GUIDE v2.5 05-Feb-2018 18:34:05

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

```

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Portada is made visible.
function Portada_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Portada (see VARARGIN)
background=imread('C:\SOTDR\Imagenes\uta.jpg');
axes(handles.background);
axis off;
imshow(background);
background1=imread('C:\SOTDR\Imagenes\fisei.jpg');
axes(handles.background1);
axis off;
imshow(background1);

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

% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%PRIMERO CERRAMOS EL GUI_Inicial
close(gcf)
%ABRIMOS EL GUI Enlace Optico
Enlace_Optico

```

Anexo B.3: Interfaz gráfica de usuario Enlace_Optico.fig

Los componentes de la GUI Enlace_Optico.fig se muestran en la Figura B.2.



Figura B.2. Interfaz gráfica de usuario Enlace_Optico.fig

Anexo B.4: Código de programación Enlace_Optico.m

```
function varargout = Enlace_Optico(varargin)
% ENLACE_OPTICO MATLAB code for Enlace_Optico.fig
% ENLACE_OPTICO, by itself, creates a new ENLACE_OPTICO or raises the existing
% singleton*.
%
% H = ENLACE_OPTICO returns the handle to a new ENLACE_OPTICO or the handle to
% the existing singleton*.
%
% ENLACE_OPTICO('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in ENLACE_OPTICO.M with the given input arguments.
%
% ENLACE_OPTICO('Property','Value',...) creates a new ENLACE_OPTICO or raises
% the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before Enlace_Optico_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to Enlace_Optico_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help Enlace_Optico
%
% Last Modified by GUIDE v2.5 22-Feb-2018 16:10:25
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Enlace_Optico_OpeningFcn, ...
                  'gui_OutputFcn',  @Enlace_Optico_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Enlace_Optico is made visible.
function Enlace_Optico_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Enlace_Optico (see VARARGIN)
%Coloca una imagen en cada botón
%Obtengo el BIT RATE de la GUI Transmisor y la coloco en la tasa de TX
global pdea31
set(handles.edit6,'String',pdea31);
%Deshabilito los radiobutton de empalmes equidistantes o distancia variable
set(handles.radiobutton2, 'Enable', 'off');
set(handles.radiobutton3, 'Enable', 'off');
%Deshabilito el numero de empalmes que funciona con el slider
set(handles.edit2,'Enable','off');
%Deshabilito el edit de la longitud del enlace
set(handles.edit1,'Enable','off');
%Deshabilito el edit de la tasa de transmisión porque ese dato viene del TX
set(handles.edit6,'Enable','off');

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

% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
global ventana
global inf
global longitudonda

```



```

inf=get(handles.listbox1,'Value');
switch inf;case 1;longitudonda = '850';ventana=str2double(longitudonda);case
2;longitudonda = '1310';ventana=str2double(longitudonda);...
    case 3;longitudonda = '1550';ventana=str2double(longitudonda);case 4;longitudonda
= '1625';ventana=str2double(longitudonda);
end
Transmisor;
% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on slider movement.
function slider_distancia_Callback(hObject, eventdata, handles)
% hObject    handle to slider_distancia (see GCBO)
%DECLARO UNA VARIABLE GLOBAL DISTANCIA DEL ENLACE
global longitud_total
%DOY VALORES A LA BARRA DESLIZADORA
handles.slider_distancia=get(hObject,'Value');
handles.slider_distancia=100.*handles.slider_distancia;
set(handles.edit1,'String',handles.slider_distancia);
longitud_total=handles.slider_distancia;

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)

% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

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

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

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

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton5 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton6 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

function edit2_Callback(hObject, eventdata, handles)
% hObject     handle to edit2 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit2 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

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

function edit3_Callback(hObject, eventdata, handles)
% hObject     handle to edit3 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit3 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in listbox3.
function listbox3_Callback(hObject, eventdata, handles)
% hObject     handle to listbox3 (see GCBO)

