



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“SOFTWARE DE APOYO PARA EL ESTUDIO
DE LOS EFECTOS ÓPTICOS EN UNA FIBRA”

TESIS

PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

PRESENTA:
ROLANDO NICOLÁS PINELO MALDONADO

ASESOR:
M.C. MARIBEL TELLO BELLO

HUAJUAPAN DE LEÓN, OAXACA

ABRIL DE 2003

DEDICATORIA

Dedico esta tesis a mi familia y en especial a mi hermana Caro.

Caro gracias por todo el apoyo que me brindaste en toda mi carrera , gracias por todo lo que sacrificaste por mi para hacer que yo cumpliera este objetivo, y gracias por demostrarme en todo momento tu valentía de enfrentar la vida jugando papeles tan difíciles como el de ser padre y madre a la vez, gracias a ti y a toda nuestra familia por apoyarme en varios aspectos y por confiar en mi.

Caro te quiero, te admiro y te respeto demasiado.

AGRADECIMIENTOS

M.C. Maribel Tello Bello

Por el gran apoyo tanto moral y académico que me brindó durante la realización de este proyecto.

A mis amigos

Ángel Sierra, Claudia, Nelba, Ekatherin, Daniel, Thelma y Olivia.

Gracias amigos por la gran amistad que me han brindado y por su apoyo incondicional en momentos difíciles en los que todos y cada uno de ustedes participaron de alguna u otra manera, ustedes son y serán siempre mis mejores amigos mis amigos de verdad.

A mis sinodales:

Dr. David Martínez Torres

M.C. Ramón Maldonado Basilio

M.C. Hilda Caballero Barbosa

M.C. Mario A. Moreno Rocha

Por su apoyo en partes importantes de esta tesis y su atención prestada sin ningún interés, así como también por servirme de pauta en la formación de un profesional con gran ética.

RESUMEN

El proyecto que se presenta a continuación ha sido desarrollado con la finalidad de aportar un software de apoyo al campo de la electrónica el cual ayude a personas interesadas a comprender el comportamiento de la luz que viaja a través de la fibra óptica. Consistiendo dicho software en el análisis de los principales fenómenos ópticos que intervienen en la propagación de la luz y el confinamiento de ésta en la fibra óptica. Además se incluye una sección en la cual se permiten realizar algunos cálculos técnicos que intervienen en la construcción de sistemas mínimos vía fibra óptica. Los análisis de los fenómenos ópticos y confinamiento de la luz en la fibra son presentados con gráficos animados tridimensionales.

Contenido

RESUMEN	4
CONTENIDO	5
CAPÍTULO I. INTRODUCCIÓN	
1.1. ANTECEDENTES	8
1.2. OBJETIVO	9
1.3. JUSTIFICACIÓN	9
1.4. ESTRUCTURA DEL DOCUMENTO	10
CAPÍTULO II. FUNDAMENTOS TEÓRICOS DE LA FIBRA ÓPTICA	
2.1. INTRODUCCIÓN	12
2.2. FENÓMENOS ÓPTICOS	13
2.2.1. REFLEXIÓN	13
2.2.2. REFRACCIÓN	14
2.2.3. LEY DE SNELL	14
2.2.4. REFLEXIÓN TOTAL INTERNA	15
2.3. PROPAGACIÓN DE LA LUZ EN LA FIBRA	
2.3.1. DEFINICIÓN DE LA FIBRA ÓPTICA	17
2.3.2. ALGUNOS CONCEPTOS IMPORTANTES	18
2.4. PÉRDIDAS Y DISPERSIÓN	
2.4.1. PÉRDIDAS EN LA POTENCIA DE LA FIBRA ÓPTICA	23
2.4.2. PÉRDIDAS EN LA FIBRA	24
2.4.3. DISPERSIÓN	25
2.4.4. ELECCIÓN DE LA LONGITUD DE ONDA	28

CAPÍTULO III. INGENIERÍA DE SOFTWARE

3.1. INTRODUCCIÓN.....	29
3.2. VISIÓN EVOLUTIVA DEL SOFTWARE.....	30
3.3. PRINCIPALES PROBLEMAS DEL SOFTWARE.....	31
3.4. APLICACIONES DEL SOFTWARE.....	32
3.5. INGENIERÍA DE SOFTWARE.....	33
3.6. METODOLOGÍAS DE LA INGENIERÍA DE SOFTWARE.....	35
3.6.1. EL CICLO DE VIDA CLÁSICO.....	36
3.6.2. CONSTRUCCIÓN DE PROTOTIPOS.....	39
3.6.3. DESARROLLO RÁPIDO DE APLICACIONES.....	40
3.6.4. DESARROLLO INCREMENTAL.....	41
3.6.5. DESARROLLO EN ESPIRAL.....	42
3.6.6. DESARROLLO BASADO EN COMPONENTES.....	44
3.6.7. MODELO DE MÉTODOS FORMALES.....	47
3.6.8. TÉCNICAS DE CUARTA GENERACIÓN (TG4).....	49
3.6.9. RESUMEN DEL MODELADO.....	50
3.7 SELECCIÓN DEL MODELO DE DESARROLLO.....	51

CAPÍTULO IV. DISEÑO DEL SOFTWARE

4.1. DESCRIPCIÓN DEL PROBLEMA.....	52
4.2. DESCRIPCIÓN FORMAL DEL SOFTWARE.....	53
4.2.1. SECCIÓN DE SIMULACIÓN DE LEYES ELEMENTALES.....	54
4.2.2. SECCIÓN DE SIMULACIÓN DE LA FIBRA ÓPTICA.....	55
4.2.3. SECCIÓN DE CÁLCULOS TÉCNICOS.....	56
4.3. PROPUESTA DE LA SOLUCIÓN.....	57
4.4. ALCANCE DE LA IMPLEMENTACIÓN.....	58
4.5. DIAGRAMA DE FLUJO DE DATOS.....	58
4.6. DIAGRAMA GENERAL DEL SISTEMA.....	62
4.7. MÉTODO DE DESARROLLO.....	64
4.8. CONCLUSIÓN DEL DISEÑO.....	68

CAPÍTULO V. IMPLEMENTACIÓN DEL SOFTWARE

5.1. IMPLEMENTACIÓN TEÓRICA.....	69
5.2. HERRAMIENTAS DE DESARROLLO	70
5.3. IMPLEMENTACIÓN DE OpenGL EN WINDOWS	71
5.4. IMPLEMENTACIÓN DE OpenGL EN DELPHI.....	71
5.5. INTERFACES GRÁFICAS DEL SOFTWARE "SOPHY I"	
5.5.1. SECCIÓN DE SIMULACIÓN DE LEYES ELEMENTALES	72
5.5.2. SECCIÓN DE SIMULACIÓN DE LA FIBRA ÓPTICA	75
5.5.3. SECCIÓN DE CÁLCULOS TÉCNICOS	76
5.6. CONCLUSIÓN DE LA IMPLEMENTACIÓN	78

CAPÍTULO VI. PRUEBAS Y CONCLUSIONES

6.1. PRUEBAS.....	79
6.2. PERSPECTIVAS.....	81
6.3. CONCLUSIONES.....	82

BIBLIOGRAFÍA	83
---------------------------	-----------

APÉNDICE A : "PRINCIPIOS DE LAS ONDAS ELECTROMAGNÉTICAS"	88
---	-----------

APÉNDICE B : "CÓDIGO FUENTE"	97
---	-----------

APÉNDICE C : "MANUAL DE USUARIO"	104
---	------------

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes

En la actualidad existen diversas herramientas de software que ayudan a resolver determinados problemas orientados a ciertas áreas científicas y/o tecnológicas. Entre algunas de estas herramientas se pueden mencionar aquellas orientadas al diseño, aprendizaje, modelado, desarrollo de software, control de dispositivos físicos, entre otros. Dentro de estos grupos de herramientas de software se encuentra el software de apoyo para el estudio o enseñanza de algún fenómeno, los cuales son de gran ayuda para los usuarios interesados en observar las causas y efectos de ciertos experimentos, que pudieran resultar costosos o que puedan requerir de mucho tiempo de los usuarios al realizarlos de forma real. En lo referente al campo de la electrónica, existe de igual forma una gran variedad de herramientas que ayudan a los científicos, investigadores y personas relacionadas con este campo a realizar tareas específicas. Sin embargo, la existencia de software de apoyo dedicados a experimentar el comportamiento de la fibra óptica a través del estudio de los fenómenos ópticos en forma de gráficos (3D), es casi nula.

1.2. Objetivo

El propósito de este proyecto es desarrollar un software de apoyo para el estudio de los fenómenos ópticos en una fibra, el cual ayude a experimentar y comprender el confinamiento de la luz dentro de la fibra óptica, mostrando resultados numéricos y gráficos, éstos últimos con gráficos animados en escenarios tridimensionales.

1.3. Justificación

Esta herramienta de simulación fue desarrollada para alcanzar el objetivo antes mencionado. Haciendo principal énfasis en que puede ser empleada como apoyo en la docencia, herramienta autodidacta, o material de consulta para personal técnico que maneje este tipo de tecnología.

Tomando en cuenta las dimensiones del núcleo de las fibras ópticas que existen en el mercado, los cuales varían entre $52\mu\text{m}$ de diámetro para fibra multimodo y $5\text{-}10\mu\text{m}$ para fibra monomodo parecido al diámetro de un cabello, es difícil llevar a cabo dicho análisis de una forma directa con la fibra óptica, a menos que se cuente con tecnología de instrumentación especializada para estudiar dicho medio de comunicación [44]. En la actualidad este tipo de tecnología de instrumentación si existe pero el precio para su adquisición es elevado, comparado con lo que en sí esta valorado este software [38].

1.4. Estructura del Documento

En este documento, se trata de presentar en forma resumida pero muy significativa el trabajo realizado para la creación del software de apoyo para el estudio de los efectos ópticos en una fibra "Sophy I". Debido a esto, el documento esta estructurado de tal manera que pueda ser entendido por cualquier lector, y que éste tenga un panorama general del objetivo del proyecto, las bases teóricas para un rápido entendimiento del tema, la forma en que se fue desarrollando desde la especificación del problema, el diseño, implementación y pruebas. Todo esto presentado de una forma organizada y detallada para su fácil comprensión.

Así entonces, el documento esta organizado de la siguiente manera:

- **Capítulo I.** Este capítulo trata acerca de los antecedentes que se tomaron en consideración para la creación del software "Sophy I", así como también, el objetivo que persigue el proyecto, y la justificación que se presenta al desarrollarlo.
- **Capítulo II.** En este segundo capítulo, se establecen las bases teóricas de la Tecnología Fibra Óptica con el fin de que el lector pueda obtener de éste, los principios básicos para entender el funcionamiento de esta tecnología.
- **Capítulo III.** Para que el software antes mencionado fuese desarrollado con calidad y profesionalismo, se llevó a cabo con la intervención de la Ingeniería de Software utilizando la metodología en espiral. Por consiguiente este capítulo abarca una significativa explicación de la ingeniería de software, principales metodologías, y una justificación de la metodología utilizada en este proyecto.
- **Capítulo IV.** En este capítulo se contempla el diseño del software el cual consiste principalmente en una serie de etapas que se fueron llevando a cabo para su desarrollo. Dentro de estas etapas se incluye una descripción formal del software, diagramas de flujo de datos del sistema, algunos centros de transformación y el diagrama general del sistema. Además de incluir la forma en como se fue aplicando la metodología en espiral.

- **Capítulo V.** Después de haber tratado el diseño del software, en este capítulo se habla acerca de la forma en como se implementó, informando acerca de los lenguajes utilizados para su codificación, al igual que algunas consideraciones que se tomaron en cuenta para utilizar dichos lenguajes, y por último, una descripción de las interfaces que se implementaron.
- **Capítulo VI.** En este último capítulo, se presenta una explicación breve y concisa de las pruebas que se le hicieron al software “Sophy I”, algunas perspectivas hechas al proyecto y al final unas conclusiones del mismo.
- **Bibliografía.** En esta parte, se presenta en forma de lista las diferentes fuentes bibliográficas utilizadas para la construcción de este trabajo.
- **Apéndices.** Este documento incluye tres apéndices donde en cada uno de ellos se presenta de forma resumida algunas partes importantes utilizadas en el desarrollo de este proyecto, y se encuentra en el siguiente orden:
 - **Apéndice A.** Principios de las ondas electromagnéticas y tipos de fibras ópticas.
 - **Apéndice B.** Las funciones más elementales implementadas en la herramienta de programación utilizada.
 - **Apéndice C.** En este último apéndice se incluye uno de los documentos más imprescindibles de toda elaboración de un sistema bajo una ingeniería de software, y éste se trata del manual de usuario, que para el software “Sophy I”, es presentado en forma explícita.

CAPÍTULO II

FUNDAMENTOS TEÓRICOS DE LA FIBRA ÓPTICA

2.1. INTRODUCCIÓN

La energía radiante obedece algunas leyes que pueden explicarse a partir de una corriente de paquetes de energía llamados fotones¹, o, a partir de un tren de ondas transversales.

El concepto de fotón es útil para explicar las interacciones de la luz con la materia, con lo cual se produce un cambio en la forma de energía, ya que tanto la luz como los objetos materiales contienen partículas las cuales por lo general se encuentran en un estado activo el cual puede ser interrumpido por la interacción con otras partículas [39].

El concepto de onda suele emplearse para explicar la propagación de la luz y algunos de los fenómenos de formación de imágenes.

La propagación de la luz puede ser analizada en detalle usando la teoría de ondas electromagnéticas, en algunos casos especiales se usa un método simplificado llamado trazo de rayos óptica geométrica. Puesto que es más sencillo y a veces suficiente, representar la propagación de la luz mediante rayos en vez de ondas. Entonces, el rayo no es mas que la

¹ Fotón : partícula de energía luminosa.

línea de avance o dirección de propagación de la energía radiante y por tanto perpendicular al frente de onda, es decir, perpendicular a la dirección del campo eléctrico y magnético (ambos campos son perpendiculares entre sí) que genera la onda. La trayectoria de los rayos es determinada aplicando las leyes de reflexión y refracción. Además este método examina la dirección de propagación, ignorando pero no eliminando los campos eléctricos y magnéticos. La luz entra en la categoría general de ondas electromagnéticas, como por ejemplo las ondas de radio. Sin embargo, la luz tiene frecuencias más grandes que las ondas de radio. La luz visible, por ejemplo cubre el rango de 0.43×10^{15} y 0.75×10^{15} Hz. La comunicación fibra óptica usa luz en el rango de frecuencia de 0.2×10^{13} a 0.37×10^{13} Hz [44].

2.2. FENÓMENOS ÓPTICOS

Como ya se ha mencionado, es más fácil explicar el comportamiento de la luz en algunos fenómenos ópticos usando el trazado de rayos, en lugar de la descripción detallada que ofrece la teoría electromagnética, o la teoría del fotón [44]. Así los efectos de reflexión, refracción y propagación de la luz en una fibra puede ser descritos en términos de rayos.

Sin embargo, la óptica geométrica es aplicable solo cuando las ondas en consideración contengan una longitud de onda mucho más pequeña que la estructura que guía la onda.

Algunos conceptos fundamentales para entender el confinamiento de la luz en la fibra son: reflexión, refracción y reflexión total interna, los cuales se presentan a continuación.

2.2.1. REFLEXIÓN

Cuando un rayo de luz incide sobre una superficie reflectora [16,44], el rayo rebota hacia atrás como una pelota lanzada a una pared. Típicamente, una superficie reflectora es aquella que es altamente esmaltada, opaca y revestida con materiales especiales.

La ley de reflexión establece que el *ángulo de incidencia es igual al ángulo de reflexión*. En la figura 2.1, el rayo de incidencia es la línea AO, el rayo reflejado es OB, y \perp es perpendicular a la superficie reflectora. Los ángulos θ_1 y θ_2 , son el ángulo de incidencia y el ángulo de reflexión respectivamente.

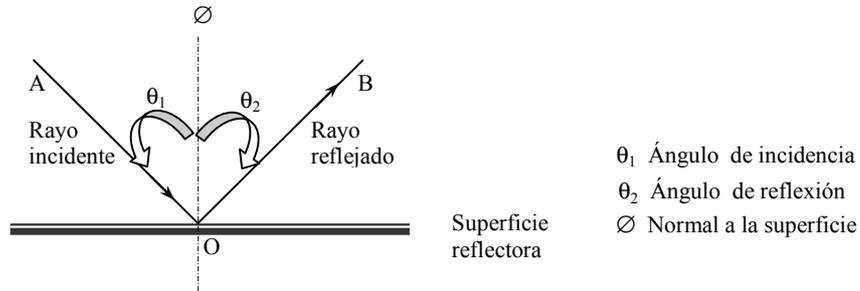


Figura 2.1. Rayo de incidencia y de reflexión.

2.2.2. REFRACCIÓN

Cuando un rayo viaja en la frontera entre dos materiales con diferentes índices de refracción n_1 y n_2 , Como se muestra en la figura 2.2, se presentan dos fenómenos: el de reflexión y el de refracción. El primer fenómeno ya ha sido analizado, ahora para éste ultimo el rayo o una parte de él pasa al segundo medio, haciendo que el fenómeno de refracción ocurra.

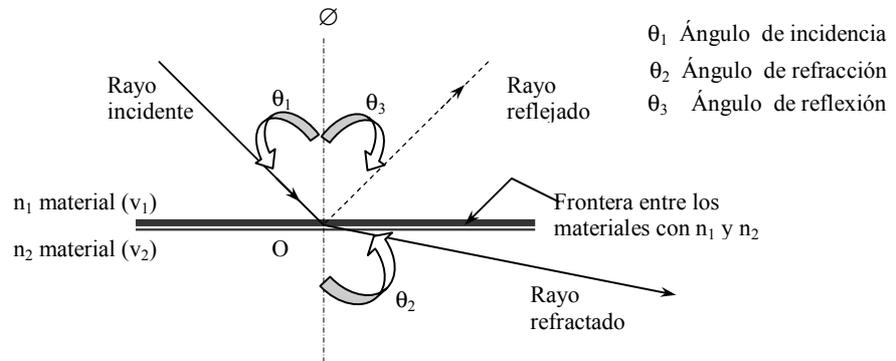


Figura 2.2. Rayos de incidencia y refracción cuando $n_1 < n_2$, por tanto $\theta_2 < \theta_1$.

2.2.3. LEY DE SNELL

Esta importante ley, llamada así en honor del matemático holandés Willebrord Van Roijen Snell, afirma que **el producto del índice de refracción del primer medio y el seno del ángulo de incidencia de un rayo, es igual al producto del índice de refracción del segundo medio y el seno del ángulo de refracción [24].** El rayo incidente, el rayo refractado y la normal a la superficie de separación de los medios en el punto de

incidencia, están en un mismo plano. En general, el índice de refracción de una sustancia transparente más densa es mayor que el de un material menos denso, es decir, la velocidad de la luz es menor en la sustancia de mayor densidad. Por tanto, si un rayo incide de forma oblicua sobre un medio con un índice de refracción mayor, se desviará hacia la normal, mientras que si incide sobre un medio con un índice de refracción menor, se desviará alejándose de ella. Los rayos que inciden en la dirección de la normal son reflejados y refractados en esa misma dirección. Entonces la ley de Snell puede ser descrita como sigue:

$$n_1 \text{ sen } \theta_1 = n_2 \text{ sen } \theta_2 \quad (2.1)$$

Vista de otra forma

$$\text{Sen } \theta_1 / \text{Sen } \theta_2 = n_2/n_1 \quad (2.2)$$

Si el rayo viaja de un alto a un bajo índice de refracción, entonces θ_2 se aleja de la perpendicular a la superficie. Por tanto $\theta_1 < \theta_2$ tenemos que

$$\text{Sen } \theta_1 / \text{Sen } \theta_2 = v_1/v_2 \quad (2.3)$$

Donde v_1 y v_2 están dados de la siguiente manera

$$v_1 = c/n_1 \quad \text{y} \quad v_2 = c/n_2$$

Donde $c = 300 \times 10^6$ m/s (velocidad de la luz en el espacio), pero cuando la luz se propaga a través de un material, la velocidad a la cual se propaga en el vacío se decrementa. Para el caso en estudio v_1 y v_2 son velocidades de propagación de la luz tanto del primer y segundo medio con índices de refracción n_1 y n_2 respectivamente.

2.2.4. REFLEXIÓN TOTAL INTERNA

Puesto que los rayos se alejan de la normal cuando entran en un medio menos denso, y su desviación con respecto a la normal aumenta a medida que aumenta el ángulo de incidencia, existe un determinado ángulo de incidencia denominado ángulo crítico, para el cual, el rayo refractado forma un ángulo de 90° con la normal a la superficie, por lo que avanza justo a lo largo de la superficie de separación entre ambos medios. Si el ángulo de incidencia se hace mayor que el ángulo crítico, los rayos de luz serán totalmente reflejados [43]. **La reflexión total no puede producirse cuando la luz pasa de un medio menos denso a otro más denso.**

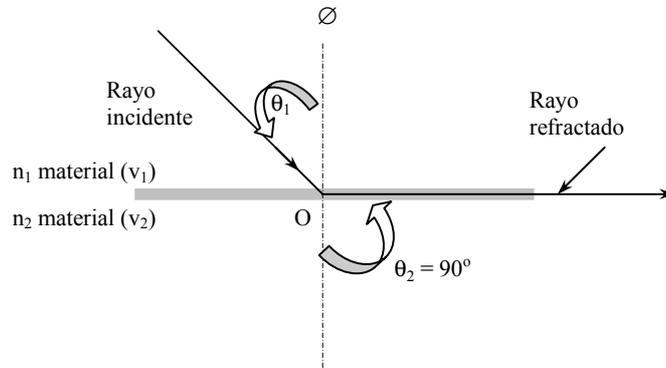


Figura 2.3. Refracción con ángulo crítico.