```

```

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

% Hints: contents = cellstr(get(hObject,'String')) returns listBox3 contents as cell
array
% contents(get(hObject,'Value')) returns selected item from listBox3

% --- Executes during object creation, after setting all properties.
function listBox3_CreateFcn(hObject, eventdata, handles)
% hObject handle to listBox3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

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

function edit4_Callback(hObject, eventdata, handles)
% hObject handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
% str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
global empalme1
global empalme2
global empalme3
global empalme4
global empalme5
global empalme6
global empalme7
global empalme8
global empalme9
global empalme10 empalmesn
global vttr f1 f2 f3 f4 f5 f6 f7 f8 f9 f10
f = get(handles.edit4, 'String');%Distancia del empalme
e=get(handles.popupmenu3, 'Value');%Numero de empalmes
d = get(handles.edit1, 'string');%Longitud del enlace
f = str2double(f);d = str2double(d);%Convierto a numero los dos datos
if f == 0 %Compruebo que la distancia del empalme no sea cero
warndlg('Ubicación de empalme debe ser diferente de cero');
return
end
if f > d %Compruebo que la distancia del empalme no sea mayor a la del enlace
d = get(handles.edit1, 'string');
r = strcat('La ubicación del empalme debe ser menor a: ', d, ' Km');
g = warndlg(r);
else
switch e
case 1
io = get(handles.edit4, 'String');io = str2double(io);
set(handles.listBox4, 'String', io); f1 = io;
empalme1 = io;
empalmesn = 1;
case 2

```

```

        f2 = f;
        vrfw = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        nk = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vrfw; nk]);
        empalme2 = str2double(nk);
        empalmesn = 2;
    case 3
        f3=f;
        vrf = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        n = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vrf; n]);
        empalme3 = str2double(n);
        empalmesn = 3;
    case 4
        f4=f;
        vrff = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        n1 = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vrff; n1]);
        empalme4 = str2double(n1);
        empalmesn = 4;
    case 5
        f5=f;
        vrff1 = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        nr = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vrff1; nr]);
        empalme5 = str2double(nr);
        empalmesn = 5;
    case 6
        f6=f;
        vrffg = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        n14 = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vrffg; n14]);
        empalme6 = str2double(n14);
        empalmesn = 6;
    case 7
        f7= f;
        vr4f = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantodad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        n13 = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vr4f; n13]);
        empalme7 = str2double(n13);
        empalmesn = 7;
    case 8
        f8 = f;
        vwrff = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        n1 = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vwrff; n1]);
        empalme8 = str2double(n1);
        empalmesn = 8;
    case 9
        f9=f;
        vr5f = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');

```

```

        ert = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vr5f; ert]);
        empalme9 = str2double(ert);
        empalmesn = 9;
    case 10
        f10=f;
        vrqff = get(handles.listbox4, 'String');
        set(handles.listbox4, 'String', 'Cargar la misma cantidad de digitos para
cada empalme para que argumentos de matriz sean consistentes.Empezar desde empalme
1');
        nlr = get(handles.edit4, 'String');
        set(handles.listbox4, 'String', [vrqff; nlr]);
        empalme10 = str2double(nlr);
        empalmesn = 10;
    end
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
btnsalir=questdlg('¿Está seguro de salir de la
aplicación?', 'SOTDR_v1.0', 'Si', 'No', 'Cancelar');
switch btnsalir
    case 'Si',
        close(gcf);
    case 'No'
        gcbf
    case 'Cancelar'
        gcbf
end

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ABRIMOS EL GUI AYUDA
SOTDR
%LINEA DE CODIGO INDISPENSABLE PARA RECUPERAR LOS DATOS
guidata(hObject, handles);

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ABRIMOS EL GUI AYUDA
Portada
%LINEA DE CODIGO INDISPENSABLE PARA RECUPERAR LOS DATOS
guidata(hObject, handles);

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ABRIMOS EL GUI AYUDA
Ayuda
%LINEA DE CODIGO INDISPENSABLE PARA RECUPERAR LOS DATOS
guidata(hObject, handles);

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

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

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
global tipoempalme
global tipoempalmenum
global valxya
global pdea14
valxya=get(handles.popupmenu1,'Value');
switch valxya;case 1;pdea14 = 'Mecánico';tipoempalmenum=1;case 2;pdea14 =
'Fusión';tipoempalmenum=2;
end
tipoempalme=pdea14;

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
global tipoconector
global valxza
global pdea20
valxza=get(handles.popupmenu2,'Value');
switch valxza;case 1;pdea20 = 'FC';case 2;pdea20 = 'LC';case 3;pdea20 = 'MT-
Array';case 4;pdea20 = 'SC';...
    case 5;pdea20 = 'SC-Duplex';case 6;pdea20 = 'ST';case 7;pdea20 = 'BFOC';
end
tipoconector=pdea20;

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
%DECLARO UNA VARIABLE GLOBAL NUMERO DE EMPALMES
global numeroempalme
%Doy valores a la barra deslizador a 10 empalme como máximo, si el enlace
%tiene demasiados empalme, la atenuación es excesiva y el enlace no es
%factible
handles.slider3=get(hObject,'Value');
handles.slider3=(100).*(handles.slider3);
set(handles.edit2,'String',handles.slider3);
numeroempalme=handles.slider3;
%Habilito los radiobutton de empalme equidistantes o distancia variable
set(handles.radiobutton2, 'Enable', 'on');
```

```

set(handles.radiobutton3, 'Enable', 'on');

% --- Executes during object creation, after setting all properties.
function slider3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
global ventana
global inf
global longitudonda
inf=get(handles.popupmenu4,'Value');
switch inf;case 1;longitudonda = '850';ventana=str2double(longitudonda);case
2;longitudonda = '1310';ventana=str2double(longitudonda);...
    case 3;longitudonda = '1550';ventana=str2double(longitudonda);case 4;longitudonda
= '1625';ventana=str2double(longitudonda);
end
Transmisor;

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