Referente a la situación descrita en la figura 2.3, es claro que cuando $\theta_2 = 90^\circ$, el rayo no viaja a través del material (traspasándolo) cuyo índice de refracción es n_2 , si no que únicamente viaja sobre la superficie de éste. Aplicando la ley de refracción de Snell para este caso, se obtiene que

$\text{Sen } \theta_1 / \text{Sen } \theta_2 = n_2/n_1$ y si además $\theta_2 = 90^\circ$, se consigue

$$\text{Sen } \theta_1 = n_2/n_1 \quad (2.4)$$

El ángulo de incidencia θ_1 para el cual $\theta_2 = 90^\circ$ es llamado ángulo crítico θ_c .

$$\text{Sen } \theta_c = n_2/n_1 \quad (2.5)$$

Teniendo como resultado

$$\theta_c = \text{Sen}^{-1} n_2/n_1 \quad (2.6)$$

Entonces si n_2 es menor que n_1 , la relación n_2/n_1 es menor que 1 y θ_c es menor que 90° . Ahora, si un rayo incide en los límites entre dos materiales con diferentes índices de refracción, y el ángulo de incidencia es igual al ángulo crítico, el rayo refractado viajará sobre la frontera de ambas superficies y nunca entrará al material con índice de refracción n_2 .

Ahora, si $\theta_1 \geq \theta_c$ toda la luz incidente es reflejada, a esta condición se le llama *reflexión total interna*. Lo cual ocurre solamente cuando la luz viaja de un alto a un bajo índice de refracción entre los materiales.

2.3. PROPAGACIÓN DE LA LUZ EN LA FIBRA

2.3.1. DEFINICIÓN DE FIBRA ÓPTICA

La fibra óptica es un filamento de vidrio (compuesta de cristales naturales) o plástico (cristales artificiales), por el cual viaja un rayo de luz. La información a transmitir va contenida en el rayo de luz que viaja a través de la fibra.

La fibra óptica presenta un gran ancho de banda, lo que supone que la información transmitida en ella es mayor que la transmitida en los conductores convencionales, pues la fibra permite manejar valores desde cientos de MHz hasta decenas de GHz.

La fibra óptica presenta dimensiones más reducidas que los medios preexistentes, lo que se traduce en una reducción del espacio del conductor. Un cable de 10 fibras tiene un diámetro aproximado de 8 o 10mm. Y proporciona la misma o más información que un coaxial de 10 tubos. Asimismo, el peso del cable de fibras ópticas es muy inferior al de los cables metálicos, redundando en facilidad de instalación [37].

La materia prima para fabricarla es abundante en la naturaleza, lo cual lleva los costos a la baja según mejoran los procesos tecnológicos, al contrario de lo que ocurre con el cobre, cuyo precio depende fundamentalmente de las reservas. De hecho, el precio de los cables de fibra ha ido disminuyendo progresivamente desde su nacimiento, aunque no al grado de ser mucho más económico que los conductores de cobre que existen actualmente.

Los dos constituyentes esenciales de las fibras ópticas son el núcleo y el revestimiento. El núcleo es la parte más interna de la fibra y es la que guía la luz. Consiste en una o varias hebras delgadas de vidrio o de plástico con diámetro de 50 a 125 micras [39]. El revestimiento es la parte que rodea y protege al núcleo. El conjunto de núcleo y revestimiento está a su vez rodeado por un forro o funda de plástico u otros materiales que lo resguardan contra la humedad, el aplastamiento, los roedores, y otros riesgos del entorno.

2.3.2. ALGUNOS CONCEPTOS IMPORTANTES

Una fibra muy típica es la de índice escalonado, la cual es mostrada en la figura 2.4. En esta fibra el índice de refracción cambia en forma súbita, desde el centro de la fibra a su cubierta. La propagación de la luz dentro de la fibra con índice escalonado se puede explicar usando la técnica de trazado de rayos. Esto significa que se puede seguir la luz en forma de rayo a través de la fibra.

Los rayos que viajan a través de la fibra son de dos tipos, un tipo está constituido por los llamados rayos meridionales, los cuales alguna vez en su trayectoria cruzan el centro de la fibra. Al otro tipo pertenecen los rayos *skew* que viajan a través ella sin tocar nunca el centro de la fibra [43]. En este contexto solo se hablará de los rayos del primer tipo.

Las condiciones necesarias para el confinamiento eficiente de los rayos depende de la estructura de la fibra óptica, así como de las características de la fuente de luz.

Reflexión total interna

Una fibra típica con índice escalonado se muestra en la figura 2.4. Dos rayos son visualizados, uno de ellos es proyectado con un ángulo θ_{L2} mucho más grande que el otro θ_{L1} .

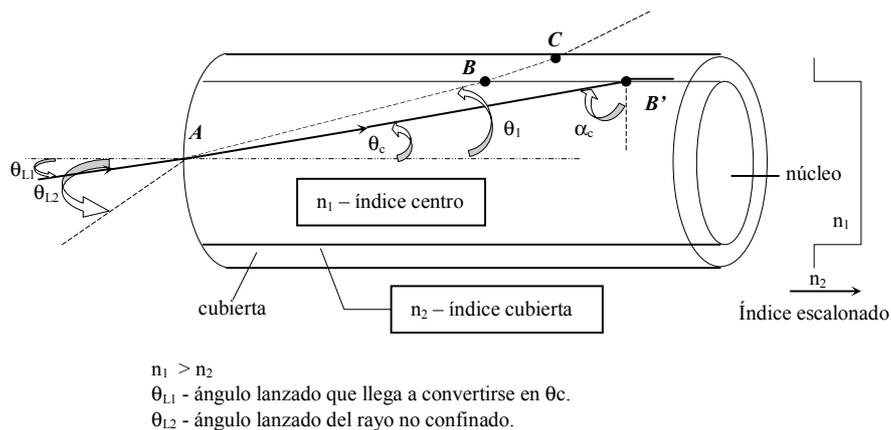


Figura 2.4. Propagación de la luz en una fibra con índice escalonado.

En el punto A, entre el aire y el núcleo, la refracción toma lugar y el primer rayo que incide con un ángulo θ_{L2} , continua con un ángulo pequeño, lo cual significa que $\theta_{L2} > \theta_1$. Continuando su trayectoria después de la refracción el rayo consigue incidir en el punto B entre los límites del centro y la cubierta. Entonces nuevamente la refracción toma lugar y el rayo incidente continua sobre la cubierta. Finalmente, el rayo vuelve a incidir entre los límites de la cubierta y el vacío, ocasionando así, que éste salga a partir del punto C y abandone la fibra por completo. De este modo se observa al tipo de rayos no confinados que no se propagan a través de la fibra.

Para el segundo rayo (la línea sólida), nuevamente la refracción toma lugar en el punto A. En el punto B', el límite entre el núcleo y la cubierta, ocurre la reflexión total interna. Este rayo es confinado en el núcleo de la fibra. Por conveniencia, se asume que el ángulo de incidencia entre el núcleo y la cubierta es llamado ángulo crítico α_c .

$$\alpha_c = \sin^{-1}(n_2/n_1) \quad (2.7)$$

Así que un rayo incidente con un ángulo mayor a α_c se propagará a través de la fibra.

El rayo crítico (línea sólida) forma un ángulo θ_c con el centro de la fibra. Los rayos con ángulos de propagación mayores que θ_c no se propagarán. Debe observarse que $\theta_{L2} > \theta_1$, y que el rayo con θ_1 existe en la fibra y no es confinado en ella.

El ángulo θ_c es llamado ángulo crítico de propagación

$$\text{Sen } \alpha_c = \text{Cos } \theta_c = n_2/n_1 \quad (2.8)$$

Es importante resaltar que la reflexión total interna ocurre solo cuando la luz viaja de un medio con un alto índice de refracción, a otro con índice más bajo.

Modos de propagación

Tomando en cuenta el concepto de la reflexión total interna se puede decir, que todos los rayos con ángulos menores a θ_c se propagan en la fibra. Estos rayos son llamados modos de propagación. El número total de modos de propagación en la fibra se incrementa conforme θ_c se incrementa. Lo anterior se ejemplifica en la figura 2.5, se ilustran 3 rayos distintos, propagándose con ángulos θ_1 , θ_2 y θ_3 respectivamente.

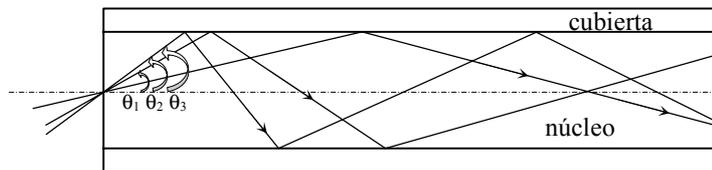


Figura 2.5. Tres modos de propagación θ_1 , θ_2 y θ_3 .

El ángulo θ_c depende de n_2/n_1 , como se observa en la ecuación (2.8).

Es común distinguir entre modos de orden alto cuyos ángulos de propagación son cercanos al ángulo crítico θ_c , de los modos de orden bajo con ángulos de propagación mucho más pequeños que el ángulo crítico θ_c .

Modo de conversión (modo de acoplamiento)

El acoplamiento de modos no es más que la conversión de un modo a otro, el cual es causado por las curvas o dobleces de la fibra óptica. Estas curvas pueden ser grandes o pequeñas, nombrándoseles macro curvaturas y micro curvaturas respectivamente.

La conversión consiste en transformar los modos de un orden alto a un orden bajo ó viceversa, y dependiendo del tipo de curvatura que presente la fibra óptica, este cambio será benéfico o no para la transmisión de la energía luminosa. En las macrocurvaturas los modos de orden bajo, son convertidos en orden alto para continuar con su trayectoria dentro de la fibra óptica sin ninguna pérdida, pero los modos de orden alto, son transformados a modos de orden muy alto ocasionando tal pérdida y la salida de la fibra óptica, ver figura 2.6.

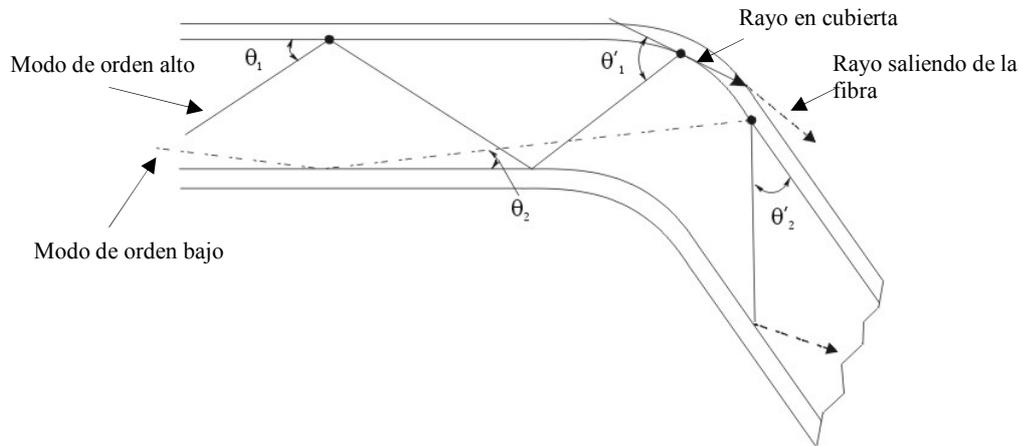


Figura 2.6. Efectos de una macrocurvatura.

Para el caso de las microcurvaturas los modos de orden bajo presentan la misma transformación que en las macrocurvaturas, mientras que los modos de orden alto no se pierden pues son convertidos en modos de orden bajo, todo esto es mostrado en la figura 2.7.

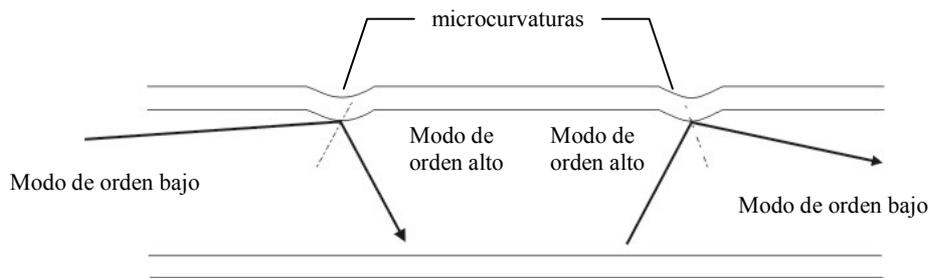


Figura 2.7. Efectos de una microcurvatura.

Ángulo de aceptación y apertura numérica

En la figura 2.8 se muestran dos rayos entrando a la fibra, los cuales llegan a ser rayos críticos en la fibra. Como se puede visualizar en el punto A, ocurre el fenómeno de la refracción, entonces θ_a no es igual a θ_c . Así que para asegurar el confinamiento de la luz dentro de la fibra óptica, los rayos deben entrar con un ángulo menor o igual que el ángulo θ_a . El ángulo $2\theta_a$ es llamado ángulo de aceptación. En tres dimensiones, este sería un cono de aceptación limitado por el ángulo $2\theta_a$.

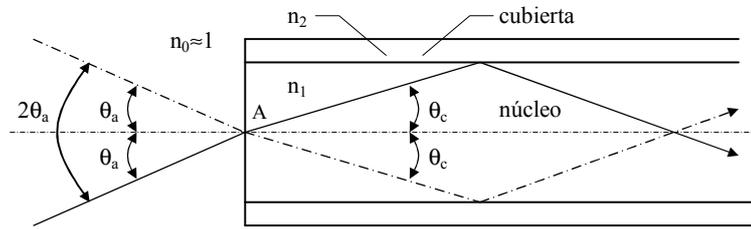


Figura 2.8. Ángulo de aceptación.

Por la ley de Snell $n_0 \text{ Sen } \theta_a = n_1 \text{ Sen } \theta_c$

$$\text{Sen } \theta_a / \text{Sen } \theta_c = n_1/n_{\text{aire}} = n_1 \quad (2.9)$$

y

$$\text{Sen } \theta_a = n_1 \text{ Sen } \theta_c \quad (2.10)$$

El término $\text{Sen } \theta_a$ es llamado Apertura Numérica (N.A) y

$$\text{N.A.} = \text{Sen } \theta_a = n_1 \text{ Sen } \theta_c \quad (2.11)$$

Para obtener N.A., en términos de los índices de refracción n_1 y n_2 , donde n_1 es el índice del núcleo y n_2 es el índice de la cubierta, usando las ecuaciones 2.9 y 2.11 y la identidad trigonométrica

$$\text{Cos } \theta = 1 - \text{Sen}^2 \theta \quad (2.12)$$

Se obtiene

$$\text{N.A.} = (n_1^2 - n_2^2)^{1/2} \quad (2.13)$$

Entonces, la mitad del ángulo de aceptación esta dado por

$$\theta_a = \text{Sen}^{-1}(\text{N.A.}) = \text{Sen}^{-1} (n_1^2 - n_2^2)^{1/2} \quad (2.14)$$

Si se define Δ como la diferencia relativa de índices de refracción

$$\Delta = (n_1^2 - n_2^2)^{1/2} / 2n_1 \quad (2.15)$$

$$\Delta = (\text{N.A.})^2 / 2n_1^2 \quad (2.16)$$

De las ecuaciones 2.13 y 2.15 se obtiene

$$\begin{aligned} (\text{N.A.})^2 &= n_1^2 - n_2^2 = 2Xn_1^2X\Delta \\ \text{N.A.} &= n_1(2\Delta)^{1/2} \end{aligned} \quad (2.17)$$

2.4. PÉRDIDAS Y DISPERSIÓN

El rendimiento que pueda ofrecer una fibra óptica depende de varios factores, entre los que se pueden mencionar la composición específica del material del que esta construida, el diámetro del núcleo, la forma en que varía el índice de refracción, entre otras. Ahora, en tal rendimiento también se tiene que considerar las imperfecciones que la fibra presenta, así como pequeñas curvaturas o dobleces (microcurvaturas), ya que tales imperfecciones y microcurvaturas afectan o provocan pérdidas en la fibra óptica. A continuación se estudiará la forma de identificar las causas básicas de la pérdida de energía en la fibra, la dispersión de ésta y los efectos que ocasionan dichas pérdidas en la fibra.

2.4.1. PÉRDIDAS EN LA POTENCIA DE LA FIBRA ÓPTICA

Como se puede observar en la figura 2.9, P_{in} es la potencia de entrada dentro de la fibra y P_{out} es la potencia de salida de ésta. Las pérdidas en la fibra se puede definir como

$$\text{Loss} = P_{in} / P_{out} \quad (2.18)$$

Usando logaritmos para representar las pérdidas en términos de decibeles, la ecuación anterior queda definida como

$$\text{Loss}_{dB} = 10 \times \log(P_{out}/P_{in}) \quad (2.19)$$

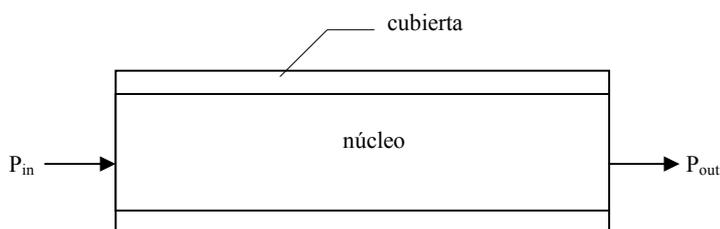


Figura 2.9. Potencia de entrada P_{in} y potencia de salida P_{out} de una fibra.

Debido a que las pérdidas se incrementan conforme aumenta la longitud de la fibra, normalmente se hace referencia a las pérdidas en decibeles por kilómetro.

2.4.2. PÉRDIDAS EN LA FIBRA

Existen fundamentalmente tres causas por las cuales la luz se pierde dentro de la fibra óptica [44].

1. **Pérdidas Materiales.** Esta pérdida es presentada debido a la absorción por materiales, esta absorción es causada por la interacción de la luz con la estructura molecular del material, así como pérdidas por impurezas del propio material.
2. **Esparcimiento de Luz.** Esta pérdida consiste en la dispersión de la luz debido a las moléculas del material, ya sea por imperfecciones en la estructura o impurezas que existan en él. La luz esparcida no se propaga dentro de la fibra si no que únicamente se pierde.
3. **Pérdidas por microcurvaturas o dobleces.** Estas pérdidas son causadas por imperfecciones o deformaciones de la estructura de la fibra.

Las pérdidas antes citadas dependen de la longitud de onda, entonces la cuidadosa selección de una longitud de onda a la cual va a operar la fibra óptica puede minimizar dichas pérdidas.

Pérdidas materiales

Las pérdidas debido a la estructura atómica del material son relativamente pequeñas, mientras que las pérdidas ocasionadas por las impurezas del material pueden ser reducidas mejorando el proceso de manufacturación.

Esparcimiento

Cuando la luz es esparcida por una obstrucción, el resultado es una pérdida en la potencia. El término obstrucción se refiere a las variaciones de densidad en el material ocasionando así un cambio en el índice de refracción. Por lo general las obstrucciones son inherentes al proceso de manufacturación y no pueden ser eliminadas, el comportamiento que presentan tales obstrucciones son como fuentes puntuales esparciendo la luz en todas direcciones.

Pérdidas por microcurvaturas o dobleces

Las variaciones estructurales en la fibra o deformaciones en ésta causan radiación de luz. La figura 2.10 ilustra la radiación causada por un cambio en el diámetro, como se puede observar el ángulo α_1 en la deformación es más pequeño que el ángulo crítico así que el rayo que incide con tal ángulo deja la fibra, mientras que si estuviese ausente la deformación, el rayo se confinaría dentro de la fibra tal y como lo ilustra la línea punteada.

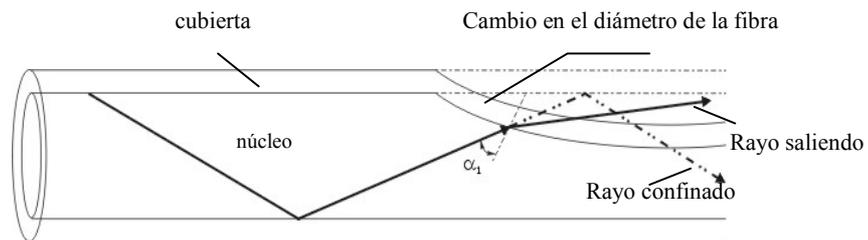


Figura 2.10. Radiación pérdida por cambios en el diámetro.