```

```

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

% --- Executes on selection change in listbox4.
function listbox4_Callback(hObject, eventdata, handles)
% hObject    handle to listbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox4 contents as cell
array
%          contents{get(hObject,'Value')} returns selected item from listbox4

% --- Executes during object creation, after setting all properties.
function listbox4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
global empalme1
global empalme2
global empalme3
global empalme4
global empalme5
global empalme6
global empalme7
global empalme8
global empalme9
global empalme10
%Deshabilito el radiobutton3 de la distancia variable para empalmes
set(handles.radiobutton3, 'Enable', 'off');
%Deshabilito el slider3 para ya no modificar (aumentar o disminuir)el numero de
empalmes
set(handles.slider3, 'Enable', 'off');
we = get(handles.slider3, 'Value');
dist = get(handles.edit4, 'String');
distancia = str2double(dist);
separacion = distancia/(we+1);
emp1 = separacion*1;
emp2 = separacion*2;
emp3 = separacion*3;
emp4 = separacion*4;
emp5 = separacion*5;
emp6 = separacion*6;
emp7 = separacion*7;
emp8 = separacion*8;
emp9 = separacion*9;
emp10 = separacion*10;
e1 = num2str(emp1);
e2 = num2str(emp2);
e3 = num2str(emp3);
e4 = num2str(emp4);
e5 = num2str(emp5);
e6 = num2str(emp6);
e7 = num2str(emp7);
e8 = num2str(emp8);
e9 = num2str(emp9);
e10 = num2str(emp10);
switch we
    case 1
        set(handles.listbox4, 'String', {e1});
    case 2
        set(handles.listbox4, 'String', {e1 e2});
    case 3
        set(handles.listbox4, 'String', {e1 e2 e3});

```



```

case 4
    set(handles.listbox4, 'String', {e1 e2 e3 e4});
case 5
    set(handles.listbox4, 'String', {e1 e2 e3 e4 e5});
case 6
    set(handles.listbox4, 'String', {e1 e2 e3 e4 e5 e6});
case 7
    set(handles.listbox4, 'String', {e1 e2 e3 e4 e5 e6 e7});
case 8
    set(handles.listbox4, 'String', {e1 e2 e3 e4 e5 e6 e7 e8});
case 9
    set(handles.listbox4, 'String', {e1 e2 e3 e4 e5 e6 e7 e8 e9});
case 10
    set(handles.listbox4, 'String', {e1 e2 e3 e4 e5 e6 e7 e8 e9 e10});
end
empalme1 = emp1;
empalme2 = emp2;
empalme3 = emp3;
empalme4 = emp4;
empalme5 = emp5;
empalme6 = emp6;
empalme7 = emp7;
empalme8 = emp8;
empalme9 = emp9;
empalme10 = emp10;

% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
%Deshabilito el radiobutton de distancia equidistantes entre empalmes
set(handles.radiobutton2, 'Enable', 'off');
%Deshabilito el slider3 para ya no modificar (aumentar o disminuir)el numero de
empalmes
set(handles.slider3, 'Enable', 'off');

```

Anexo B.5: Interfaz gráfica de usuario Transmisor.fig

Los componentes de la GUI Transmisor.fig se muestran en la Figura B.3.



Figura B.3. Interfaz gráfica de usuario Transmisor.fig

Anexo B.6: Código de programación Transmisor.m

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

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

% Last Modified by GUIDE v2.5 20-Feb-2018 18:37:42

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

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

```

    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

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

% Choose default command line output for Transmisor
handles.output = hObject;
%Deshabilito el edit5 de la longitud de onda
set(handles.edit5,'Enable','off');
%Deshabilito el edit6 del BIT RATE
set(handles.edit6,'Enable','off');
%Recibo el dato de la longitud de onda o ventana de TX
global ventana
set(handles.edit5,'String',ventana);
% Update handles structure
guidata(hObject, handles);

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

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

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

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
global sitioa
sitioa=get(handles.edit1,'string');

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
global fabricamodelorx
fabricamodelorx=get(handles.edit2,'string');

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)

```

```

global potenciatx
e=get(handles.edit3,'string');
potenciatx=str2double(e);
%Comprobar si es un numero edit3
if isnan(potenciatx)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit3,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
global tiemposubidatx
f=get(handles.edit4,'string');
tiemposubidatx=str2double(f);
%Comprobar si es un numero edit5
if isnan(tiemposubidatx)
    errordlg('El valor debe ser numérico','ERROR');
    set(handles.edit4,'String','0');
end

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
global tipodetx
global valza
global pdea8
valza=get(handles.popupmenu1,'Value');
switch valza;case 2;pdea8 = 'Led';case 3;pdea8 = 'Laser';
end
if (valza==1)
    errordlg('Seleccione un tipo de Transmisor','ERROR')
else
    tipodetx=pdea8;
end

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

```

```

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
global bitrate
global bitratenum
global qazwa
global pdea31
qazwa=get(handles.popupmenu2,'Value');
switch qazwa;case 2;pdea31 = '500 Kbps';bitratenum=500;case 3;pdea31 = '1
Mbps';bitratenum=1;case 4;pdea31='3 Mbps';bitratenum=3;case 5;pdea31='5
Mbps';bitratenum=5;...
    case 6;pdea31 = '51.84 Mbps';bitratenum=51.84;case 7;pdea31 = '155.52
Mbps';bitratenum=155.52;case 8;pdea31='622.08 Mbps';bitratenum=622.08;case
9;pdea31='1244.16 Mbps';bitratenum=1244.16;...
    case 10;pdea31 = '2588.32 Mbps';bitratenum=2588.32;case 11;pdea31 = '9953.28
Mbps';bitratenum=9953.28;case 12;pdea31='39814.32 Mbps';bitratenum=39814.32;
end
if (qazwa==1)
    errordlg('Seleccione un tipo de Bit Rate','ERROR')
else
    bitrate=pdea31;
    bitratenum=str2double(num2str(bitratenum));
    set(handles.edit6,'String',bitratenum);
end

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
global modulacion
global qaza
global pdea30
qaza=get(handles.popupmenu3,'Value');
switch qaza;case 2;pdea30 = 'NRZ';case 3;pdea30 = 'RZ';...
    case 4;pdea30 = 'Manchester';
end
if (qaza==1)
    errordlg('Seleccione un tipo de Modulaci3n','ERROR')
else
    modulacion=pdea30;
end

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)

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

```

```

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

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%       str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
global anchoespectral
f=get(handles.edit7,'string');
anchoespectral=str2double(f);
%Comprobar si es un numero edit5
if isnan(anchoespectral)
    errorldg('El valor debe ser numérico','ERROR');
    set(handles.edit7,'String','0');
end

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
%Habilito los componentes
set(handles.edit1,'Enable','on');
set(handles.popupmenu1,'Enable','on');
set(handles.edit2,'Enable','on');
set(handles.popupmenu2,'Enable','on');
set(handles.edit3,'Enable','on');
set(handles.popupmenu3,'Enable','on');
set(handles.edit4,'Enable','on');
set(handles.edit7,'Enable','on');
%Asigno espacio vacío en los componentes
set(handles.edit1,'String',' ');
set(handles.popupmenu1,'Value',1);
set(handles.edit2,'String',' ');
set(handles.popupmenu2,'Value',1);
set(handles.edit3,'String',' ');
set(handles.popupmenu3,'Value',1);
set(handles.edit4,'String',' ');
set(handles.edit7,'String',' ');

```

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% Compruebo que no este vacio ningun campo
if ((strcmp(handles.edit1.String, '
')) || (handles.popupmenu1.Value==1) || (strcmp(handles.edit2.String, '
')) || (handles.popupmenu2.Value==1) || (strcmp(handles.edit3.String, '
')) || (handles.popupmenu3.Value==1) || (strcmp(handles.edit4.String, '
')) || (strcmp(handles.edit7.String, ' ')))
    warndlg('Por favor todos los campos deben estar llenos','!! WARNING !!');