2.4.3. DISPERSIÓN

El término dispersión es usado para describir el efecto “ensanchamiento de pulso” en la fibra óptica. La figura 2.11 muestra que el pulso que se obtiene en la salida de la fibra es mucho más ancho que el pulso de entrada. Como la señal que viaja a través de la fibra es un pulso de luz, éste llega a ser más ancho por la variación del fenómeno de propagación. Teniendo presente lo anterior podemos definir la dispersión como la producción del ensanchamiento de un pulso luminoso de salida, causado por una idealización de un pulso de entrada con una anchura cerca de cero. En otras palabras suponiendo un pulso de entrada con una anchura igual a cero, el ancho del pulso de salida es totalmente el resultado de la dispersión en la fibra.

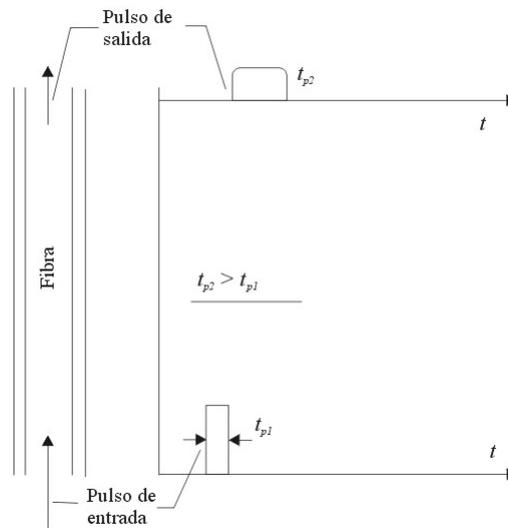


Figura 2.11. Ensanchamiento de pulso.

Considerando el ancho de pulso de entrada t_{p1} y el ancho del pulso de salida como t_{p2} se define la dispersión Δt como

$$\Delta t = (t_{p2}^2 - t_{p1}^2)^{1/2} \quad (2.20)$$

La dispersión total de una fibra depende de su longitud, por eso cuando más grande es dicha longitud es mas la dispersión que se genera. Por lo general la dispersión de una fibra en particular es dada por el fabricante, y se mide en nanosegundos por kilómetro (ns/km) o picosegundos por kilómetro (ps/km), así entonces la dispersión de una fibra se obtiene de

$$\Delta t = L \times (\text{dispersión/km}) \quad (2.21)$$

Donde Δt es la dispersión de la fibra, L es la longitud de la fibra y dispersión/km es proporcionada por el fabricante.

Dispersión intermodal

Este tipo de dispersión "Intermodal" es identificada por el hecho de que las ondas se propagan en modos. Por ello ésta no es mas que una dispersión entre los modos causada por la diferencia en la propagación del tiempo entre un modo y otro. Para entender este tipo de dispersión considérese dos modos propagándose a lo largo del eje de la fibra como se ilustra en la figura 2.12, el modo de propagación con ángulo θ_c y otro modo con un ángulo de propagación igual a cero (modo cero).

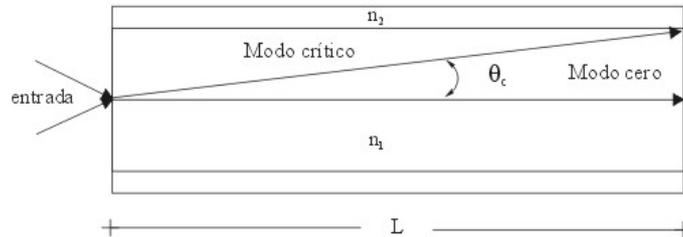


Figura 2.12. Dispersión intermodal (tiempo de retardo).

Para el modo cero el tiempo de viaje será mínimo y puede ser expresado por

$$t_{d0} = L/(c/n_1) \quad (\text{retardo mínimo de propagación}) \quad (2.22)$$

donde L es la longitud de la fibra, n_1 es el índice de refracción del núcleo y c/n_1 es la velocidad de la luz en la fibra. Para el rayo que viaja con un ángulo θ_c el retardo máximo se expresará de la siguiente forma

$$t_{dc} = (L/\cos \theta_c)/(c/n_1) \quad (\text{retardo máximo de propagación}) \quad (2.23)$$

Entonces la diferencia entre el retardo mínimo t_{d0} y el máximo t_{dc} es el tiempo durante el cual el pulso de energía llegará a la salida de la fibra. Por tanto se obtiene la dispersión de la siguiente manera

$$\Delta t = t_{dc} - t_{d0} \quad (2.24)$$

En términos de la apertura numérica la dispersión intermodal esta dada por

$$\Delta t = (Lx(N.A.)^2)/(2 x n_1 x c) \quad (2.25)$$

o

$$\Delta t/L = (N.A.)^2/(2 x n_1 x c) \quad (2.26)$$

Donde Δt es la cantidad del ensanchamiento del pulso y $\Delta t/L$ es el ensanchamiento por unidad de la longitud de la fibra.

Como se puede observar la dispersión intermodal esta directamente relacionada con la apertura numérica N.A., donde una gran apertura numérica significa una gran dispersión. De la misma manera un gran ángulo de aceptación, aunque hace más fácil la inyección de la potencia en la fibra, también ocasiona un incremento en la dispersión teniendo como resultado una reducción en la razón de transmisión de datos.

Dispersión intramodal

La Dispersión Intramodal a veces llamada Dispersión Cromática, es un resultado directo del hecho que la luz en la fibra consiste de un grupo de frecuencias. Esta dispersión esta relacionada con el ancho de línea $\Delta\lambda$ y frecuentemente esta dada en términos de picosegundos por kilómetro por nanómetro (ps/(kmxnm)) del ancho de línea. Aumentando el ancho de línea se incrementa la dispersión.

2.4.4. ELECCIÓN DE LA LONGITUD DE ONDA

En los inicios del desarrollo de la tecnología de la fibra óptica la longitud de onda utilizada fue de 820 nm [44], esto por la disponibilidad de fuentes y detectores que manejan este rango de longitud de onda. Para un mejor rendimiento la selección de la longitud de onda debe ser tomada de tal forma que pueda minimizar las pérdidas y dispersión en la fibra.

Existen tres ventanas de bajas pérdidas para ayudar a elegir dentro de un rango de longitud de onda que permite disminuir las pérdidas en la fibra. En la tabla 2.1 se muestran tres rangos de frecuencia y las pérdidas que se presentan en su uso [44].

λ (nm)	Pérdida aproximada por kilómetro (dB)
820 – 880	2.2
1200 – 1320	0.6
1550 – 1610	0.2

Tabla 2.1. Ventanas de bajas pérdidas.

CAPÍTULO III

INGENIERÍA DE SOFTWARE

3.1. INTRODUCCIÓN

En sus inicios el mundo de la informática se orientó básicamente al hardware y a su mejoramiento [19]. En la actualidad el hardware ha conseguido niveles tales que, al modernizar algo en cuestiones de informática, lo que más se tiene en consideración es el software.

La Ingeniería del Hardware se ha dedicado a investigar nuevas tecnologías y aplicarlas para desarrollar hardware de bajo costo y alta calidad, una vez conseguido esto, lo que se intenta ahora es realizar lo mismo con el Software [15, 31]. Así mismo, en los inicios de la informática, la programación se veía como un oficio artesanal, existían pocos métodos formales y pocas personas los usaban. El programador desarrollaba su trabajo aplicando la técnica de “*ensayo y error*”.

Actualmente, el software es el elemento de mayor costo en la ingeniería de sistemas [31]. Por ello la Ingeniería de Software busca conseguir un software de calidad a un precio razonable, y a la vez entregar el producto en los plazos establecidos.

3.2. VISIÓN EVOLUTIVA DEL SOFTWARE.

Durante las tres primeras décadas de la Informática, el principal desafío era el desarrollo del hardware de las computadoras, de forma que se redujera el costo de procesamiento y almacenamiento de datos.

En los primeros años de la era de la computación el software se realizaba de una forma muy empírica sin ninguna planificación, además era diseñado a medida para cada aplicación y tenía una distribución relativamente pequeña. Así mismo el software se desarrollaba y era manipulado por una única persona, la cual si existía un error lo corregía por tanto no existía documentación ni la seguridad de que la persona estaría allí siempre para cualquier contratiempo [31].

La segunda era de los sistemas de computadora se extiende desde la mitad de la década de los sesenta hasta finales de los setenta. En ésta década aparecieron las computadoras de la tercera generación y se desarrollaron técnicas de programación como la multiprogramación y el tiempo compartido. Y mientras las computadoras estaban haciéndose más complejas, resultó obvio que la demanda por los productos de software creció en mayor cantidad que la capacidad de producir y mantener dicho software. En esta misma década se lograron significativos avances en los dispositivos de almacenamiento, lo que condujo a la primera generación de sistemas de gestión de bases de datos. Por último en esta segunda era el software se estableció como un producto, el cual se desarrollaba para tener una amplia distribución en un mercado multidisciplinario.

La tercera era en la evolución de los sistemas computacionales, empezó a mediados de 1970 extendiéndose mas allá de una década. Esta época se caracterizó por el surgimiento de los sistemas distribuidos donde el procesamiento de la información es distribuido sobre varias máquinas más que confinado a una única computadora, así también las redes de área local y global, y la llegada de los microprocesadores, produciendo un extenso grupo de productos inteligentes [36].

La cuarta era, y considerada la última en la evolución de los sistemas de computadora se caracteriza por el desarrollo de sistemas personales mas potentes; los sistemas expertos y el software de inteligencia artificial han salido del laboratorio para entrar en aplicaciones prácticas de una gran variedad de problemas del mundo real, software de redes neuronales artificiales, entre otros [31].

3.3. PRINCIPALES PROBLEMAS DEL SOFTWARE

Existen problemas que han persistido a través de la evolución de los sistemas basados en computadora y que continúan aumentando, algunos de estos son [31,15]:

- Los avances del software continúan dejando nuestra habilidad de construir software para alcanzar el potencial del hardware.
- Nuestra habilidad de construir nuevos programas no puede ir al ritmo de la demanda de nuevos programas, ni podemos construir programas lo suficientemente rápido como para cumplir las necesidades del mercado y de los negocios.
- El uso extremo de computadoras ha hecho a la sociedad cada vez más dependiente de la operación fiable del software. Cuando el software falla, pueden ocurrir daños económicos enormes y ocasionar sufrimiento humano.
- Luchamos por construir software informático que tenga fiabilidad y alta calidad.
- Nuestra habilidad de soportar y mejorar los programas existentes se ve amenazada por diseños pobres y recursos inadecuados.
- Baja calidad del software con relativa frecuencia.
- La planificación y estimación de costes económicos y temporales son frecuentemente muy imprecisos.
- Elevado coste de corrección de errores en el desarrollo
- Proyectos fuera de plazo y presupuesto.

Con el desarrollo del software surgieron un gran número de problemas como los expuestos hasta este punto, de los cuales se pueden generalizar los siguientes; el desarrollo del software era demasiado lento ocasionando un gran retraso en su entrega, poca calidad en el software, presupuestos inexactos, bajo desempeño, y con dificultades en su mantenimiento, entre otros. Generalmente estos problemas fueron considerados en ese entonces como un trance conocido como “crisis del software” [15,19,31]. Como una solución a esta crisis surgió la disciplina *ingeniería de software* la cual fue propuesta por primera vez en una conferencia de la NATO en el año de 1968 por el profesor Fritz Bauer [15,26,36].

3.4. APLICACIONES DEL SOFTWARE

Es difícil establecer categorías genéricas para las aplicaciones del software, ya que conforme aumenta su complejidad es más difícil de encasillar en una u otra categoría. Las siguientes áreas del software indican la amplitud de las aplicaciones potenciales [19,31]:

Software de Sistemas. Conjunto de programas creado como herramienta para otros programas (sistemas operativos, compiladores, CASE, editores y utilidades de gestión de archivos). Se caracteriza por existir una fuerte interacción con el hardware del ordenador; una gran utilización por múltiples usuarios; una operación concurrente que requiere una planificación, una compartición de recursos y una sofisticada gestión de procesos; unas estructuras de datos complejas y múltiples interfaces externas.

Software de tiempo real. Conjunto de programas que mide, analiza y controla sucesos del mundo real a medida que ocurren. Se caracterizan por los siguientes componentes:

- Adquisición de Datos: recoge y formatea los datos de entrada.
- Análisis: transforma la información dependiendo de la aplicación.
- Control/Salida: responde a la entrada (entorno externo).
- Monitorización: coordina el resto de los componentes, de forma que pueda mantenerse la respuesta en tiempo real (entre 1 msg a 1 min) .

Software de gestión. Realiza tareas de procesamiento de datos y operaciones de cálculo interactivo (aplicaciones de gestión de empresas).

Software de ingeniería y científico. Utiliza algoritmos de manejo de números cuyas aplicaciones contemplan algunas áreas como astronomía, cálculo, biología molecular, fabricación automática, entre otras. Además de nuevas aplicaciones como el diseño asistido por computadora (CAD), la simulación de sistemas y otras aplicaciones interactivas.

Software empotrado. Este tipo de software reside en memorias de solo lectura (ROM) y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo. Este software puede ejecutar funciones muy limitadas y curiosas (controlar un horno de microondas) o suministrar una función significativa y con capacidad de control (funciones digitales en un automóvil, tales como control de gasolina, indicaciones en el salpicadero, sistemas de frenado, entre otros).

Software de computadoras personales. Es software de uso general como son los procesadores de textos, hojas de cálculo, gestores de bases de datos, juegos, aplicaciones financieras, etc.

Software de inteligencia artificial. Es aquel que hace uso de algoritmos no numéricos para resolver problemas complejos en los que no son adecuados el cálculo o el análisis directo. Forman parte de este grupo aplicaciones de sistemas expertos, reconocimiento de patrones (OCR), redes neuronales artificiales, prueba de teoremas, juegos, entre otros.

Software Educativo. Es aquel sistema de software que forma parte y ha sido concebido especialmente para apoyar los fines educacionales de un courseware. Siendo el courseware un sistema que se utiliza para promover un cierto aprendizaje. Y dentro de ese sistema, se incluyen recursos tecnológicos, uno de los cuales es un subsistema software. Así pues, el producto courseware puede incluir muchos otros elementos, además del subsistema software; por ejemplo, documentos en diversos formatos y soportes (audio, vídeo, papel), así como métodos, procedimientos o técnicas orientadas a promover y/o controlar determinadas variables del proceso de enseñanza. Este software implica que de alguna manera, parte del contenido objeto del aprendizaje que se pretende lograr estará integrado o sustentado, implícita o explícitamente en el software [5,14].

Hoy el problema está dado por el desafío de reducir el coste y mejorar la calidad de las soluciones basadas en computadoras, soluciones que se implementan con el software.

3.5. INGENIERÍA DE SOFTWARE

Como respuesta a la "crisis del software" han evolucionado un conjunto de técnicas denominadas colectivamente Ingeniería de Software. Estas técnicas se enfrentan con el software como un producto de ingeniería.

La crisis que durante cierto tiempo ha afectado a la industria informática tiene por raíz el desarrollo del software. Esta crisis fue el resultado directo del desarrollo del hardware. Lamentablemente, los costos del software no han disminuido con los del hardware. La práctica

de la ingeniería de software tiene por objeto la construcción de grandes y complejos sistemas de una forma rentable.

Hoy en día existe un consenso sobre la necesidad de un enfoque más disciplinado para el desarrollo de sistemas de computación, este enfoque está comprendido en la frase "Ingeniería de Software" [34].

La Ingeniería de Software se considera hoy como una disciplina por derecho propio. El ingeniero de software debe poder evaluar y aplicar las técnicas de computación existentes de manera rentable y fácil de usar [36].

El reconocimiento de las realidades del software es el primer paso hacia la formulación de soluciones prácticas para el desarrollo del mismo.

Se han propuesto varias definiciones de Ingeniería de Software, las que tienen como elemento común que la ingeniería de software trata de la construcción de sistemas utilizando principios de ingeniería para el desarrollo de estos [34].

De lo anterior se ha considerado como completa la definición de ingeniería de software de Sommerville, la cual nos dice, " La ingeniería de software es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza" [36].

De la definición antes mencionada se desprenden dos palabras claves: una de ellas es la "disciplina de la ingeniería" la cual consiste en que el ingeniero busca soluciones a problemas haciendo uso de teorías, métodos y herramientas, además de trabajar en algunas ocasiones bajo restricciones financieras y organizacionales. La segunda palabra clave "todos los aspectos de la producción de software", indica que la ingeniería de software no solo contempla los procesos técnicos de desarrollo, si no también actividades como la administración de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

La ingeniería del software abarca tres elementos clave como son los métodos, herramientas y procedimientos.

Los **Métodos** nos enseñan cómo construir técnicamente el software, abarcando [31]:

- Planificación y estimación de proyectos
- Análisis de requisitos del sistema y del software
- Diseño de estructuras de datos
- Diseño de programas y procedimientos algorítmicos
- Codificación
- Prueba
- Mantenimiento

Las **herramientas** suministran un soporte automático o semiautomático para los métodos. En la actualidad existen herramientas para soportar cada uno de los métodos mencionados anteriormente. Algunas de estas herramientas son comúnmente conocidas como herramientas CASE (Ingeniería de software asistida por computadora), estas herramientas combinan software, hardware y base de datos sobre ingeniería de software. Así, una herramienta CASE es un producto computacional enfocado a apoyar una o mas técnicas dentro de un método de desarrollo de software. Existe varias maneras de clasificar a las herramientas CASE vistas desde diferentes perspectivas, en donde cada una de dichas clasificaciones encierran un gran número de estas herramientas. Algunas clasificaciones y ejemplos de herramientas CASE son mencionadas en las siguientes bibliografías [15,19,31,36], así como en los siguientes sitios de Internet [11,13,22].

Los **procedimientos** son la aplicación de los métodos utilizando las herramientas. Son los elementos de unión entre los métodos y las herramientas. Definen la secuencia en la que se aplican los métodos, ayudan a asegurar la calidad y a coordinar los cambios.

3.6. METODOLOGÍAS DE LA INGENIERÍA DE SOFTWARE

Los campos en que se utilizan las computadoras hoy en día son innumerables. Se ha llegado al punto en que es posible decir que no hay un solo campo del conocimiento humano en el que no se haya aplicado la computación, pues se han creado aplicaciones para áreas del conocimiento humano tan distintas como lo pueden ser la astrofísica y la cocina. Por esta razón el software que se puede crear es tan variado como las ideas que se forman en las mentes de las personas. Aún así, todos los proyectos de creación de software siguen un

proceso muy parecido para llegar desde la idea original hasta un producto terminado y funcional. Este proceso básico está conformado por análisis, diseño, implementación, pruebas y mantenimiento.

Entonces, este ciclo de vida básico como a veces se le llama es tan solo la idea general tras cualquier proceso particular de desarrollo de software. Actualmente existen diversos modelos o métodos para cumplir con los pasos necesarios de este ciclo, cada uno con un enfoque distinto. Dentro de los que se pueden nombrar el modelo de construcción de prototipos, el modelo de desarrollo rápido de aplicaciones, el de procesos evolutivos - que a su vez se divide en los modelos incremental, en espiral, de ensamblaje de componentes y de desarrollo concurrente -, el modelo de métodos formales, y por último las técnicas de cuarta generación [41].

Todos estos modelos atacan el problema desde un ángulo distinto, y cada uno tiene su forma particular de aplicación, pero a fin de cuentas, todos ellos ayudan a realizar la función para la que fueron diseñados: *el desarrollo de software mediante técnicas de ingeniería*.

El paradigma óptimo y el peor no existe; pues un paradigma puede ser tan bueno y a la vez puede ser el peor para llevar a cabo el desarrollo de un software. Solo que éste depende del problema en cuestión y de la facilidad con que el ingeniero lo aplique.

A lo largo de la historia de la Ingeniería de Software se han propuesto varios paradigmas diferentes, cada uno exhibiendo ventajas e inconvenientes, pero todos tienen una serie de fases genéricas en común. Algunos de éstos son mencionados a continuación.

3.6.1. EL CICLO DE VIDA CLÁSICO

El primer paradigma que empezaron a utilizar los programadores y personas relacionadas con el desarrollo de software fue el paradigma de ciclo de vida clásico, también conocido como “Modelo en Cascada” propuesto por Royce en el año de 1970 [31,36,43]. Consta de pasos que sugieren un enfoque sistemático y secuencial del desarrollo de software, que comienzan en un nivel de sistemas y progresa a través del análisis, diseño, codificación, prueba y mantenimiento. Este modelo se basa en etapas o fases en las que se orienta cualquier proceso de ingeniería convencional, como las que se llevan a cabo en cualquier tipo

de construcción, siendo estas transformadas en actividades fundamentales de desarrollo de software como son [1,19,31,36]:

Ingeniería y análisis del sistema

En esta etapa se determina si el proyecto es o no factible de realizarse y se determinan tiempos y costos aproximados, estableciendo así la ruta crítica de cada actividad. Esto es, porque la falta de planeación de un sistema es la causa principal de retrasos en programación, incremento de costos, poca calidad, y altos costos de mantenimiento en los desarrollos de productos de software.

Con frecuencia se dice que es imposible realizar una planeación inicial, porque la información precisa sobre las metas del proyecto, necesidades del cliente y restricciones del producto no se conocen al comenzar el proyecto de desarrollo. Sin embargo, uno de los principales propósitos de esta fase es aclarar los objetivos, problemas o necesidades y restricciones. La dificultad de la planeación no debe desalentar tan importante actividad.