if((handles.popupmenu1.Value==1) || (handles.popupmenu2.Value==1) || (handles.popupmenu3.
Value==1))
    helpdlg('Revise el tipo de Transmisor, Bit Rate y/o Modulación','SOTDR v1.0');
end
else
    %barra de espera
    h= waitbar(0,'SIMULADOR OTDR','Name','SOTDR v 1.0:
CARGANDO!!!','Color','White','Units','Pixels','WindowStyle','modal');
    for i=1:1000,
        waitbar(i/1000)
    end
    close(h)
    %fin barra de espera
    %bloqueo los componentes
    set(handles.edit1,'Enable','off');
    set(handles.popupmenu1,'Enable','off');
    set(handles.edit2,'Enable','off');
    set(handles.popupmenu2,'Enable','off');
    set(handles.edit3,'Enable','off');
    set(handles.popupmenu3,'Enable','off');
    set(handles.edit4,'Enable','off');
    set(handles.edit7,'Enable','off');
end

```

Anexo B.7: Interfaz gráfica de usuario Fibra_Optica.fig

Los componentes de la GUI Fibra_Optica.fig se muestran en la Figura B.4.

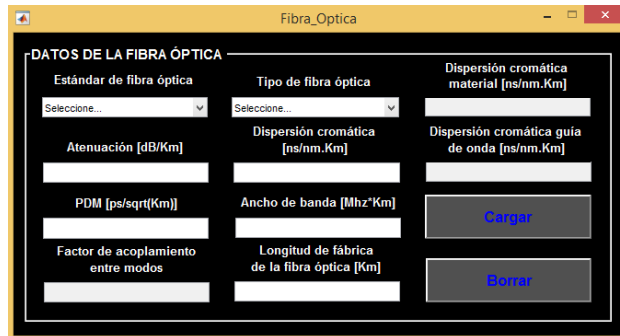


Figura B.4. Interfaz gráfica de usuario Fibra_Optica.fig

Anexo B.8: Código de programación Fibra_Optica.m

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

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

% Last Modified by GUIDE v2.5 21-Feb-2018 12:39:04

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

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```



```

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

% Choose default command line output for Fibra_Optica
handles.output = hObject;
%Deshabilito componentes no utiles para el cálculo de la potencia total y
%atenuacion total
%Deshabilito el edit3 de la Dispersion cromatica del material
set(handles.edit3,'Enable','off');
%Deshabilito el edit6 de la Dispersion cromatica guia de onda
set(handles.edit6,'Enable','off');
%Deshabilito el edit7 del factor de acoplamiento entre modos
set(handles.edit7,'Enable','off');
% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
global estandarfibra
global valxa
global pdea6
valxa=get(handles.popupmenu1,'Value');
switch valxa;case 2;pdea6 = 'UIT-T G.651';case 3;pdea6 = 'UIT-T G.652A';case 4;pdea6
= 'UIT-T G.652B';case 5;pdea6 = 'UIT-T G.652C';...
        case 6;pdea6 = 'UIT-T G.652D';case 7;pdea6 = 'UIT-T G.653';case 8;pdea6 =
'UIT-T G.654';case 9;pdea6 = 'UIT-T G.655';...
        case 10;pdea6 = 'UIT-T G.656';case 11;pdea6 = 'UIT-T G.657';case 12;pdea6
= 'Sin estándar';
end
if (valxa==1)
    errordlg('Seleccione un tipo de Estándar de fibra óptica','ERROR')
else
    estandarfibra=pdea6;
end

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
global tipodefo
global valyza
global pdea22
valyza=get(handles.popupmenu2,'Value');
switch valyza;case 2;pdea22 = 'Monomodo';case 3;pdea22 = 'Multimodo';

```

```

end
if (valyza==1)
    errordlg('Seleccione un tipo de Fibra Óptica','ERROR')
else
    tipodefo=pdea22;
end

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
global atenuacionfo
e=get(handles.edit1,'string');
atenuacionfo=str2double(e);
%Comprobar si es un numero edit1
if isnan(atenuacionfo)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit1,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
global dispersioncromatica
g=get(handles.edit2,'string');
dispersioncromatica=str2double(g);
%Comprobar si es un numero edit3
if isnan(dispersioncromatica)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit2,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
global dispersioncromaterial
f=get(handles.edit3,'string');
dispersioncromaterial=str2double(f);
%Comprobar si es un numero edit3
if isnan(dispersioncromaterial)

```

```

        errordlg('El valor debe ser numérico','ERROR')
        set(handles.edit3,'String','0')
    end

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
global PDM
p=get(handles.edit4,'string');
PDM=str2double(p);
%Comprobar si es un numero edit3
if isnan(PDM)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit4,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
global anchodebanda
q=get(handles.edit5,'string');
anchodebanda=str2double(q);
%Comprobar si es un numero edit3
if isnan(anchodebanda)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit5,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
global dispersioncroguiaonda
h=get(handles.edit6,'string');
dispersioncroguiaonda=str2double(h);
%Comprobar si es un numero edit3
if isnan(dispersioncroguiaonda)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit6,'String','0')
end
end

```

```

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
global factoracoplamiento
r=get(handles.edit7,'string');
factoracoplamiento=str2double(r);
%Comprobar si es un numero edit3
if isnan(factoracoplamiento)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit7,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
global longitudfabrica
s=get(handles.edit8,'string');
longitudfabrica=str2double(s);
%Comprobar si es un numero edit3
if isnan(longitudfabrica)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit8,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
%Compruebo que no este vacio ningun campo
if ((strcmp(handles.edit1.String, '
')) || (handles.popupmenu1.Value==1) || (strcmp(handles.edit2.String, '
')) || (handles.popupmenu2.Value==1) || (strcmp(handles.edit4.String, '
')) || (strcmp(handles.edit5.String, ' ')) || (strcmp(handles.edit8.String, ' ')))
    warndlg('Por favor todos los campos deben estar llenos','!! WARNING !!');
    if((handles.popupmenu1.Value==1) || (handles.popupmenu2.Value==1))
        helpdlg('Revice el Estándar y el Tipo de fibra óptica','SOTDR v1.0');
    end
else

```

```

%barra de espera
h= waitbar(0,'SIMULADOR OTDR','Name','SOTDR v 1.0:
CARGANDO!!!','Color','White','Units','Pixels','WindowStyle','modal');
for i=1:1000,
    waitbar(i/1000)
end
close(h)
%fin barra de espera
%bloqueo los componentes
set(handles.edit1,'Enable','off');
set(handles.popupmenu1,'Enable','off');
set(handles.edit2,'Enable','off');
set(handles.popupmenu2,'Enable','off');
set(handles.edit4,'Enable','off');
set(handles.edit5,'Enable','off');
set(handles.edit8,'Enable','off');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
%Habilito los componentes
set(handles.popupmenu1,'Enable','on');
set(handles.popupmenu2,'Enable','on');
set(handles.edit1,'Enable','on');
set(handles.edit2,'Enable','on');
set(handles.edit4,'Enable','on');
set(handles.edit5,'Enable','on');
set(handles.edit8,'Enable','on');
%Asigno espacio vacio en los componentes
set(handles.popupmenu1,'Value',1);
set(handles.popupmenu2,'Value',1);
set(handles.edit1,'String',' ');
set(handles.edit2,'String',' ');
set(handles.edit4,'String',' ');
set(handles.edit5,'String',' ');
set(handles.edit8,'String',' ');

```

Anexo B.9: Interfaz gráfica de usuario Receptor.fig

Los componentes de la GUI Receptor.fig se muestran en la Figura B.5.



Figura B.5. Interfaz gráfica de usuario Receptor.fig

Anexo B.10: Código de programación Receptor.m

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

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

% Last Modified by GUIDE v2.5 19-Feb-2018 10:15:46

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

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Receptor is made visible.
```

```

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

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

% Update handles structure
guidata(hObject, handles);

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

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

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

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
global sitiob
sitiob=get(handles.edit1,'string');

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
global tipoderx
global valya
global pdea7
valya=get(handles.popupmenu1,'Value');
switch valya;case 2;pdea7 = 'PIN';case 3;pdea7 = 'Diodo avalancha';...
    case 4;pdea7 = 'Fotodiodo';
end
if (valya==1)
    errorldg('Seleccione un tipo de Receptor','ERROR')
else
    tipoderx=pdea7;
end

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)

```

```

global fabricamodelorx
fabricamodelorx=get(handles.edit1,'string');

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
global margen
e=get(handles.edit3,'string');
margen=str2double(e);
%Comprobar si es un numero edit3
if isnan(margen)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit3,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
global sensibilidad
d=get(handles.edit4,'string');
sensibilidad=str2double(d);
%Comprobar si es un número edi4
if isnan(sensibilidad)
    errordlg('El valor debe ser numérico','ERROR')
    set(handles.edit4,'String','0')
end

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
global tiemposubidarx
f=get(handles.edit5,'string');
tiemposubidarx=str2double(f);
%Comprobar si es un numero edit5
if isnan(tiemposubidarx)
    errordlg('El valor debe ser numérico','ERROR');
    set(handles.edit5,'String','0');
end

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