Análisis

Es el proceso en el cual se obtienen los requisitos del software, además de comprender la naturaleza de los programas a construirse, el dominio de la información, las funciones requeridas, rendimiento e interconexión. En otras palabras, en esta etapa se trata de comprender mejor el problema con la finalidad de que los requerimientos del usuario se especifiquen de la forma más correcta y precisa que sea posible y que el grupo de trabajo traslade esos requerimientos a un diseño del software con un mínimo de errores .

Diseño

El diseño del software es realmente un proceso multipaso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interface, estos atributos están orientados y son aplicables al diseño estructurado [31,36]. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

Codificación

Llegado este punto se empieza a codificar algoritmos y estructuras de datos, definidos en las etapas anteriores, en el correspondiente lenguaje de programación y/o para un

determinado sistema gestor de bases de datos. Durante la fase de codificación de un sistema se lleva a cabo la traducción del diseño al programa [1,31].

Prueba

El objetivo de las pruebas es garantizar que el sistema haya sido desarrollado correctamente, sin errores de diseño y/o programación. Es conveniente que sean planteadas al menos tanto a nivel de cada módulo (aislado del resto), como de integración del sistema (según sea la naturaleza del proyecto en cuestión se podrán tener en cuenta pruebas adicionales, por ejemplo de rendimiento).

Mantenimiento

Finalmente la aplicación resultante se encuentra ya en fase de producción (en funcionamiento para el cliente, cumpliendo ya los objetivos para los que ha sido creada). A partir de este momento se entra en la etapa de mantenimiento, que supondrá pequeñas operaciones tanto de corrección como de mejora de la aplicación, así como otras de mayor importancia, fruto de la propia evolución.

La figura 3.1 muestra el paradigma del ciclo de vida clásico para la ingeniería de software. También conocido como el “método en cascada”.

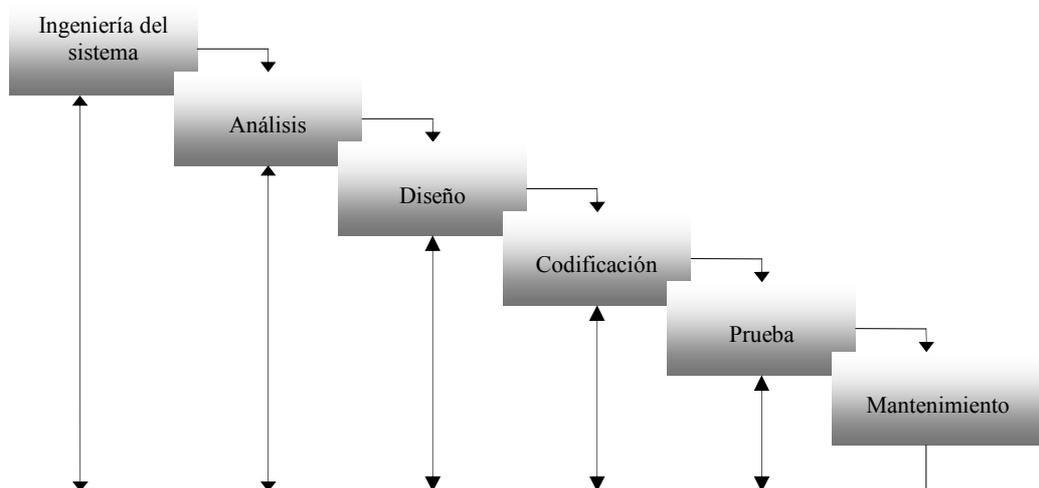


Figura 3.1 El ciclo de vida clásico.

3.6.2. CONSTRUCCIÓN DE PROTOTIPOS

En este modelo se desarrolla una versión del software de una manera rápida y de poca calidad por la velocidad de su creación, y originalmente fue introducido por primera vez en el año de 1987 por Curtis [43].

Normalmente el cliente define unos objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida. La construcción de prototipos es un proceso que facilita al programador la creación de un modelo del software a construir, en un breve plazo de tiempo [19,31]. El modelo tomará una de las tres formas siguientes:

1. Un prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de cómo se producirá tal interacción.
2. Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado, o
3. Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

La figura 3.2 ilustra la secuencia de sucesos del paradigma de construcción de prototipos. Como en todos los métodos de desarrollo de software, la construcción de prototipos comienza con la recolección de los requisitos. El técnico y el cliente se reúnen y definen los requisitos conocidos y perfilan las áreas en donde será necesario una mayor definición. Luego se produce un “diseño rápido”. El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario, además de que conduce a la construcción de un prototipo. El prototipo es evaluado por el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar [19]. Se produce un proceso interactivo en el que el prototipo es “afinado” para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que se tiene que hacer.



Figura 3.2. Construcción de prototipos.

3.6.3. DESARROLLO RÁPIDO DE APLICACIONES

En el modelo *Desarrollo Rápido de Aplicaciones (DRA) (Rapid Application Development, RAD)* el desarrollo del software es lineal secuencial, que es utilizado para un ciclo de desarrollo extremadamente corto [31]. Este modelo es una adaptación de alta velocidad del modelo lineal secuencial, el cual utiliza un enfoque de construcción basada en componentes y utilizando en algunos casos técnicas de cuarta generación. Si se comprende bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo (de 60 a 90 días) [25]. Cuando este modelo es utilizado para el desarrollo de sistemas de información, se enfoca o comprende las siguientes fases [21]:

Modelado de gestión. Esta fase consiste en modelar el flujo de información y la gestión de ésta entre los procesos, las personas y lugares en donde se genera ya sea como entrada o como salida.

Modelado de datos. En esta fase el flujo de la información se refina como un conjunto de objetos de datos. Se definen las características de cada uno de los objetos y las relaciones entre estos.

Modelado de proceso. Una vez identificados los objetos en la fase anterior, estos son utilizados para representar e implementar funciones de gestión para lograr el flujo de información planteado en un inicio.

Generación de aplicaciones. El DRA asume la utilización de técnicas de cuarta generación. Este proceso DRA trabaja para volver a utilizar componentes de programas ya existentes o crear componentes reutilizables. En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.

Pruebas y entregas. Como el proceso DRA enfatiza la reutilización, y debido a que muchos de los componentes de los programas ya se han comprobado (examinado), esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

3.6.4. DESARROLLO INCREMENTAL

En este paradigma se realiza un análisis, un diseño, se construye una primera versión con lo esencial del sistema, se prueba y se entrega, luego se vuelven a repetir los mismos pasos (análisis y diseño) para las funciones creadas solo que ahora más completas, se construye y se entrega una nueva versión, y nuevamente son repetidos todos estos pasos hasta tener un sistema lo más completo posible. Cada incremento debe entregar un producto operacional . Cada incremento es revisado por el usuario para aportar mas sobre las nuevas versiones [19,31,36].

En un proceso de desarrollo incremental, los clientes identifican de forma somera, los servicios que proveerá el sistema. Se Identifican que servicios son más importantes y cuales menos importantes [19]. Entonces, se definen varios incrementos en donde cada uno proporciona un subconjunto de funcionalidad del sistema, o mejor dicho se produce un incremento del software [36]. La asignación de servicios a los incrementos depende de la prioridad del servicio. Los servicios de prioridad más alta son los que se entregan primero al cliente. Este modelo de desarrollo fue introducido por primera vez en el año de 1980 por Mills [36,43].

Cuando se utiliza un modelo incremental, el primer incremento a menudo es un producto esencial (núcleo) [31]. El cliente puede utilizar el producto central, y dependiendo de su respuesta se planea un próximo incremento del software, o se modifica este producto central a

fin de cumplir mejor las necesidades del cliente. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo.

Este modelo a diferencia de otros modelos evolutivos como el de construcción de prototipos, se centra en la entrega de un producto operacional en cada incremento.

3.6.5. DESARROLLO EN ESPIRAL

El modelo en espiral fue propuesto originalmente por Boehm en 1988 [31,36,43], el cual presenta las características tanto del ciclo de vida clásico, como el de creación de prototipos, añadiendo al mismo tiempo un nuevo elemento; el análisis de riesgo, donde éste no se encuentra presente en los anteriores modelos. El modelo define cuatro actividades principales, representadas por cuatro cuadrantes, como es mostrado en la figura 3.3 [19]:

1. Planificación: Determinación de objetivos, alternativas y restricciones.
2. Análisis de riesgo: Análisis de alternativas e identificación/resolución de riesgos.
3. Ingeniería: Desarrollo del producto de “siguiente nivel”.
4. Evaluación del cliente: Valoración de los resultados de la ingeniería.

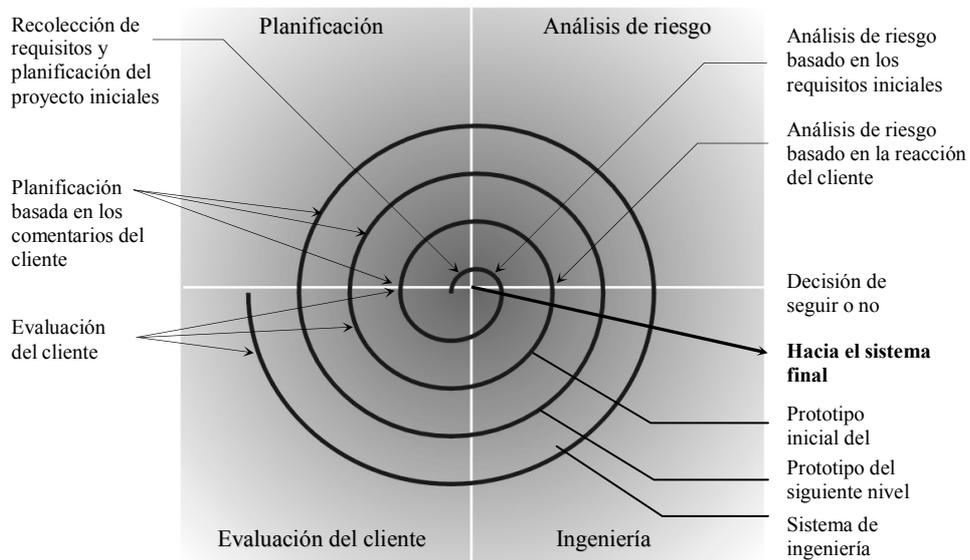


Figura 3.3. El modelo en espiral.

Por cada iteración alrededor de la espiral (comenzando en el centro y siguiendo el exterior), se construyen sucesivas versiones del software, cada vez más completas y mejoradas. Durante la primera iteración alrededor de la espiral se lleva a cabo la actividad de planificación, donde se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgos indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado del desarrollo como al cliente. Se pueden usar simulaciones y otros modelos para definir el problema más a fondo el problema y refinar los requisitos [19,31].

El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación del cliente) y sugiere modificaciones. Con base en los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de “seguir o no seguir”. Si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.

Sin embargo, en la mayoría de los casos, se sigue avanzando alrededor del camino de la espiral, y ese camino lleva a los desarrolladores hacia fuera, hacia un modelo más completo del sistema y al final al propio sistema operacional. Cada vuelta alrededor de la espiral requiere ingeniería, que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o con el de creación de prototipos. Debe tenerse en cuenta que el número de actividades de desarrollo que ocurre en el cuadrante inferior derecho aumenta al alejarse del centro de la espiral [31].

El paradigma del modelo en espiral para la ingeniería del software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala [31]. Utiliza un enfoque “evolutivo” para la ingeniería del software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como mecanismo de reducción del riesgo en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándolo dentro de un marco de trabajo interactivo que refleja de forma más práctica el mundo real. El modelo en espiral demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Pero, al igual que otros paradigmas, el modelo en espiral no es la panacea. Puede ser difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable. Requiere una considerable habilidad para la valoración del riesgo.

3.6.6. DESARROLLO BASADO EN COMPONENTES

El modelo de desarrollo basado en componentes surgió por la idea de una reutilización de componentes de software, similar a la reutilización de partes en otras disciplinas de ingeniería como la electrónica. Así pues, para explicar el modelo de desarrollo basado en componentes es necesario dar una visión muy general de lo que es la reutilización de software, la cual se describe enseguida.

Historia de la reutilización.

En los inicios del desarrollo de software diversas empresas empezaron con el desarrollo de sistemas pequeños y con metodologías artesanales, pero con el tiempo creció el número de clientes y la complejidad de los sistemas, lo cual obligó a la adopción de procesos de desarrollo poco formales por parte de las diversas empresas oferentes de software. Con ello, cada empresa adoptó los procesos de desarrollo basándose en su experiencia y en sus necesidades funcionales [2].

Por lo que, la forma de los ciclos de desarrollo de sistemas obedecía más bien a la naturaleza de los clientes, al nicho de mercado y al tamaño de los sistemas, pero no a un proceso de desarrollo estandarizado, que garantizara la confiabilidad o el éxito en los productos emitidos.

Hoy en día las empresas desarrolladoras de software han adoptado novedosos mecanismos de desarrollo para poder así, tener la oportunidad de incursionar en diferentes mercados. De allí la necesidad de ver que la reutilización de software es una herramienta importante, ya que logra desarrollar software de alta calidad de manera confiable; puesto que permite a los desarrolladores utilizar el conocimiento acumulado anteriormente para brindar soluciones con un menor costo y en un tiempo prudencialmente menor [4].

Así entonces, la programación por módulos demostró que se podían desarrollar procedimientos estándar, los cuales podían transportarse de un programa a otro conservando su funcionalidad. El surgimiento de lenguajes de programación que utilizaban librerías,

demostraba la necesidad creciente en la adopción de mecanismos de reutilización de software.

Actualmente las empresas desarrolladoras se dan a la tarea de crear grandes "arsenales" de software reutilizable, para disminuir así los costos de desarrollo y mantenimiento del software.

Definición de reutilización

El concepto de reutilización de software es muy variable, debido a que los distintos expertos en el campo lo basan de acuerdo a su experiencia y concepción del mundo que le rodea, además es un término nuevo lo cual oscurece su carácter real. Por ejemplo [23]:

La reutilización de software es una estrategia a largo plazo basada en metas específicas a corto plazo. En otras palabras, la reutilización debe jugar un papel importante en los esfuerzos de la organización para mejorar la calidad, reducir los costos y el tiempo de mercado..." [17]

"La reutilización es una característica importante para un componente de software de alta calidad..." [4]

"La reutilización de software permite reducir los costos de diseño, codificación y comprobación, porque se amortiza el esfuerzo a lo largo de varios años. Al reducirse la cantidad de código se simplifica también la comprensión, lo cual hace que aumente la probabilidad de que el código sea correcto..."[32]

Basándonos en las definiciones anteriores, se puede afirmar que:

La reutilización de software es una disciplina organizacional evolutiva, donde el objetivo fundamental es lograr una disminución sustancial de los costos de desarrollo y mantenimiento del software, y aumentar la producción y calidad del mismo.

Desarrollo basado en componentes.

Este desarrollo también conocido como ingeniería de software basada en componentes, surgió a finales de los 90 como un enfoque basado en la reutilización para el desarrollo de sistemas de software [36].

El objetivo de la tecnología de componentes software es construir aplicaciones complejas mediante ensamblado de módulos que han sido previamente diseñados por otros, a fin de ser reusados en múltiples aplicaciones [35]. La ingeniería de programación que sigue esta estrategia de diseño se la conoce por el acrónimo CBSE (Component-Based Software Engineering) y es actualmente una de las tendencias más prometedoras para incrementar la calidad del software, abreviar los tiempos de acceso al mercado y gestionar el continuo incremento de su complejidad.

Los componentes pueden ser considerados como proveedores independientes de servicios, esto significa que cuando un sistema necesite de algún servicio simplemente le llama sin importar que tal componente esté en uso, o en que lenguaje de programación fue desarrollado. Así entonces, un componente consta de dos características importantes para tener la función de componente reutilizable, una de estas es que el componente es una entidad ejecutable e independiente, ya que el código fuente no está disponible y ya no tiene que ser compilado como otros componentes del sistema en construcción, la otra característica es que cada uno de los componentes publican su propia interfaz para poder interactuar con el sistema.

El desarrollo de software basado en componentes es siempre guiado por un marco de trabajo. Donde dicho marco no es más que una colección de componentes junto con una arquitectura de software, el cual define las interfaces que los componentes pueden tomar y las reglas que gobiernan la composición entre ellos [12,27,33].

Con el desarrollo de software basado en componentes han surgido una variedad de lenguajes, herramientas, técnicas, entre otros, para el apoyo en la construcción de sistemas. Dentro de las que se pueden mencionar: Diseño de patrones, arquitecturas de software, marcos de trabajo de componentes, estilos arquitectónicos, *glue* y *scripts*, entre otros. La explicación detalladas de las técnicas antes mencionadas y de otras interesantes, están fuera del alcance de esta tesis, sin embargo para una mayor comprensión de ellas refiérase a [12,27,33,42].

El desarrollo orientado a componentes se puede integrar en un proceso de desarrollo de sistemas incorporando una actividad de reutilización específica. En este proceso una vez analizado los requerimientos del sistema se tiene la oportunidad de especificar los componentes a utilizar, donde algunos de estos pueden ser simplemente tomados de un conjunto de componentes disponibles (ya antes creados).

En el desarrollo orientado a la reutilización, los requerimientos del sistema se modifican de acuerdo con los componentes reutilizables disponibles ya que el diseño también se basa en componentes existentes [33,36]. Debido a esto puede ser que el diseño sea menos eficiente que uno de propósito específico. Sin embargo, los bajos costos en el desarrollo, una entrega más rápida del sistema y el incremento en la fiabilidad del sistema compensa esto.

3.6.7. MODELO DE MÉTODOS FORMALES.

La especificación del software es el resultado del proceso de análisis. El análisis se centra en los ámbitos de información, funcional y de comportamiento del problema en cuestión. Para comprender mejor lo que se requiere se divide el problema en partes y se desarrollan representaciones o modelos que muestren la esencia de los requisitos (modelo esencial). Si éste no refleja los requisitos del cliente, entonces, inevitablemente, el diseñador construirá un sistema incorrecto. Por otra parte, si la especificación es incompleta, ambigua o inconsistente, aunque haya sido aceptada por el cliente, el diseñador no podrá satisfacer los requisitos adecuadamente.

Por tanto, la forma de especificar un sistema tiene una gran influencia en la calidad de la solución implementada finalmente. Tradicionalmente los ingenieros de software han venido trabajando con especificaciones incompletas, inconsistentes o erróneas, lo que invariablemente lleva a la confusión y a la frustración en todas las etapas del ciclo de vida. Como consecuencia de esto, la calidad, la corrección y la completitud del software disminuyen.

Los métodos de análisis que hemos visto anteriormente en este capítulo conducen a especificaciones de este tipo, normalmente redactadas en lenguaje natural o, en el mejor de los casos, en pseudocódigo. Para modelar y describir los requisitos del software se utiliza una combinación de diagramas, textos, tablas y notaciones sencillas.

La otra alternativa consiste en describir el sistema utilizando un lenguaje de especificación formal, basado en la lógica o las matemáticas, con una sintaxis y una semántica formal [6]. De esta forma se elimina la ambigüedad y sería posible demostrar la completitud de la especificación e incluso la corrección de una implementación de esta especificación. Las continuas revisiones que recomiendan los métodos clásicos se sustituyen aquí por análisis y demostraciones matemáticas, con la ventaja de que, si conseguimos demostrar que un

programa es correcto, nunca presentará un error, cosa que no es posible conseguir utilizando métodos informales.

La ingeniería se basa fundamentalmente en el uso de modelos matemáticos y en el cálculo para extraer conclusiones sobre la viabilidad o la utilidad de los diseños que se realizan.

Los métodos formales proporcionan al desarrollo de software las mismas ventajas que presenta otras ramas de la ingeniería como por ejemplo el análisis matemático basado en modelos [36]. Los métodos formales pueden ser utilizados para especificar el modelo de comportamiento de un sistema, para verificar formalmente que la especificación al igual que la implementación satisfacen propiedades funcionales y de seguridad del sistema. En principio, el uso de técnicas formales puede producir implementaciones libres de errores. Sin embargo, esto requiere una verificación completa a lo largo de todo el proceso de desarrollo, desde la especificación de los requisitos hasta la implementación, lo que, en la práctica, se realiza muy raramente.

Según esto los métodos formales son la matemática aplicada de la ingeniería del software, y juega un papel en el desarrollo de software [31,36]. Proporcionan un método para calcular, y por tanto para predecir, cuál va a ser el comportamiento de un sistema software con anterioridad a su implementación.

El uso de métodos formales está especialmente indicado en aquellos sistemas en los que la corrección, completitud o fiabilidad del software juegan un papel fundamental (protocolos de comunicación, software de control de una central nuclear, gestión del tráfico aéreo, etc.). Los lenguajes de especificación formal conducen a representaciones de los requisitos que pueden ser analizadas o verificadas formalmente [3,6].

Si bien son más precisas y por tanto más claras, las especificaciones formales tienen como inconvenientes que son más difíciles de usar y de entender. Además, se pierde la posibilidad de revisarlas con el cliente. Las demostraciones de corrección y completitud son tan complejas que, hasta ahora, sólo han podido aplicarse a casos muy sencillos. Por estos motivos, el uso de especificaciones formales se ha venido reservando a aplicaciones muy concretas del software, en las que la seguridad y fiabilidad sean un componente fundamental.