```



```

function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global tipoderx
%Muestro el tipo de receptor seleccionado
set(handles.text3,'String',tipoderx);
%Compruebo que no este vacio ningun campo
if ((strcmp(handles.edit1.String, '
'))||(handles.popupmenu1.Value==1)|| (strcmp(handles.edit2.String, '
'))||(strcmp(handles.edit3.String, ' '))||(strcmp(handles.edit4.String, '
'))||(strcmp(handles.edit5.String, ' ')))
    warndlg('Por favor todos los campos deben estar llenos','!! WARNING !!');
    if(handles.popupmenu1.Value==1)
        helpdlg('Revice el tipo de Receptor','SOTDR v1.0');
    end
else
    %barra de espera
    h= waitbar(0,'SIMULADOR OTDR','Name','SOTDR v 1.0:
CARGANDO!!!','Color','White','Units','Pixels','WindowStyle','modal');
    for i=1:1000,
        waitbar(i/1000)
    end
    close(h)
    %fin barra de espera
    %bloqueo los componentes
    set(handles.edit1,'Enable','off');
    set(handles.popupmenu1,'Enable','off');
    set(handles.edit2,'Enable','off');
    set(handles.edit3,'Enable','off');
    set(handles.edit4,'Enable','off');
    set(handles.edit5,'Enable','off');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
%Habilito los componentes
set(handles.edit1,'Enable','on');
set(handles.popupmenu1,'Enable','on');
set(handles.edit2,'Enable','on');
set(handles.edit3,'Enable','on');
set(handles.edit4,'Enable','on');
set(handles.edit5,'Enable','on');
%Asigno espacio vacio en los comònetes
set(handles.edit1,'String',' ');
set(handles.popupmenu1,'Value',1);
set(handles.edit2,'String',' ');
set(handles.edit3,'String',' ');
set(handles.edit4,'String',' ');
set(handles.edit5,'String',' ');

```

Anexo B.11: Interfaz gráfica de usuario SOTDR.fig

Los componentes de la GUI SOTDR.fig se muestran en la Figura B.6.

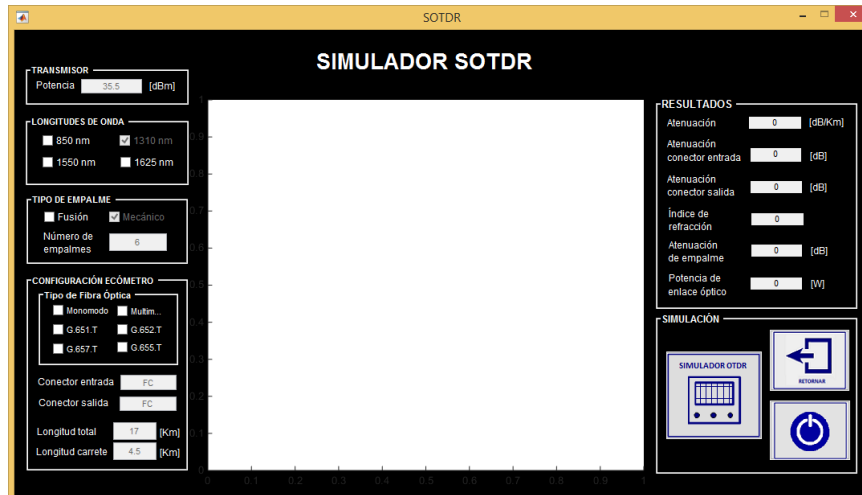


Figura B.6. Interfaz gráfica de usuario SOTDR.fig

Anexo B.12: Código de programación SOTDR.m

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

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

% Last Modified by GUIDE v2.5 05-Feb-2018 14:48:44

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

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before SOTDR is made visible.
function SOTDR_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to SOTDR (see VARARGIN)
% Choose default command line output for SOTDR
handles.output = hObject;
%RECIBO LOS VALORES DE LOS FORMULARIOS ANTERIORES
%LONGITUD DE ONDA*****
%Obtengo la longitud de fabrica de la fibra óptica
global longitudfabrica
global longitud_total
global potenciadx
global ventana
global numeroempalmes
global tipoempalmenum
global tipoconector
%Obtengo la potencia del transmisor
set(handles.edit3,'String',potenciadx);
%Deshabilito la potencia del transmisor
set(handles.edit3,'Enable','off');
%Habilito la longitud de onda seleccionada en la GUI Enlace_Optico
if(ventana==850)
    set(handles.checkbox1,'Value',1);
    set(handles.checkbox1,'Enable','off');
end
if(ventana==1310)
    set(handles.checkbox2,'Value',1);
    set(handles.checkbox2,'Enable','off');
end
if(ventana==1550)
    set(handles.checkbox3,'Value',1);
    set(handles.checkbox3,'Enable','off');
end
if(ventana==1625)
    set(handles.checkbox4,'Value',1);
    set(handles.checkbox4,'Enable','off');
end
%Habilito el tipo de empalme seleccionado en la GUI Enlace_Optico
if(tipoempalmenum==1)
    set(handles.checkbox6,'Value',1);
    set(handles.checkbox6,'Enable','off');
end
if(tipoempalmenum==2)
    set(handles.checkbox5,'Value',1);
    set(handles.checkbox5,'Enable','off');
end