Por último, la aplicación de estos métodos sólo a las partes críticas de un sistema es una estrategia práctica y útil. Además de que estos métodos son más viables para las personas

encargadas de desarrollar software que deben construir aplicaciones de mucha seguridad (software de aviación, dispositivos médicos, entre otros) [31].

3.6.8. TÉCNICAS DE CUARTA GENERACIÓN (TG4)

Estas técnicas incluyen un amplio grupo de herramientas de software donde cada una de estas tienen una función específica que puede ser aplicada en alguna de las etapas del desarrollo de software. El paradigma técnicas de cuarta generación pretende especificar el software usando formas de lenguaje especializado o notaciones gráficas que describan el problema que hay que resolver [19]. Para poder realizar esto el paradigma TG4 cuenta con algunas de las siguientes herramientas: lenguajes no procedimentales de consulta a base de datos, generación de informes, manejo de datos, interacción y definición de pantallas, generación de códigos, etc [31].

Este modelo como otros presenta una serie de ventajas y desventajas que los modelos anteriormente vistos.

Ventajas [19,31]

- Mejora significativa en la productividad de la gente que construye el software, porque el tiempo de desarrollo es menor.
- Las técnicas de cuarta generación junto con las herramientas de ingeniería del software y los generadores de código (lenguajes de cuarta generación) ofrecen una solución fiable a muchos de los problemas del software, porque permite al desarrollador centrarse en la representación de los resultados deseados, que es lo que se traduce automáticamente en código fuente que produce dichos resultados.
- El tiempo requerido para producir software se reduce considerablemente para aplicaciones pequeñas y de tamaño medio, así como la cantidad de análisis y diseño para las aplicaciones pequeñas, ya que existen en el mercado herramientas de ingeniería de software que dan apoyo a estas actividades.
- Utiliza generadores de código, eliminando la etapa de codificación.

Desventajas [19,31]

- Las herramientas actuales de cuarta generación no son más fáciles de utilizar que los lenguajes de programación, ya que no son lo suficientemente sofisticadas para entender el lenguaje natural.
- El uso de técnicas de cuarta generación sin diseño causa las mismas dificultades que cuando se desarrolla software mediante enfoques convencionales.
- Para proyectos grandes, las técnicas de cuarta generación exigen el mismo o más tiempo de análisis, diseño y prueba, perdiendo así un tiempo sustancial, debido a que la información que estas herramientas necesitan debe ser detallada, y si el cliente no proporciona este tipo de información se tienen que aplicar más esfuerzos para la gestión del proyecto.
- El ámbito de aplicación actual de las técnicas de cuarta generación se limita al análisis y obtención de información a grandes bases de datos, ya que, debido a las características y funcionalidad de las herramientas, exigen restricciones de uso.

3.6.9. RESUMEN DEL MODELADO

Hasta este punto se han presentado siete diferentes modelos de desarrollo de software, todos ellos fueron expuestos de tal forma que el lector tuviese una idea general de su importancia y la manera en como se pueden aplicar en el proceso de desarrollo del software.

Los modelos presentados en este documento, así como los que no se alcanzaron a mencionar ayudan a resolver un problema a través del desarrollo de software, solo que cada uno de estos modelos ofrece una manera original y a la vez distinta de llevar a cabo tal solución utilizando técnicas de ingeniería de software.

Entonces por la explicación antes vista de cada uno de los modelos se puede decir, que ningún modelo puede considerarse como el mejor o el peor, pues cada uno puede ser aplicado para un problema específico, y no solo eso, si no que también va a depender de la habilidad que tenga la persona encargada del proyecto (por lo general el ingeniero de software), el tamaño del mismo y otros varios factores que nos permiten elegir entre varias alternativas de llevar a cabo tal actividad.

3.7. SELECCIÓN DEL MODELO DE DESARROLLO

A pesar de los diversos modelos existentes en la actualidad y de los que surgen, la selección del paradigma a utilizar para llevar a cabo el desarrollo del software “Sophy I” no presento ninguna controversia, y esto se debió a que el problema de desarrollar un software que ayudara a estudiar los efectos ópticos en una fibra, presentaba determinadas características que condujo a tal decisión:

- El proyecto no era muy extenso, lo que se consideró una ventaja porque en su desarrollo no era necesario un grupo numeroso de trabajo, pues era suficiente una persona.
- Debido a la forma de estructurar el software, la cual resultó muy sencilla y entendible tanto para el cliente como para el desarrollador, no hubo necesidad de elegir un modelo en el cual se enfocara el software en forma de objetos, clases, componentes o procesos formales.
- Un punto importante fue la disponibilidad de constante comunicación con el cliente, lo cual facilitaba diversas actividades desde la especificación de los requerimientos hasta la verificación de la implementación del software.

Además de los puntos antes mencionados para la elección del modelo de desarrollo a utilizar para la creación del software “Sophy I”, se tomó en consideración los modelos en los cuales ya se tenía un poco de experiencia en el desarrollo de sistemas, y al final se eligió como modelo o paradigma de desarrollo de software el “modelo de espiral”.

Este modelo como ya ha sido tratado anteriormente (ver sección 3.7.5) ayuda a desarrollar software a través de iteraciones, y en cada una de ellas se llevan a cabo etapas como son la planificación, análisis de riesgo, ingeniería y evaluación de una versión del software creado en dicha iteración. Por tanto debido a la flexibilidad que presentó el problema, la continua participación del cliente y la experiencia del desarrollador en este paradigma, se optó por llevar a cabo la creación del software “Sophy” utilizando el modelo en espiral.

CAPÍTULO IV

DISEÑO DEL SOFTWARE

4.1. DESCRIPCIÓN DEL PROBLEMA

El problema que se describe a continuación, no es precisamente indispensable en el campo de la electrónica pero no por ello deja de ser importante, por ello el menester identificado como “problema” es la falta de herramientas de software en electrónica, principalmente enfocados a la enseñanza de la tecnología fibra óptica. Para comprender el funcionamiento de la tecnología fibra óptica es necesario tener algunos conocimientos de una ciencia exacta como lo es la física, ya que como en la fibra óptica lo que se transporta es energía luminosa (luz) es necesario describir en términos físicos como viaja esta energía de un lugar a otro y como interactúa con otros medios. Por ende, para poder entender los efectos ópticos que ocurren en una fibra óptica cuando la energía que en ésta viaja sufre algunas transformaciones, es ineludible estudiar dichos efectos como son : ley de reflexión, ley de refracción y fenómeno de reflexión total interna, esto utilizando alguna teoría como las vistas en la sección 2.1. En donde como ya se había aclarado en la sección 2.2 es más fácil explicar dicho comportamiento de la luz a través de fenómenos ópticos utilizando el trazado de rayos u óptica geométrica.

Los fenómenos ópticos a tratar son como ya se han mencionado: ley de reflexión, ley de refracción y fenómeno de reflexión total interna. Estos fenómenos ópticos son indispensables en el estudio planteado ya que con ellos se pretende entender y comprender por qué la luz cuando incide sobre una superficie se refleja, o por qué en otros casos la atraviesa. Y cómo la fibra óptica funciona sobre el principio en el cual la información que en ella viaja (energía luminosa) se lleva a cabo a través de reflexiones de la luz sin que esta tenga que salirse o abandonar el medio para un transporte ideal de la misma. Entonces es necesario tomar tales conocimientos sobre fibra óptica y tratar de plasmarlos en una herramienta de software, la cual nos permita de una manera mas fácil, cómoda y entendible el analizar estos fenómenos. La forma en que dicha herramienta debe de representar los fenómenos de interés, es en forma gráfica con animación en escenarios tridimensionales con la finalidad de dar una representación mas apegada a la realidad.

Además de que la herramienta a desarrollar entregue un análisis numérico y una representación gráfica de cada uno de los fenómenos antes citados, ésta también debe contar con una sección, en la cual se permita realizar un pequeño análisis de la construcción de un sistema de comunicaciones vía fibra óptica, éste último será suficientemente representado en forma numérica.

Hasta este punto se ha hablado de forma breve pero muy descriptiva el problema a resolver, enseguida se dará una descripción formal y detallada de la posible solución que éste puede tener.

4.2. DESCRIPCIÓN FORMAL DEL SOFTWARE

El sistema que se plantea en este proyecto consta de tres secciones, las cuales han sido diseñadas de tal forma que puedan cubrir en forma integral el estudio de los efectos ópticos en una fibra y algunos cálculos técnicos básicos los cuales nos ayudan a entender la construcción de sistemas mínimos vía fibra óptica.

Las secciones en las cuales se divide esta aplicación son:

1. Sección de simulación de leyes elementales
2. Sección de simulación del confinamiento de la luz dentro de la fibra óptica
3. Sección de cálculos técnicos para la fibra óptica multimodo

4.2.1. SECCIÓN DE SIMULACIÓN DE LEYES ELEMENTALES

Esta sección fue diseñada de tal manera que permite ver gráficamente las principales leyes, que rigen la trayectoria de la luz al incidir en alguna superficie, las cuales ya fueron comentadas en la sección 2.2. Estas leyes son:

- Ley de reflexión
- Ley de refracción
- El fenómeno de reflexión total interna

Cada una de estas leyes puede ser analizada por separado, con la posibilidad de introducir algunos parámetros para que la simulación mostrada sea personalizada conforme la decisión del usuario. Para el análisis de cualquiera de estas leyes, se cuenta con una pantalla segmentada que se presenta en la figura 4.1., en donde una parte de ésta es utilizada para introducir los parámetros necesarios para ejecutar la simulación. Otra área sirve para observar gráficamente dicha simulación, ésto dentro de un escenario tridimensional con gráficos animados, y una última área, la cual sirve para controlar la simulación gráfica como es reproducir, pausar, detener e iniciar una nueva simulación y ver de diversos ángulos el experimento.

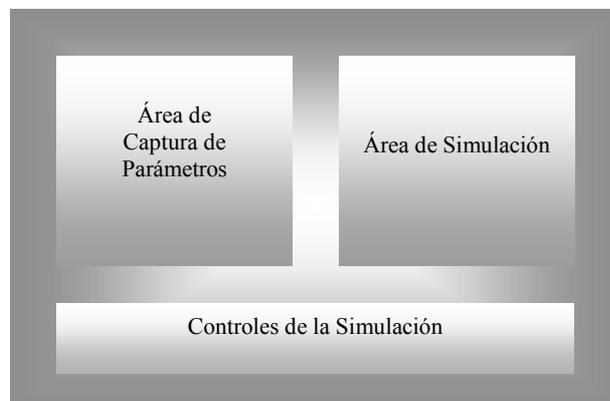


Figura 4.1. Divisiones de la pantalla.

Los parámetros que se requieren para que pueda ser llevada a cabo la simulación de cada una de las leyes incluidas en esta sección son los siguientes:

- En la ley de reflexión se requiere conocer únicamente el ángulo de incidencia del rayo que llega a la superficie.
- Para la ley de refracción se necesitan los valores numéricos de los índices de refracción de los medios por donde viaja el rayo, y el ángulo al cual debe incidir el rayo que se simula.
- Y por último, para el fenómeno de la reflexión total interna se necesita conocer nuevamente los índices de refracción, así como el ángulo al cual incide el rayo, este ángulo estará basado en el ángulo crítico calculado automáticamente por la simulación.

4.2.2. SECCIÓN DE SIMULACIÓN DE LA FIBRA ÓPTICA

En esta sección se simula gráficamente la fibra óptica y la luz que viaja dentro de ésta. Aquí la trayectoria de la luz es representada por líneas, y ésta se comporta de acuerdo con los valores que le sean asignados a los parámetros de la fibra óptica y al rayo de luz. Esta sección es una de las más interesantes, pues basándose en los principios mostrados en la primera sección, se puede experimentar y determinar aquellos rayos de luz que viajan a lo largo de la fibra óptica, y los cuales son los indicados para transportar información. Aunque también, permite visualizar los rayos que no son confinados dentro de la fibra óptica.

Los parámetros que en esta sección intervienen son:

Parámetros propios de la fibra óptica como son los índices de refracción del núcleo n_1 y la cubierta n_2 .

Con estos parámetros se calcula automáticamente la apertura numérica N. A., y un dato esencial para la simulación del ángulo de aceptación $2\theta_a$. Este último parámetro se representa gráficamente en la simulación como un cono, el cual indica que los rayos que lleguen al centro de la fibra óptica con un ángulo proporcionado por el usuario, menor o igual a los límites de este cono, se propagarán a lo largo de la fibra óptica. Mientras que los rayos que incidan con ángulos mayores que el ángulo de aceptación, son los que en algún momento saldrán de la fibra óptica. Los parámetros antes mencionados y otros que son calculados automáticamente por la simulación, son mostrados en la figura 4.2. y mencionados en el siguiente párrafo.

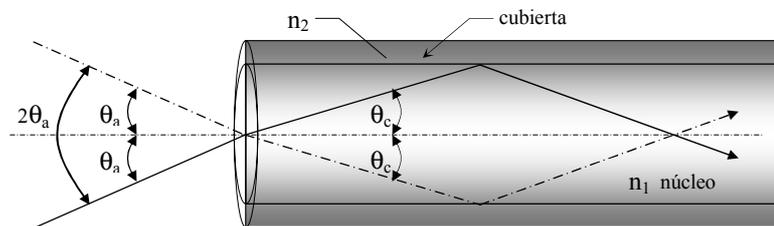


Figura 4.2. Parámetros que intervienen en la simulación de la fibra óptica.

Además del cálculo de la Apertura Numérica N.A, y el ángulo de aceptación θ_a , otros valores que son calculados por esta sección son, el ángulo crítico α_c y el ángulo crítico de propagación θ_c .

4.2.3. SECCIÓN DE CÁLCULOS TÉCNICOS.

En esta sección se obtiene información técnica sobre las fibras ópticas multimodo esto con la finalidad que ya ha sido mencionada en el inicio de este documento y la cual es únicamente la de ayudar a entender la construcción de sistemas mínimos vía fibra óptica.

Los parámetros que se requieren en esta sección para realizar algunos cálculos interesantes de tipo técnico y los cuales ya han sido previamente expuestos en el capítulo II son los siguientes:

- Potencia de entrada a la fibra óptica, la cual es proporcionada por la fuente de luz que se conecta para emitir las señales dentro de la fibra.
- Parámetros propios de la fibra óptica, entre los que podrían mencionarse como los índices de refracción del núcleo y la cubierta, la longitud de onda a la cual trabaja la fibra óptica en estudio, el radio del núcleo y la longitud de la fibra óptica en total.
- Existen otros parámetros de dos dispositivos que son también de vital importancia para la construcción de sistemas de comunicaciones por medio de este tipo de tecnología. Los cuales son importantes tomar en cuenta por la pérdida que presentan al utilizarlos. Estos dispositivos son los conectores y los empalmes, dispositivos que son necesarios para conectar o unir dos tramos de fibra óptica para extender la longitud del sistema que se está construyendo. Los parámetros que deben ser

tomados en cuenta, son la cantidad de conectores y empalmes que serán utilizados, y la pérdida individual promedio que puede presentar cada uno de estos elementos.

Los resultados que arroja esta sección son los siguientes:

- La pérdida total del sistema basado en los siguientes parámetros: la longitud de la fibra óptica, la longitud de onda a la que trabaja, las pérdidas obtenidas por el uso de conectores y empalmes.
- El número de modos que se generan
- La Dispersión Intermodal
- Y por último la potencia de salida, la cual nos indica cuanta potencia el sistema entregará a pesar de sufrir pérdidas por los puntos anteriormente vistos.

La forma en como se obtienen estos cálculos han sido explicados en el capítulo II de este documento.

4.3. PROPUESTA DE LA SOLUCIÓN

Una vez presentada la descripción formal del software en la cual se presentan las características y requisitos que debe contener la herramienta a desarrollar, con ayuda del modelo de desarrollo y herramientas de diseño así como los diagramas de flujos de datos y diagramas de estructura, se llevó a cabo el diseño e implementación de este software a través de diversas versiones hasta llegar a una versión final cumpliendo con los objetivos propuestos. La implementación del software como ya se mencionó en la descripción formal debe ser elaborada con una herramienta de desarrollo que permita la programación de los fenómenos ópticos usando gráficos tridimensionales, dicha herramienta es tratada en el capítulo V de este documento.

4.4. ALCANCE DE LA IMPLEMENTACIÓN

El alcance de este proyecto abarca única y exclusivamente el estudio de los fenómenos ópticos en una fibra, o lo que es lo mismo, el análisis de tales fenómenos a través de la modificación de sus ángulos incidentes, cálculos automáticos por parte del software de los ángulos resultantes (reflexión o refracción) y la simulación gráfica de estos fenómenos. Además de abarcar un pequeño conjunto de cálculos técnicos que ya han sido explicados en la sección 4.1.3. de este capítulo.

Con esta última sección no se intenta presentar un simulador completo para sistemas de comunicaciones vía fibra óptica, ya que para llevar a cabo tal simulador es necesario contemplar un gran número de parámetros del sistema que se proyecta construir. Así pues, la sección solo aspira dar una visión general de cómo se construyen los sistemas bajo esta tecnología (fibra óptica). Con lo antes mencionado se está llegando a justificar el objetivo en un inicio planteado el cual trata de desarrollar una herramienta que apoye el estudio de los efectos ópticos de una fibra óptica.

4.5. DIAGRAMAS DE FLUJO DE DATOS

Para el diseño del software simulador se utilizaron diversas herramientas para representar el sistema a implementar. Una de estas herramientas, fueron los diagramas de flujos de datos y diagramas de estructuras, estos últimos son explicados en el siguiente apartado.

Los diagramas de flujo de datos, ayudan a figurar el software como flujos de información, aparte de las transformaciones que dicha información tendría que pasar al trasladarse desde una entrada a una salida [8,31]. Estos diagramas brindan la oportunidad de representar el sistema o software a cualquier nivel de abstracción. Es decir, los diagramas de flujo de datos pueden ser divididos en niveles que simbolicen un mayor flujo de información y un mayor detalle funcional.

Para el diseño de este proyecto, se desarrollaron diagramas de flujo de datos a un segundo nivel de detalle. Los diagramas que se representaron fueron:

- Diagrama de flujo de datos de nivel 0 o modelo fundamental del sistema,
- diagrama de flujo de datos de nivel 1, y
- diagrama de flujo de datos de nivel 2.

La notación básica utilizada para representar un diagrama de flujo de datos es mostrada en la tabla 4.1 [8].

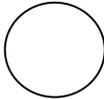
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Entidad externa	Representa unidades externas al sistema que se comunican con él y que están fuera de su control.
	Proceso	Un transformador de información (una función) que reside dentro de los límites del sistema a ser modelado.
	Flujo de datos	El flujo describe el movimiento de paquetes de datos de una parte del sistema a otra.

Tabla 4.1. Notación básica para los diagramas de flujo de datos.

Modelo fundamental del sistema

Con este diagrama de nivel 0 ó modelo fundamental del sistema mostrado en la figura 4.3, se representa en forma gráfica al software completo como un solo proceso, con datos de entrada y de salida, estos últimos representados por flechas. Los datos y órdenes de entrada son proporcionados por el usuario, el software representado por un solo proceso ejecuta lo solicitado por él, y envía los datos procesados a una entidad de salida que en este caso es un monitor, donde se pueden visualizar de forma gráfica y tabulada los resultados.

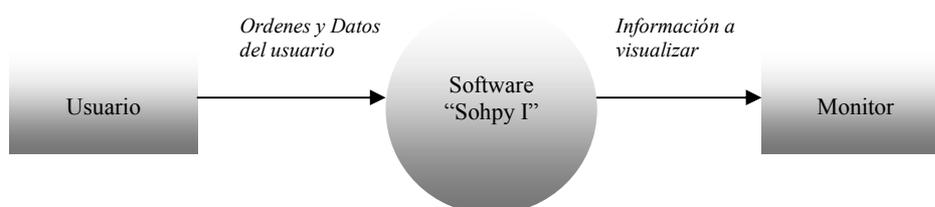


Figura 4.3. Diagrama de flujo de datos a nivel 0 del software "Sophy I".

Diagrama de nivel 1

A continuación se muestra el diagrama de nivel 1 en la figura 4.4, en este diagrama se particiona el diagrama de flujo de datos de nivel 0 para mostrar más a detalle el sistema en general. En el nivel 1, aparecen representados procesos y caminos de flujos de información adicionales, representando las sub-funciones del sistema general.

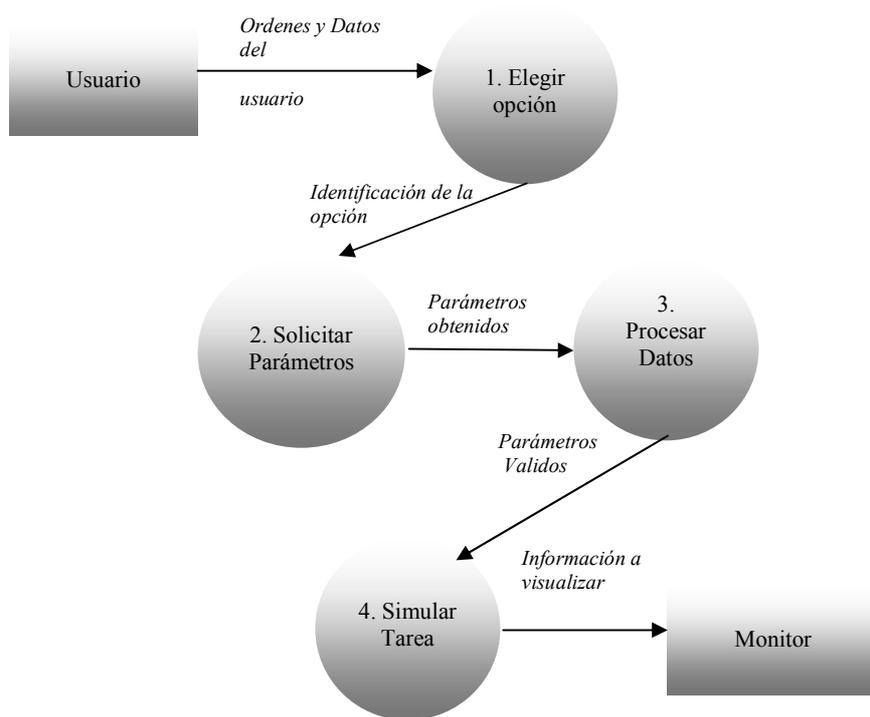


Figura 4.4. Diagrama de flujo de datos a nivel 1 del software "Sophy I".

Diagrama de nivel 2 que refina el proceso "elegir opción"

En este nuevo diagrama de nivel 2, se representa gráficamente y más a detalle el proceso 1, del DFD de nivel 1, el cual fue nombrado como "1. Elegir Opción". En el presente diagrama se muestran las subfunciones que intervienen en el proceso "Elegir Opción". Así, como también los flujos de datos que son enviados y recibidos en cada una de las subfunciones mostradas. El diagrama de nivel 2 es mostrado en la figura 4.5.

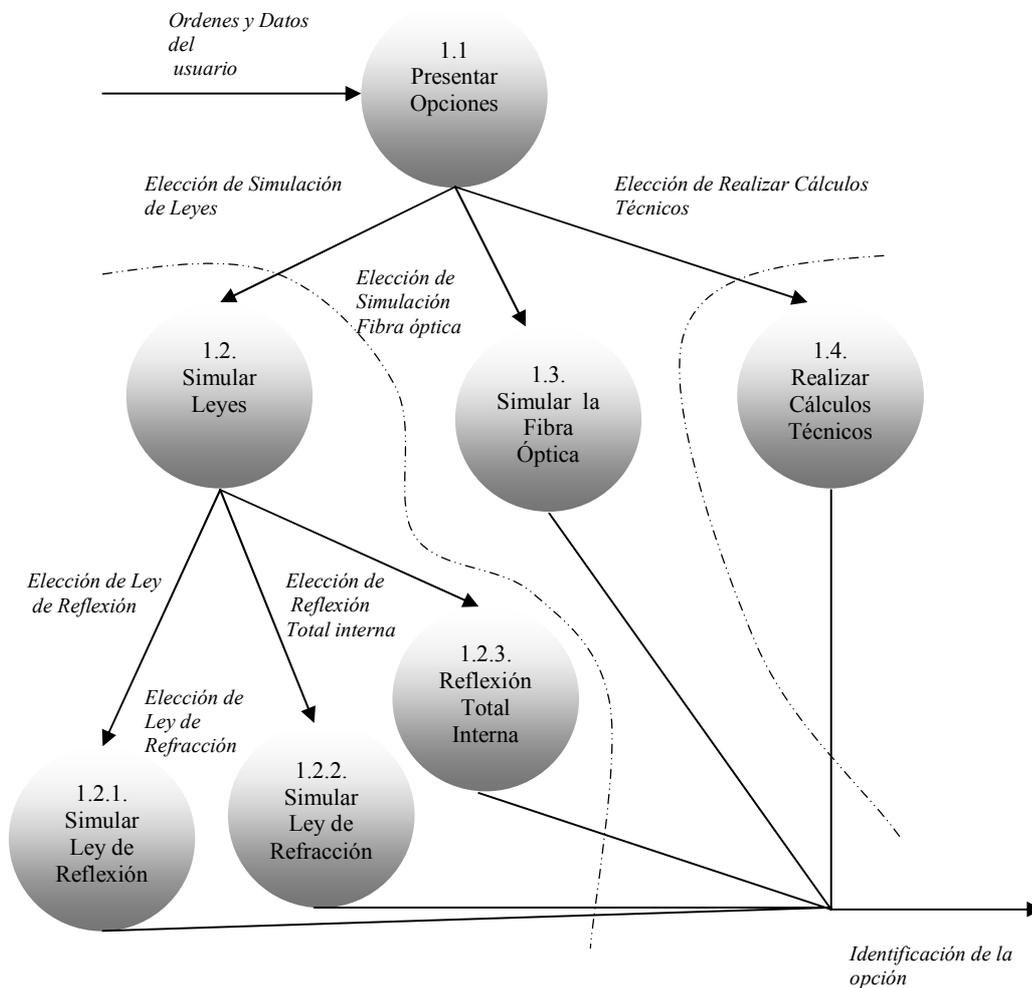


Figura 4.5. Diagrama de flujo de datos de nivel 2 del proceso "Elegir Opción".

En este diagrama también se muestran los centros de transformación, especificando los límites de los flujos de entrada y salida. Lo cual significa, que del diagrama de flujo de datos, se limitan los procesos que puedan ser considerados como una sola parte de toda la estructura de un sistema. En el diagrama mostrado en la figura 4.5, se formaron ciertos centros de transformación, marcados con una línea punteada. De los cuales uno de ellos abarcaba el proceso 1.2, y los subprocesos que se derivaron de éste, como son el subproceso 1.2.1, 1.2.2 y 1.2.3. Los otros centros de transformación se limitaron únicamente a un solo proceso.

4.6. DIAGRAMA GENERAL DEL SISTEMA

Tomando como base los diagramas de flujo de datos, y principalmente los centros de transformación, se estableció el diseño final de la estructura del software. Teniendo como resultado final el diagrama general del software “Sophy I”, el cual es un diagrama de estructura representando al sistema completo a través de módulos que se encargan de realizar tareas específicas. La notación básica para la elaboración de los diagramas de estructura es mostrada en la tabla 4.2 [8].

SÍMBOLO	NOMBRE
	Modulo
	Flujo de datos

Tabla 4.2. Notación básica para diagramas de estructura.

Para el sistema diseñado, cada uno los módulos representan las secciones que conforman el software “Sophy I”. El diagrama de estructura es mostrado en la figura 4.6.

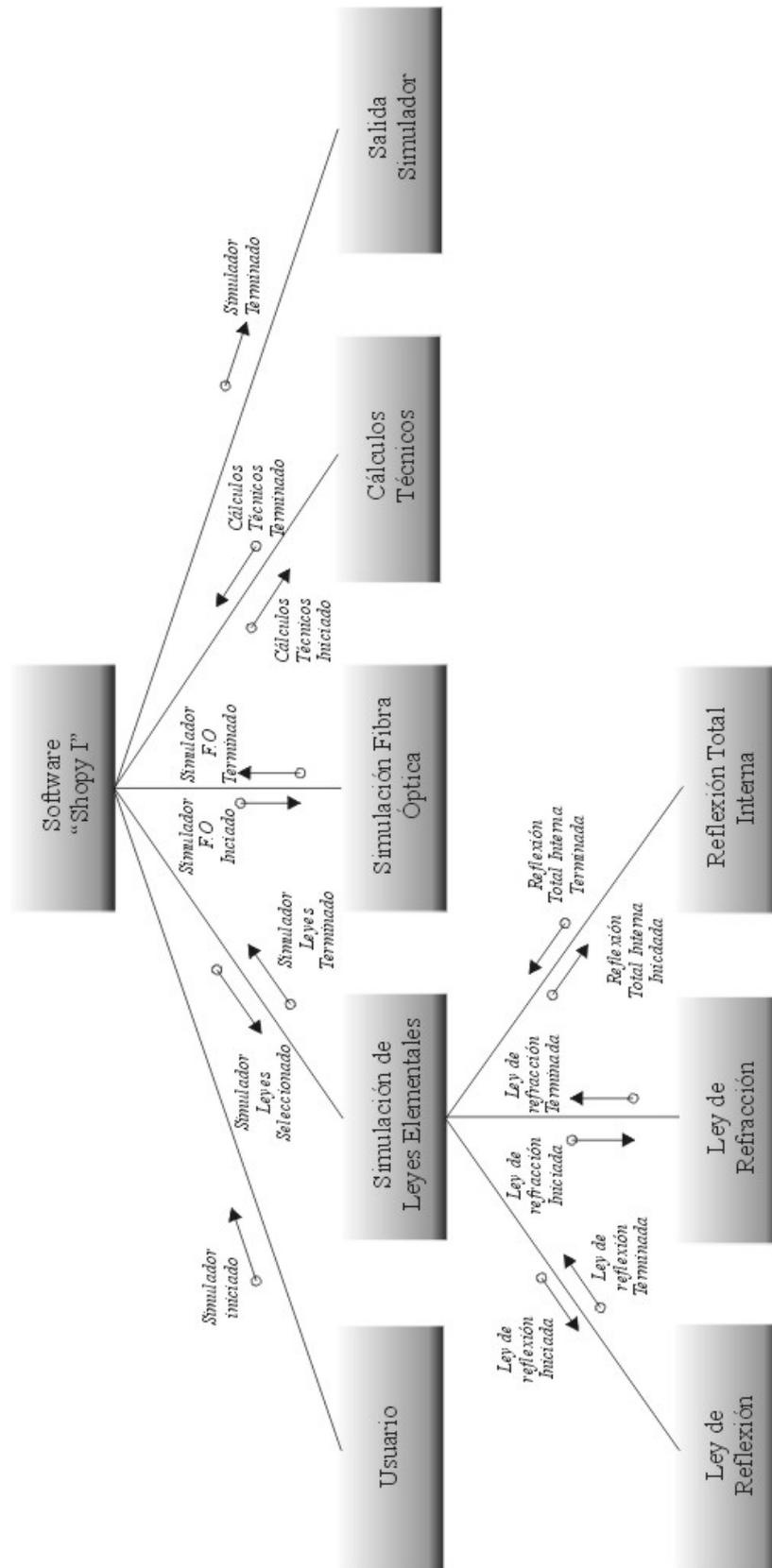


Figura 4.6. Diagrama general del sistema.

4.7. MÉTODO DE DESARROLLO

El método utilizado para el desarrollo de esta aplicación fue el método en espiral como ya antes se había mencionado en el tercer capítulo de este proyecto.

El software fue desarrollado e implementado para llegar a satisfacer a un gran número de usuarios, pero la supervisión del mismo se efectuó principalmente por la M.C Maribel Tello Bello, profesor investigador del Instituto de Electrónica y Computación. Esta supervisión fue necesaria por ser ella el cliente principal, además de tener los conocimientos necesarios y especialización en el área de estudio de la "Fibra Óptica".

El número de iteraciones realizadas para este proyecto fueron en total seis, las cuales a continuación se describen muy brevemente :

1ª. Iteración

El tiempo invertido en la 1ª. Iteración fue proporcionalmente mayor con respecto al dedicado a las demás iteraciones realizadas. Esto se debió a que para iniciar con la primer etapa del método de desarrollo, la cual es una captura de requisitos y planificación inicial, fue necesario adentrarse al tema "Principios y Funcionamiento de la Fibra Óptica", con la finalidad de entender mejor al cliente y poder establecer una comunicación fluida con él.

Un punto que es importante recalcar, es que, aunque la investigación de la herramienta de desarrollo no forma parte del método, el estudio de ésta y la puesta en práctica del Método de Espiral se fueron llevando en forma paralela. Esto sin obstruir ninguna de las iteraciones del método. El lenguaje de programación que se utilizó como herramienta de desarrollo, y el cual se mantuvo en estudio fue OpenGL, mismo que es explicado en el siguiente capítulo.

Una vez obtenida la primera captura de requisitos y una planificación inicial, se prosiguió a realizar un análisis. De este primer análisis se obtuvo la identificación del problema a resolver y se estructuró en forma general el software. La estructura del software, consistió en identificar las diversas funciones que tendría que realizar para que llegara a ser considerado como una herramienta de apoyo para el estudio de los efectos ópticos en una fibra. Una vez identificadas, se clasificaron según su importancia para que al final se obtuviese como resultado el software estructurado en tres secciones principales:

- I. Simulación de leyes elementales.
- II. Simulación del confinamiento de la luz en la fibra óptica, y
- III. Sección de cálculos técnicos.

Los principales riesgos que se identificaron en esta primer etapa, los cuales se resolvieron a la brevedad posible, fueron: el desconocimiento parcial de la tecnología de “Fibra Óptica” y el desconocimiento de la herramienta de desarrollo. La forma en que se le brindo solución a cada uno de ellos fue: para el primer riesgo mediante la investigación a fondo y autoestudio del tema, así como también constantes asesorías con el experto en fibra óptica. El segundo riesgo fue resuelto únicamente de forma autodidacta, basándose en Internet, libros y artículos de revistas.

En esta iteración se presentó un prototipo inicial de toda la estructura del sistema. Dicho prototipo fue presentado en papel únicamente, con la finalidad de que el cliente aprobara la idea mostrada. Así mismo, se presentó un pequeño prototipo implementado ya en el lenguaje OpenGL. El pequeño prototipo mostraba en forma general como se llevarían a cabo las simulaciones en escenarios tridimensionales con gráficos animados. El prototipo únicamente mostraba la simulación de la ley de reflexión, el cual es mostrado en la figura 4.7.

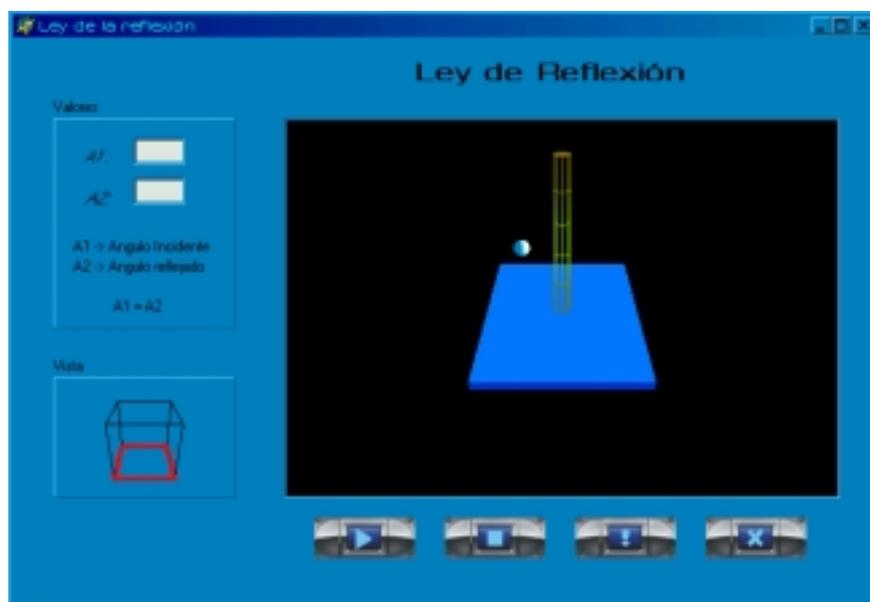


Figura 4.7. Prototipo de la ley de reflexión implementado en OpenGL .

Las evaluaciones hechas a los prototipos fueron aceptadas por el cliente. Aunque para el prototipo de Simulación de la Ley de Reflexión, el cliente sugirió algunas modificaciones como son: incluir etiquetas a las partes de la simulación, así como también una línea punteada que indicará la normal a la superficie de reflexión, y otra línea para representar la trayectoria del rayo.

2ª. Y 3ª. Iteración

Estas dos iteraciones tuvieron un proceso muy similar entre ellas, pues en ambas se llevaron a cabo las mismas actividades, además de que en cada una de éstas se desarrolló un prototipo implementado con OpenGL. En la segunda iteración, se implementó la simulación de la ley de refracción, y en la tercera iteración se implementó el fenómeno de reflexión total interna.

En estas iteraciones como en las siguientes, ya no se presentaron otro tipo de riesgos diferentes a los ya identificados en la primera iteración. Así que, conforme surgían riesgos de desconocimiento de nuevos temas relacionados con la tecnología de la Fibra Óptica, o de la forma de utilización de la herramienta de desarrollo, éstos se resolvían, investigando y manteniendo una constante comunicación con personas relacionadas con el tema.

4ª. Iteración

En esta iteración ya se tenían bases suficientes para pasar de un simple prototipo en papel, a un prototipo implementado en OpenGL. Con ello la primera sección del software “Sophy I” había sido codificada por completo, hasta antes de empezar esta cuarta iteración. A pesar de que los principios de graficación y animación en escenarios 3D con OpenGL ya habían sido comprendidos y analizados. Para la implementación de la segunda sección del simulador de fibra óptica, se requería hacer uso de algunos efectos más especializados de gráficos en OpenGL. Unos de estos efectos consistía en crear figuras geométricas en 3D con un texturizado transparente. Este efecto y otros, fueron investigados y asimilados para poder llevar a cabo la implementación de tal sección.

Una vez resueltas las dificultades encontradas, se concluyó la implementación de la segunda sección del software “Sophy I”, la cual era la simulación del confinamiento de la luz dentro de la fibra óptica. Dicha sección es mostrada en la figura 4.8, la cual va a ser tratada más a detalle en el capítulo siguiente.

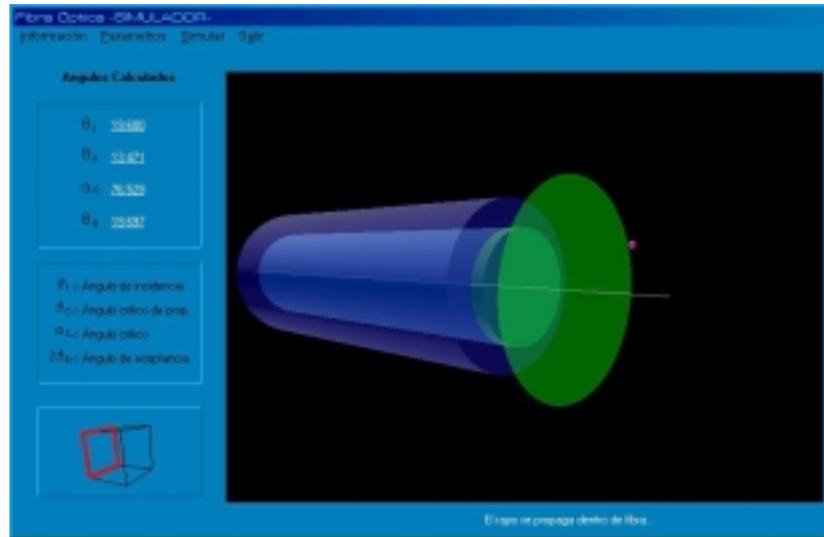


Figura 4.8. Prototipo de la ley de reflexión implementado en OpenGL .

5ª. Iteración

En esta iteración se llevó a cabo el desarrollo e implementación de la última sección, la cual consiste en el manejo de algunos cálculos técnicos. En esta sección ya no intervino la programación gráfica, pues solo era información técnica de la fibra óptica multimodo. La selección de esta fibra óptica obedece al hecho de que todos los sistemas que se establecieron antes de la aparición de la fibra monomodo, se realizaron con la fibra multimodo, además de que los cambios en cuanto al análisis que se presenta en esta sección son mínimos.

Una vez concluida esta sección, fue presentada al cliente para ser evaluada y corregida en determinadas partes.

En la captura inicial y planificación de esta iteración, se estudió mas acerca de algunos aspectos de la comunicación vía fibra óptica, esto con el fin de implementar cálculos que pudieran ser útiles para las personas interesadas. Dentro de los temas estudiados para esta sección, se mencionan algunos de ellos:

- Pérdidas en la fibra óptica
- Dispersión intermodal e intramodal
- Longitud de onda
- Empalmes y Conectores

6ª. Iteración

En esta última iteración se presentó al cliente el proyecto completo, el cual constaba de las tres secciones ya antes mencionadas. Así también, en esta iteración se diseñó e implementó la interfaz principal que tendría que presentar el software. La interfaz final como cada una de las secciones que comprende este software, serán explicadas mas a detalle en el capítulo siguiente.

En esta iteración se probó cada una de las secciones que conformaban el software, desde la sección de simulación de leyes, la simulación del confinamiento de la luz y la sección de cálculos técnicos, y se encontraron algunos problemas que en su mayoría fueron solo de validación de datos. Estos problemas fueron resueltos completamente, teniendo así, el producto terminado.

4.8. CONCLUSIÓN DEL DISEÑO

Para llevar a cabo el diseño del sistema fue necesario elaborar una captura inicial de requerimientos (por medio del cual se obtuvo una idea concreta sobre lo que debería de realizar el sistema), lo anterior se tomó como referencia para realizar una descripción formal del software. Considerando esta descripción formal y tomando en cuenta la ayuda de algunas herramientas de diseño como los diagramas de flujo de datos, centros de transformación y diagramas de estructura, se procedió con el modelo de desarrollo en espiral, para que la creación de este proyecto cumpliera con la calidad que brinda la Ingeniería de Software.