%Obtengo el tipo de conector entrada
set(handles.edit5,'String',tipoconector);
%Deshabilito el tipo de conector entrada
set(handles.edit5,'Enable','off');
%Obtengo el tipo de conector salida
set(handles.edit6,'String',tipoconector);
%Deshabilito el tipo de conector salida
set(handles.edit6,'Enable','off');

%Obtengo el numero de empalmes
set(handles.edit4,'String',numeroempalmes);
%Deshabilito el numero de empalmes
set(handles.edit4,'Enable','off');

%Obtengo la longitud de fabrica de la fibra óptica
set(handles.edit2,'String',longitudfabrica);
%Deshabilito la longitud de fabrica de la fibra óptica
set(handles.edit2,'Enable','off');

```

```

%Obtengo la longitud total del enlace
set(handles.edit1,'String',longitud_total);
%Deshabilito la longitud total del enlace
set(handles.edit1,'Enable','off');
%Deshabilito los radiobutton de empalmes equidistantes o distancia variable
%set(handles.radiobutton2, 'Enable', 'off');
%set(handles.radiobutton3, 'Enable', 'off');
%Deshabilito el numero de empalmes que funciona con el slider
%set(handles.edit2,'Enable','off');
%Deshabilito el edit de la longitud del enlace
%set(handles.edit1,'Enable','off');
%Deshabilito el edit de la tasa de transmisión porque ese dato viene del TX
%set(handles.edit6,'Enable','off');
% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on button press in checkbox5.
function checkbox5_Callback(hObject, eventdata, handles)
% hObject handle to checkbox5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox5

% --- Executes on button press in checkbox6.
function checkbox6_Callback(hObject, eventdata, handles)
% hObject handle to checkbox6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox6

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

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

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

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

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

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

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

```

```

% handles    structure with handles and user data (see GUIDATA)

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

% --- Executes on selection change in listBox1.
function listBox1_Callback(hObject, eventdata, handles)
% hObject    handle to listBox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listBox1 contents as cell
array
%          contents(get(hObject,'Value')) returns selected item from listBox1

% --- Executes during object creation, after setting all properties.
function listBox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listBox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in checkbox7.
function checkbox7_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox7

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

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

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

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

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

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on selection change in listBox2.
function listBox2_Callback(hObject, eventdata, handles)
% hObject    handle to listBox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listBox2 contents as cell
array
%           contents(get(hObject,'Value')) returns selected item from listBox2

% --- Executes during object creation, after setting all properties.
function listBox2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listBox2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

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

% --- Executes on selection change in listBox3.
function listBox3_Callback(hObject, eventdata, handles)
% hObject      handle to listBox3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listBox3 contents as cell
array
%           contents(get(hObject,'Value')) returns selected item from listBox3

% --- Executes during object creation, after setting all properties.
function listBox3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listBox3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

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

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%           str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

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

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%           str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

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

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

% --- Executes on button press in checkbox12.
function checkbox12_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox12

% --- Executes on button press in checkbox13.
function checkbox13_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox13

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%       str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%       str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%       str2double(get(hObject,'String')) returns contents of edit5 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ABRIMOS EL GUI AYUDA
Enlace_Optico
%LINEA DE CODIGO INDISPENSABLE PARA RECUPERAR LOS DATOS
guidata(hObject, handles);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
btnsalir=questdlg('¿Está seguro de salir de la
aplicación?', 'SOTDR_v1.0', 'Si', 'No', 'Cancelar');
switch btnsalir
    case 'Si',
        close(gcf);
    case 'No'
        gcbf
    case 'Cancelar'
        gcbf
end

```


Anexo B.13: Interfaz gráfica de usuario Ayuda.fig

Los componentes de la GUI Ayuda.fig se muestran en la Figura B.7.



Figura B.7. Interfaz gráfica de usuario Ayuda.fig

Anexo B.14: Código de programación Ayuda.m

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

% Copyright 2002-2003 The MathWorks, Inc.

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

% Last Modified by GUIDE v2.5 28-May-2008 09:36:18

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

```

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Ayuda is made visible.
function Ayuda_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Ayuda (see VARARGIN)
background=imread('C:\SOTDR\Imagenes\logo2.png');
axes(handles.background);
axis off;
imshow(background);
background1=imread('C:\SOTDR\Imagenes\fisei.jpg');
axes(handles.background1);
axis off;
imshow(background1);
% Choose default command line output for Ayuda
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

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

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

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%PRIMERO CERRAMOS EL GUI
close(gcf)
%ABRIMOS EL GUI Portada
Portada

```