Una vez formulada la estructura general del sistema, ya se está en condiciones de implementar el diseño que ha sido planteado.

Valorando lo que se ha presentado en este capítulo, a continuación se indican los elementos necesarios para llevar a cabo la implementación del sistema.

CAPÍTULO V

IMPLEMENTACIÓN DEL SOFTWARE

5.1. IMPLEMENTACIÓN TEÓRICA

Hasta este capítulo ya se han tratado las bases teóricas necesarias tanto de conceptos relacionados con la fibra óptica como de los lineamientos que serán de utilidad para llevar a cabo el desarrollo e implementación del software. Por una parte se ha cubierto la teoría necesaria para estudiar los efectos ópticos en una fibra, al igual que algunos cálculos técnicos elementales utilizados en esta tecnología, mientras que por otra parte se ha analizado el modelo de desarrollo a utilizar para llevar a cabo este proyecto y se ha plasmado el diseño que da origen a la herramienta "Sophy I". En el modelo de desarrollo utilizado (Modelo en espiral) al igual que en otros modelos se debe de llevar a cabo una etapa muy importante y necesaria para la creación de un software, siendo ésta la etapa de implementación. La forma de implementación de este software puede ser clasificada en dos partes principales: la primera que trata de utilizar la teoría básica presentada en el capítulo II para efectuar implementaciones de cálculos necesarios y vistos en la descripción formal en el capítulo anterior, y la segunda es la forma de implementar los resultados de dichos cálculos a través de gráficos. Para la implementación de la primera parte no existe ninguna dificultad, pues ésta podría ser realizada en cualquier lenguaje de programación que permitiera realizar operaciones aritméticas, pero para la segunda parte el lenguaje a utilizar ya necesita de características especiales, siendo una de ellas y la primordial la realización de gráficos, solo que para el objetivo del proyecto los gráficos a presentar deben ser en tercera dimensión, es por ello que se utilizaron las herramientas de programación nombradas a continuación.

5.2. HERRAMIENTAS DE DESARROLLO

Para la implementación del software se optó por usar la herramienta de desarrollo Delphi. Así como también, la librería de gráficos 3D OpenGL. Ambos, lenguaje de programación y librería, fueron elegidos por las siguientes razones:

- Delphi es una herramienta que permite realizar aplicaciones con interfaces gráficas amigables para el usuario. Además de poder utilizar la programación de Object Pascal de Borland, la cual es muy sencilla de manejar.
- La librería de gráficos 3D OpenGL, contiene una serie de primitivas para dibujar gráficos en dos y tres dimensiones, incluyendo un gran número de funciones para dar formato a las imágenes.

Para la implementación de este software se requirió del apoyo de un lenguaje que permitiera ver las simulaciones de una manera gráfica y más amena para el usuario. Esto con la finalidad que tal simulación llegara a ser representada en una forma más clara y apegada a la realidad mostrándose en escenarios tridimensionales.

La librería utilizada proporciona una serie de funciones para crear figuras geométricas, con la cual se puede llegar a construir modelos bidimensionales o tridimensionales. Ahora bien como OpenGL es una librería, en cuestión de interfaz es necesario hacer uso de algún lenguaje de programación. El lenguaje de programación que se eligió para esta tarea, fue Object Pascal de Borland, dicho lenguaje se utilizó para poder realizar e implementar los cálculos que se requerían en la simulación.

En la actualidad la programación se lleva a cabo con la ayuda de herramientas RAD (Rapid Application Development). Delphi es una de estas herramientas que esta basada en el lenguaje de programación Pascal, la cual puede ayudar a los programadores a desarrollar aplicaciones, evitando la creación de elementos secundarios como son botones, ventanas, menús, cuadros de texto, entre otros.

5.3. Implementación de OpenGL en Windows

En Windows, OpenGL es implementado con los archivos `opengl32.dll` y `glu32.dll`, los cuales contienen los principales comandos de gráficos [18]. Ambos archivos son utilizados únicamente para versiones de 32-bits de Windows, como por ejemplo NT, 95 o Superiores.

Los dos archivos DLL (Dynamic-Link Libraries) contienen un poderoso conjunto de rutinas con las cuales se pueden producir impresionantes escenarios animados en tres dimensiones.

Por lo general, los archivos DLL antes mencionados ya vienen incluidos en las versiones de Windows a partir de la versión 95 en la ruta

`C:\>WINDOWS\SYSTEM`

Pero si alguno de estos archivos no se localizara en el directorio indicado, es posible obtenerlos de Internet y almacenarlos en la ruta correspondiente.

5.4. Implementación de OpenGL en Delphi

Una vez instalados los archivos DLL correspondientes, se continua colocando un archivo cabecera en las librerías del lenguaje de programación Delphi, para que se puedan utilizar las diferentes funciones de OpenGL pero aplicadas en el lenguaje antes citado.

El archivo que Delphi necesita para reconocer aplicaciones que incluyan tipos de datos y funciones de OpenGL es la unidad de interfaz llamada `OpenGL.dcu`. Esta unidad no es mas que un archivo donde se encuentran todas las rutinas necesarias para dibujar esferas, cilindros, discos, entre otros.

El archivo puede ser obtenido de Internet del sitio:

<http://www.opengl.org>

Y debe ser colocado en la siguiente ruta:

`C:\>Archivos de Programas\Borland\Delphi5\lib`

Hasta este punto se han analizado los requisitos que fueron necesarios para la implementación del Software, con el fin de poder utilizar la librería OpenGL bajo la plataforma de Windows y utilizando el lenguaje visual Delphi.

En el siguiente apartado, se presentan algunas de las interfaces creadas bajo estas herramientas.

5.5. Interfaces Gráficas del Software “Sophy I”

A continuación se explican brevemente las interfaces más importantes que utiliza este software. En cada una de las interfaces, se mencionan únicamente los parámetros que son necesarios para poder ejecutar la simulación y los valores que calcula automáticamente el software.

Las funciones elementales para el cálculo de algunos parámetros imprescindibles utilizados en las secciones de este software, al igual que determinadas funciones para la implementación de las animaciones utilizando la librería OpenGL, son expuestas en el apéndice B de este documento. Mientras que la forma de utilizar los elementos que se muestran en las interfaces, así como la explicación más detallada del funcionamiento de cada una de ellas, es mencionada en el manual de usuario, incluido en el apéndice C.

A continuación se explicarán en forma breve, cada una de las interfaces de las secciones en que está dividido el software.

5.5.1. SECCIÓN DE SIMULACIÓN DE LEYES ELEMENTALES

Cuando se elige la primera sección del menú principal, el software presenta una interfaz en donde se puede elegir entre la demostración de la ley de reflexión, la ley de refracción o el fenómeno de reflexión total interna.

Ley de reflexión

La interfaz para la simulación de la ley de reflexión es mostrada en la figura 5.1. Teniendo presente esta pantalla el usuario puede introducir el ángulo de incidencia y al ejecutar la simulación, se podrá observar en forma gráfica dicho fenómeno, además de mostrar el único valor calculado que es el ángulo de reflexión.

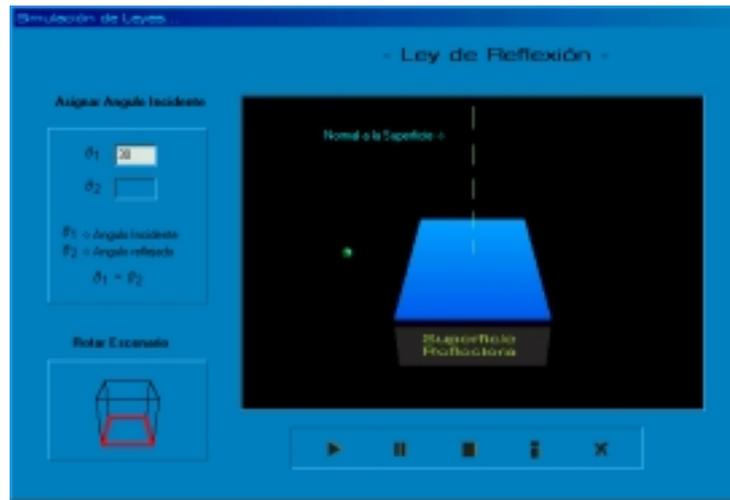


Figura 5.1. Interfaz para la simulación de la ley de reflexión

Ley de refracción

Para realizar esta simulación, se implementó la interfaz mostrada en la figura 5.2. Los parámetros que deben ser introducidos por el usuario son: los índices de refracción n_1 y n_2 , además del ángulo de incidencia θ_1 . Con estos parámetros se ejecuta la simulación y se realiza el cálculo correspondiente al ángulo de refracción θ_2 .

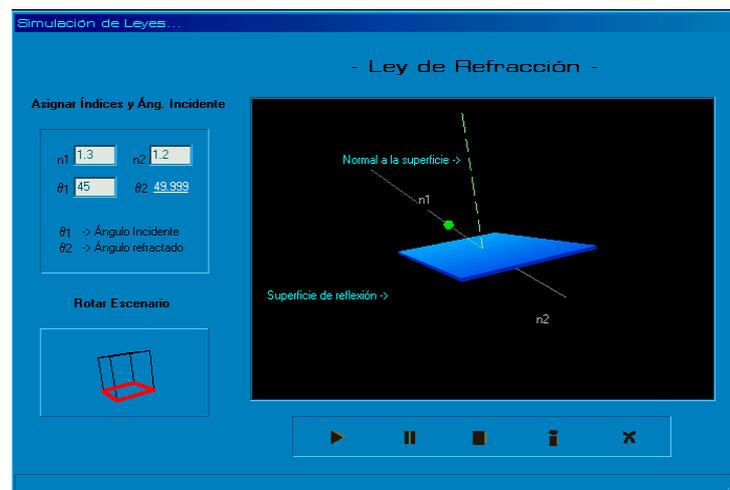


Figura 5.2. Interfaz para la simulación de la ley de refracción.

En la simulación de esta ley el sistema realiza el cálculo de un ángulo llamado “ángulo crítico” θ_c , el cual no es visible por el usuario. En donde, si el ángulo de incidencia es mayor al ángulo crítico calculado, esta simulación no se ejecutará, ya que con dichos valores del ángulo de incidencia y los índices de refracción, el rayo de luz se reflejará en la superficie, ocasionando un caso especial del fenómeno de Reflexión Total Interna.

Fenómeno de reflexión total interna

La interfaz implementada para la simulación del fenómeno de reflexión total interna es mostrada en la figura 5.3.

Los parámetros que intervienen en esta simulación son; los índices de refracción n_1 y n_2 , y el ángulo de incidencia θ_1 . El valor de éste último, debe ser mayor al ángulo crítico calculado por el software, la consideración antes mencionada es para que la demostración del fenómeno de reflexión total interna se pueda llevar a cabo, ya que este fenómeno solo se presentará si el ángulo de incidencia es mayor al ángulo crítico, produciendo una reflexión del rayo de luz analizado, de lo contrario se presentará la ley de refracción, es decir, la luz atravesará la superficie.

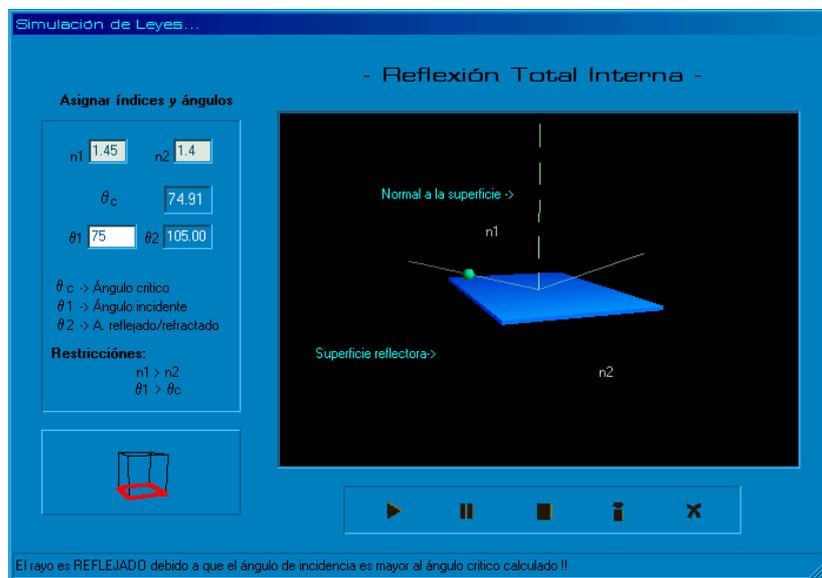


Figura 5.3. Interfaz para la simulación del fenómeno reflexión total interna.

5.5.2. SECCIÓN DE SIMULACIÓN DE LA FIBRA ÓPTICA

Para la simulación de la fibra óptica, se implementó la interfaz mostrada en la figura 5.4. Esta interfaz, cuenta con un área en donde se muestra la simulación gráficamente, otra donde se muestran los ángulos de mayor importancia (ángulo de incidencia, ángulo crítico de propagación, ángulo crítico y ángulo de aceptación) que fueron calculados por el software y un menú de opciones. El menú de opciones es la única forma con la cual, el usuario puede interactuar con la presente sección, es por ello, que se describe cada una de las opciones con las que cuenta el menú:

- Una opción para obtener ayuda acerca de temas afines a la sección (definición de la FO, tipos de fibra, tipos de rayos, entre otros).
- Una opción principal para la simulación de la fibra óptica, en donde se introducen los parámetros que son necesarios para poder llevar a cabo tal simulación. Los parámetros que el usuario introduce son: los índices de refracción del núcleo y la cubierta, y el ángulo de incidencia del rayo que será representado.
- En esta sección no se incluyeron como componentes los controles de la simulación gráfica (ejecutar, pausar, detener, etc), como en las secciones ya antes vistas, esto por cuestiones de diseño en la interfaz. Ahora los controles fueron representados y colocados dentro de una opción del menú que incluye esta sección. Entonces, en otra opción mas del menú actualmente descrito se incluyen los controles para la simulación de la fibra óptica (Ejecutar y Detener).

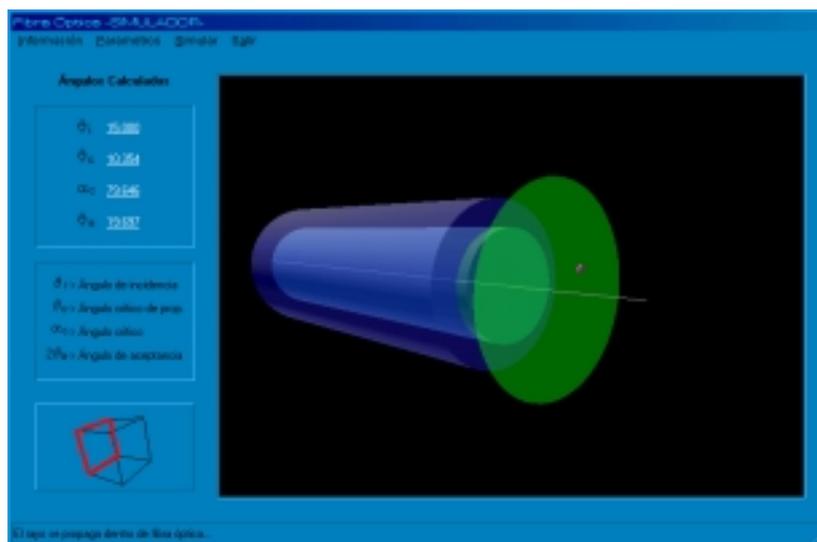


Figura 5.4. Interfaz para la simulación de la luz que viaja a través de la fibra óptica.

La simulación de esta sección consiste en un rayo de luz que viaja a través de la fibra óptica, la trayectoria de éste es controlada por varios factores como son: los índices de refracción tanto de la cubierta como del núcleo de la fibra óptica. Estos índices son asignados por el usuario, con estos datos el software calcula automáticamente la apertura numérica y el ángulo de aceptación, en donde éste último es representado con un gráfico tridimensional como un cono. Una vez teniendo presente el ángulo de aceptación, el usuario puede asignar el ángulo al cual incide el rayo en el centro del núcleo de la fibra. El valor de este último debe de estar dentro de los límites del ángulo de aceptación, esto debido a que, los rayos con un ángulo de incidencia mayor o fuera de los límites, saldrán de la fibra óptica representando la pérdida de información en la fibra. Mientras que los rayos con ángulos de incidencia menores al ángulo de aceptación se propagarán dentro de la fibra óptica.

5.5.3. SECCIÓN DE CÁLCULOS TÉCNICOS

La última sección del simulador, es donde el usuario puede procesar algunos cálculos técnicos para la construcción de sistemas de comunicación vía fibra óptica.

Los parámetros que son introducidos por el usuario en esta sección son:

- La potencia de entrada, cuyas unidades son manipuladas en Microwatts (μW)
- Los índices de refracción del núcleo y la cubierta de la fibra óptica
- La longitud de onda λ que presenta la fibra óptica en uso, cuyas unidades están dadas en Nanómetros (nm).
- El radio del núcleo, cuyas unidades son manipuladas en Micrómetros (μm s), y
- La longitud total de la fibra óptica, dada en Kilómetros (Km).

Los cálculos que genera esta sección son:

- Pérdidas totales. Estas pérdidas dependen del número de conectores y empalmes que se están utilizando para la construcción del sistema de comunicaciones, así como la pérdida que presenta cada uno de éstos. Además, de la longitud de la fibra óptica, y la longitud de onda con que trabaja dicha fibra.
- Potencia de salida. Teniendo presente la pérdida total del sistema y la potencia de entrada se obtiene la potencia de salida que el sistema entregará.

- Número de modos (MN). Este parámetro es calculado en base a los valores de la apertura numérica N.A, la longitud de onda λ que presenta la fibra óptica en estudio, y el radio del núcleo de ésta.
- Dispersión intermodal. Para la realización de este cálculo, se requieren de los mismos parámetros utilizados para el cálculo del número de modos (apertura numérica, longitud de onda, y el radio del núcleo), además de la longitud de la fibra óptica.

Algunos de los conceptos presentados en los cálculos que realiza el software, y parámetros requeridos, fueron explicados teóricamente en el capítulo 2 de este documento.

La interfaz implementada para llevar a cabo los cálculos antes mencionados, es mostrada en la figura 5.5.



Figura 5.5. Interfaz para la sección de cálculos técnicos.

5.6. Conclusión de la Implementación

En este capítulo se han señalado de forma integral los puntos más importantes para llevar a cabo la implementación del sistema, incluyendo una descripción y justificación del lenguaje de programación Delphi, quien combinado con la librería de gráficos 3D OpenGL, fueron utilizados para la codificación e implementación del sistema.

Por otra parte este capítulo presentó de forma breve y concisa, las interfaces gráficas de las que esta constituido el software de simulación.

Como punto final de este documento se presenta un último capítulo en donde se explican las pruebas que se realizaron en cada una de las iteraciones, las mejoras que se le pueden asignar a dicho software en un futuro, y unas conclusiones generales de todo el proyecto.

CAPÍTULO VI

PRUEBAS Y CONCLUSIONES

6.1. Pruebas

Este proyecto fue desarrollado bajo la metodología en espiral, la cual como ya se conoció anteriormente consta de diversas iteraciones. Para la construcción del simulador, se llevaron a cabo determinado número de iteraciones, en las cuales se fueron implementando uno a uno los módulos que iban conformando el sistema. Es decir, en cada iteración se implementaron cada una de las secciones del software de simulación, expuestas en el capítulo de diseño de este documento.

En cada una de las iteraciones del método, se realizaron cierto tipo de pruebas con la finalidad de incrementar el factor de confiabilidad, disminuyendo el número de errores del sistema.

Las pruebas que se realizaron fueron de dos tipos, de verificación y de aceptación. Las primeras fueron realizadas por la persona que portó el papel de programador en este proyecto y las segundas fueron realizadas por el cliente al que se le desarrolló el software.

Las pruebas de verificación consistieron en probar cada uno de los módulos implementados, tanto de forma externa o superficial y de forma interna. Las pruebas en forma externa fueron hechas en primera instancia por el programador, en las cuales se comprobó que dependiendo de los datos de entrada (parámetros), el software generaba determinados datos de salida (cálculos automáticos y simulación gráfica), conociéndose este tipo de pruebas como “Pruebas de Caja Negra”.

Dentro de las pruebas de verificación hechas por el programador, se realizaron pruebas internas o mejor conocidas como “Pruebas de Caja Blanca”, en estas se verificó en forma minuciosa el funcionamiento interno del código de cada uno de los módulos.

Ya terminadas las pruebas de verificación antes mencionadas a cada uno de los módulos del sistema, se detectaron algunos errores los cuales fueron corregidos. Obteniendo así, una primera versión del sistema, la cual posteriormente fue presentada al cliente, quien realizó sus propias pruebas (llamadas “Pruebas de Aceptación”).

En las pruebas de aceptación el cliente probó cada uno de los módulos del sistema, tomando en cuenta únicamente los datos de entrada y verificando la salida que arrojaba el sistema.

Las pruebas de aceptación al igual que las pruebas de verificación fueron realizadas en cada una de las iteraciones del desarrollo del software. Por lo tanto, el cliente realizó pruebas de aceptación a las secciones que iban siendo implementadas, de la misma forma a los módulos que formaban parte de alguna de estas secciones, como fue la “Sección de Simulación de Leyes Elementales”, en la cual el cliente verificó la implementación de la Ley de Reflexión, Ley de Refracción y Reflexión Total Interna. Una vez terminadas este tipo de pruebas, se identificaron algunas correcciones mínimas en cuestiones de interfase y cálculo de parámetros, las cuales fueron resueltas a la brevedad posible, obteniendo como resultado un software de apoyo para el estudio de los efectos ópticos en una fibra y con la mayor calidad posible.

6.2. Perspectivas

El desarrollo de tecnología electrónica en nuestro país es muy escaso, esto por varios factores ya sea políticos, económicos, culturales, y demás. Provocando un atraso tecnológico que puede acarrear consecuencias en varios ámbitos, uno de los tantos problemas que ocasiona tal atraso es el financiamiento para el desarrollo de hardware o software. Sin embargo, a pesar del presente problema, algunos científicos, investigadores y estudiantes aportan algunos proyectos en este campo donde la mayoría de estos son dispositivos o herramientas de hardware contando con una aportación mínima de software. Es por esto, que es importante enfocar nuevas aplicaciones de software para el apoyo y avance en este campo.

El sistema desarrollado en esta tesis puede continuar su desarrollo como por ejemplo en la sección de cálculos técnicos, incluyendo algunos catálogos de conectores y empalmes, así como también, catálogos de fuentes de luz y receptores. Con el fin de que el sistema pueda presentar mayor comodidad al usuario, sin necesidad de que este tenga que investigar algunos parámetros que requiere la sección de cálculos técnicos.

En sí, todo el software principalmente las secciones de la simulación de las leyes elementales y la fibra óptica, pueden ser migrados a otros sistemas operativos pues para la programación gráfica se utilizó la librería OpenGL, la cual es considerada multiplataforma, y así como existen archivos para trabajar con esta librería bajo la plataforma de Windows, también existen para otros sistemas operativos como Unix.

6.3. Conclusiones

Al término de este proyecto se llegó a cubrir el objetivo planteado al principio, el cual fue crear un software de apoyo para el estudio de los efectos ópticos en una fibra cuyas simulaciones fueran representadas con gráficos animados en 3D. Teniendo como principal aportación en el campo de la electrónica una herramienta que facilita el estudio de la fibra óptica.

Por ende, este software permite obtener un análisis y una simulación del comportamiento de la fibra óptica, sin la necesidad de utilizar tecnología de instrumentación, la cual si existe pero el precio para su adquisición es muy costoso.

Al concluir este proyecto, se ha visto que se han abarcado y conjugado un determinado número de conocimientos de diferentes áreas científicas y tecnológicas. Por una parte, se consideraron los conocimientos del área de la electrónica, especialmente de las tecnologías usadas en las comunicaciones, en particular las de la fibra óptica, la cual hoy en día, es uno de los medios más confiables y seguros para la transmisión de información. Y por otra parte, se tienen los conocimientos del área computacional, la cual proporciona un gran número de herramientas para representar soluciones de problemas a través de software. Al combinar la computación con la electrónica, se está logrando ampliar el campo de aplicación de esta área, ya que en un inicio fue enfocada únicamente al campo empresarial. Por todo esto se puede decir que se ha hecho un logro más en las áreas antes mencionadas, pero propiamente se ha conseguido un mejor logro en el área profesional, pues con este proyecto se está afirmando que como un profesional en computación puede hacerse uso de las herramientas de desarrollo de software y aplicarlas en cualquier otra área, para resolver algún problema en particular.

Con la culminación de este proyecto, se ha podido demostrar que ha sido un trabajo realizado aplicando la mayoría de los conocimientos adquiridos en la universidad, desde conocimientos de tipo científico y tecnológico, hasta ciertos valores que se fueron adquiriendo conforme se fue creciendo como persona y principalmente como profesionista.

BIBLIOGRAFÍA

- [1] Ardisson M., *Ingeniería de software: Ciclo de vida del desarrollo del software*, IIMAS-UNAM, Septiembre de 1988, 21 pag.
- [2] Anubis, *Metodologías de reutilización del software*, <http://anubis.ecci.ucr.ac.cr/~ci1402/p/1998-13.htm>, última fecha de acceso: 28 de Enero de 2003.
- [3] Berzins V., *Languages for Specification, Design and Prototyping*, Modern Software Engineering. P.A. Ng, R.T. Yeh., Van Nostrand Reinhold, 1990.
- [4] Biggerstaff T., *Special issue on software reusability*. IEEE Trans. Software Engineering, Vol 10 SE-10 N° 5, Sep1984.
- [5] Bostock S., *Course Engineering: An overview of the courseware development process*, Keele University, Febrero de 1996.
- [6] Butler R. W., *Formal Methods for Life-Critical Software*. AIAA Computing in Aerospace 9th Conference. San Diego, 1993.
- [7] Carni P. L. Et. Al., *Optical fiber communication*, McGraw-Hill Book Company, U.S.A, 1980, 883 pag.
- [8] Centro de Computación Profesional de México, *Análisis y diseño de sistemas*, McGraw-Hill, Agosto 2001, 179 pag.

- [9] Community Borland, *Borland Developer Network*,
<http://www.community.borland.com/delphi>, última fecha de acceso: 16 de enero de 2002.
- [10] Delphi3d, *OpenGL for Delphi 5*, <http://www.delphi3d.net/>, última fecha de acceso: 16 de enero de 2002
- [11] Departamento De Innovación, *Herramientas CASE para el análisis y diseño estructurado*, Subdirección de sistemas –DCCA- UNAM, Agosto de 1998.
- [12] Garlan D. and Shaw M., *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall, April 1996.
- [13] Catalogo de software, *Software de desarrollo-Desarrollo*,
<http://www.catalogodesoftware.com/software/AreasE.asp?aid=96>, última fecha de acceso: 7 de Marzo de 2003.
- [14] Gonzáles D. J., *Integración de un modelo de proceso sistemático en el desarrollo de software educativo*, Tesis Doctoral, Depto. de Informática, Universidad de Vigo, Vigo España, Septiembre 2002.
- [15] Gonzalo C. A., *Ingeniería de software: Práctica de la programación*, Addison Wesley Iberoamericana S.A., Madrid España, 1993, 519 pag.
- [16] Hecht E. y Zajac A., *Óptica*, Addison-Wesley Iberoamericana, 586 pag., E.U.A, 1986.
- [17] Henry E., *Large-scale Industrial reuse to reduce cost and cycle time*. Matra Cap Systems. IEEE Software 1995, pp 47.
- [18] Jacobs J. Q., *Delphi Developer's Guide to OpenGL*, Wordware Publishing Inc, USA, 465 pag., 1999.
- [19] Jones G. W., *Software Engineering*, John Wiley & Sons, Inc., USA, 1990, 479 pag.

- [20] Kanaya I., *OpenGL for Delphi 5*, <http://www.chihara.aist-nara.ac.jp/people/STAFF/kanaya/index.html>, última fecha de acceso: 16 de Enero de 2002.
- [21] Kerr J. y Hunter R., *Inside RAD*, McGraw-Hill, 1994.
- [22] KnowledgeStorm, *CASE Tools*, <http://www.knowledgestorm.com/search/browse/156/156.jsp>, última fecha de acceso: 7 de Marzo de 2003.
- [23] Krueguer C. W., *Software reusable*, School of Computer Sciencie – Carnegie Mellon University, Pittsburg Pennsylvania, Octubre de 1991.
- [24] Marin M., *Fibras Ópticas Tecnología, Aplicaciones y Proyectos*, Jackson S.A, España, 1990.
- [25] Martín J., *Rapid Application Development*, Prentice-Hall, 1991
- [26] Nauer P. And Randell, B., Ene. 1968. *Software Engineering: Report on a conference by the NATO Science committec*, NATO Scientific Affairs División, Brussels, Belgium.
- [27] Nierstrasz O. and Schneider, J,G., *Components, scripts and glue*, Software composition group, University of Berne, Switzerland, August 1999.
- [28] Opengl, *OpenGL*, <http://www.opengl.org/> , última fecha 31 de Julio de 2001.
- [29] Orionis, *3D en Delphi*, <http://www.orionis.com.mx/tallerdelphi/articulos.htm>, última fecha de acceso: 10 de Diciembre de 2001.
- [30] Osier D. et. al., *Aprendiendo Delphi 3*, Prentice Hall, México, 599 pag., 1998.
- [31] Pressman R. S., *Ingeniería de Software: Un Enfoque Práctico*, Cuarta Edición, McGRaw-Hill, España, 1997.

- [32] Rumbaugh, J. et al. *Object-Oriented Modelling and Design*. Prentice Hall, Inc. Simon & Schuster Internacional Group, 1991., pp 372.
- [33] Shneider, J.G, *Components, Scripts, and Glue: A conceptual framework for software composition*, PhD Dissertation, University of Bern, Switzerland, Octubre de 1999.
- [34] Sld, *Experiencias de la aplicación de la ingeniería de software en sistemas de gestión*, <http://www.sld.cu/instituciones/cecam/memorias.html>, última fecha de acceso: 20 de Enero de 2003.
- [35] Szyperski C., “*Component Software: Beyond Object-Oriented Programming*” Addison-Wesley, 1999.
- [36] Somerville I., *Ingeniería de Software*, Pearson Educación, 712 pag., México, 2002.
- [37] Tempel Red/Com, *Tutoriales - Fibra Óptica*, http://www.tempel.es/tempelredcom/tutor/fibra_optica.html, última fecha de acceso: 11 de septiembre de 2002.
- [38] Tech-System Electronics, Inc., *Index*, <http://shop.store.yahoo.com/techsystems/index.html>, última fecha de acceso: 2 de Febrero de 2003.
- [39] Tippens P. E., *Física Conceptos y Aplicaciones*, McGraw-Hill, México, 934 pag., 1988.
- [40] Udlap, *Fibra óptica*, http://mailweb.udlap.mx/~lgojeda/telecom3/fibra_optica, última fecha de acceso: 7 de Marzo de 2002.
- [41] Udlap, *Conceptos de ingeniería de software -case y análisis estructurado-*, http://info.pue.udlap.mx/~tesis/lis/tecuapetla_1_fj/capítulo2.pdf, última fecha de acceso: 20 de Enero de 2003.
- [42] Vallecillo M. A., *Un modelo de componentes para el desarrollo de aplicaciones distribuidas*, Tesis Doctoral, Dpto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga España, Marzo de 1999.

[43] Yingxu W., and Graham K., *Software engineering proceses: Principles and applications*, CRC Press, U.S.A, 2000, 708 pag.

[44] Zanger H. and Zanger C., *Fiber Optics Comunication and Other Applications*, Macmillan Publishing Company, USA, 215 pag., 1991.

APÉNDICE A

PRINCIPIOS DE LAS ONDAS ELECTROMAGNÉTICAS

En las ondas de luz, como en todas las ondas electromagnéticas, existen campos eléctricos y magnéticos en cada punto del espacio, que fluctúan con rapidez [44]. Como estos campos tienen una magnitud y una dirección determinada, son cantidades vectoriales. Los campos eléctrico y magnético son perpendiculares entre sí, y también perpendiculares a la dirección de propagación de la onda. La onda luminosa más sencilla es una onda sinusoidal pura, llamada así porque una gráfica de la intensidad del campo eléctrico o magnético trazada en cualquier momento a lo largo de la dirección de propagación sería la gráfica de una función seno.

El número de oscilaciones o vibraciones por segundo en un punto de la onda luminosa se conoce como frecuencia. La longitud de onda es la distancia a lo largo de la dirección de propagación entre dos puntos con la misma 'fase', es decir, puntos que ocupan posiciones equivalentes en la onda. Por ejemplo, la longitud de onda es igual a la distancia que va de un máximo de la onda sinusoidal a otro, o de un mínimo a otro.

Como ejemplo se podría mencionar lo siguiente: Cuando a través de un alambre pasa corriente alterna, entonces se genera un campo electromagnético que se propaga a través del alambre. Ahora, como su nombre lo implica, el campo electromagnético consiste de un campo eléctrico y un campo magnético. Estos campos son perpendiculares entre sí, además de ser perpendiculares a la dirección de propagación.

El campo electromagnético, debido a que está determinado por una onda que es inducida por una corriente senoidal en un conductor, entonces el campo puede incrementar, decrementar e

invertir su polaridad con el tiempo, ocasionando que su intensidad varíe senoidalmente de igual forma con el tiempo.

La velocidad de propagación de las ondas electromagnéticas en el vacío es representada por c , donde

$$c = 300 \times 10^6 \text{ m/s} \quad (1)$$

Esta velocidad es independiente de la frecuencia. Ahora bien, la velocidad de propagación en cualquier otro medio diferente al vacío depende particularmente del material y siempre es menor que c [16,44].

La luz esta formada por un grupo de ondas electromagnéticas con diferentes frecuencias. Entonces cuando se hace referencia a la velocidad de propagación, realmente de lo que se trata es a una velocidad de grupo, es decir, la velocidad de un grupo de ondas [38].

LONGITUD DE ONDA (λ)

La longitud de onda ya ha sido definida anteriormente y es denotada con λ . En el espectro visible, las diferencias en longitud de onda se manifiestan como diferencias de color. El rango visible va desde 350 nanómetros (violeta) hasta 750 nanómetros (rojo), aproximadamente (un nanómetro, nm, es una milmillonésima de metro). La luz blanca es una mezcla de todas las longitudes de onda visibles. No existen límites definidos entre las diferentes longitudes de onda, pero puede considerarse que la radiación ultravioleta va desde los 350 nm hasta los 10 nm. Los rayos infrarrojos, que incluyen la energía calorífica radiante, abarcan las longitudes de onda situadas aproximadamente entre 750 nm y 1 mm [44].

La velocidad de una onda electromagnética es el producto de su frecuencia y su longitud de onda. En el vacío, la velocidad es la misma para todas las longitudes de onda. La velocidad de la luz en las sustancias materiales es menor que en el vacío y varía para las distintas longitudes de onda.

Como la velocidad de la onda en el espacio libre es c , entonces la distancia que viaja en un tiempo T es

$$\lambda = T \times c \quad \text{ó} \quad \lambda = c/f, \quad (T=1/f) \quad (2)$$

La velocidad de fase denotada por V_p es igual a c en el espacio vacío, y en otros materiales tal velocidad esta dada por

$$V_p = \lambda \times f \quad (3)$$

Las definiciones de λ y V_p son aplicables para ondas monocromáticas, lo cual significa que se tratan de ondas de una sola frecuencia o un solo color.

ÍNDICE DE REFRACCIÓN

La relación entre la velocidad de la luz en el vacío y la velocidad de una sustancia se conoce como índice de refracción de la sustancia para la longitud de onda. El índice de refracción del aire es 1,00029 y apenas varía muy poco con la longitud de onda. En la mayoría de las aplicaciones resulta suficientemente preciso considerar que es igual a 1.

$$n = c/V_p \quad (4)$$

Para el vidrio que es el material con el cual se fabrica la fibra óptica tiene un índice de refracción que varía entre los siguientes valores [44]:

Material	Índice de refracción (n)	Velocidad de la luz ($V_p \times 10^6$ m/s)
Vacío	1	300
Aire	1.0003	299.9
Hielo	1.3	230.77
Agua	1.33	255.56
Alcohol	1.36	220.59
Vidrio	1.46-1.96	205.48 – 151
Cuarzo	1.54	194.8
Diamante	2.42	123.97

Tabla 1. Índices de refracción típicos y velocidades de la luz.

Es importante notar que la frecuencia de una onda no cambia, lo cual indica que λ_1 en el material con índice de refracción n_1 esta relacionado con λ en el vacío por:

$$\lambda_1 = \lambda / n_1 \quad (5)$$

FIBRAS ÓPTICAS

La fibra con la estructura más sencilla y más manejable, es la fibra multimodo con índice escalonado. Otras dos estructuras son la fibra con índice degradado o variable, y la fibra monomodo con índice escalonado. Estas tres estructuras están ilustradas en la figura 1, así como también se muestra el perfil del índice de refracción y algunas dimensiones típicas [7,44].

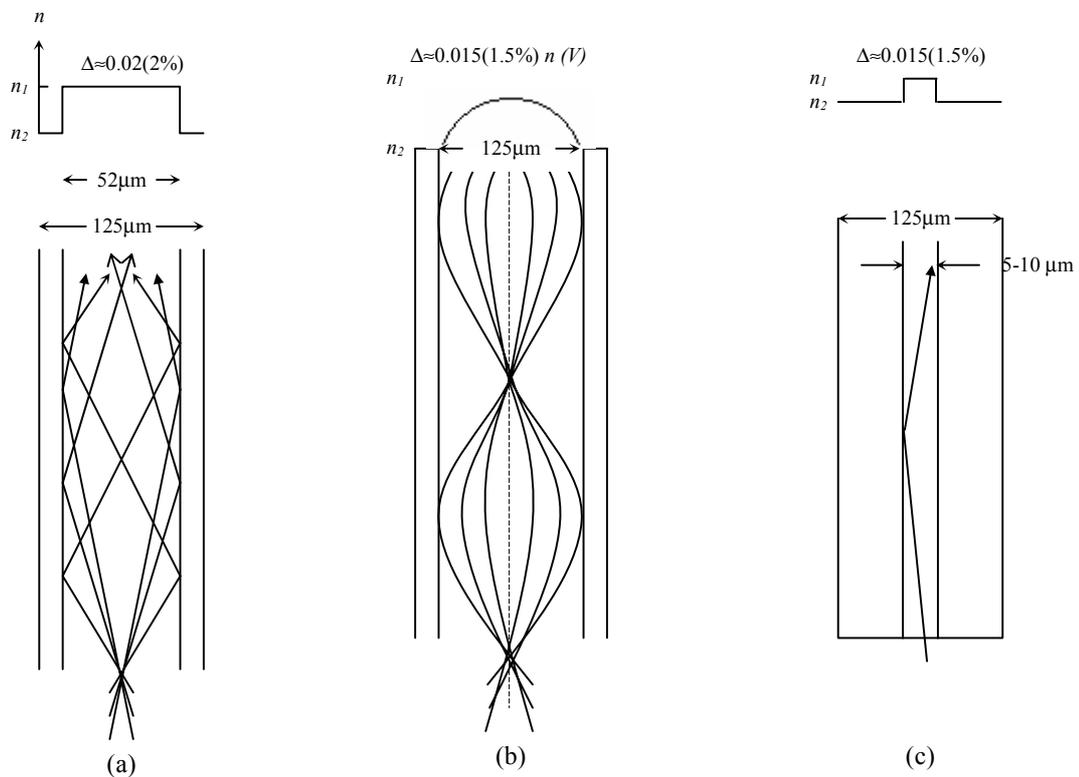


Figura 1. Diferente tipos de fibra (a) multimodo con índice escalonado (b) índice degradado (c) monomodo con índice escalonado.

Para ver el funcionamiento completo en términos de la razón de información y atenuación, la fibra monomodo tiene las mejores características (alta razón entre información y una mínima atenuación) mientras la fibra multimodo tiene la peor. Esto no significa que la fibra monomodo tendría que ser usada en la mayoría de los casos ya que esto representaría un mayor costo y una mayor dificultad para trabajar con su pequeño tamaño.

FIBRAS MULTIMODO CON ÍNDICE ESCALONADO

Al observar en la figura 1(a) y 1(c) probablemente parecerá que no existe una diferencia significativa entre las dos fibras de índices escalonados a menos que sea el tamaño. Claro, esta es la única diferencia fundamental. Las consecuencias de esta diferencia de tamaño son substanciales.

En la fibra multimodo con índice escalonado ilustrada en la figura 1(a), la luz se propaga en muchos modos. El número total de modos M_N se incrementa conforme incrementa la apertura numérica (N.A). Para un gran número de modos, M_N puede ser aproximado por

$$M_N = V^2/2 \quad (6)$$

La frecuencia normalizada V es una relación entre el tamaño de la fibra, los índices de refracción, y la longitud de onda. V es la frecuencia normalizada y esta dada por

$$V = [(2\pi\alpha)/\lambda] \times \text{N.A.} \quad (7)$$

$$V = [(2\pi\alpha)/\lambda] \times n_1 \times (2\Delta)^{1/2} \quad (8)$$

Donde α es el radio del núcleo, λ es la longitud de onda, n_1 índice de refracción del núcleo y Δ la diferencia relativa de los índices de refracción.

Una gran apertura numérica N.A significa mas modos, y más modos también significan una alta dispersión. Así mismo, un gran diámetro significa más modos y una alta dispersión. Es útil saber que una alta dispersión significa una tasa de transferencia de información muy baja, lo cual conduce a hacer menos eficiente la transmisión.

El radio del núcleo de una fibra de comunicaciones multimodo varía entre los 50 μm y los 200 μm , con la cubierta un poco gruesa, típicamente igual al radio del núcleo.

En la tabla 2 se muestran algunos datos relacionados para una fibra multimodo de índice escalonado. La fibra multimodo con índice escalonado es relativamente fácil de manufacturar, además de ser menos costosa que los otros dos tipos de fibra (de índice degradado y de monomodo) [44].

Material Núcleo/ Cubierta	Núcleo Diámetro (μm)	Cubierta Diámetro (μm)	Pérdida (dB/km @850 nm)	BW (MHz x km)	N.A.
Silicio/Silicio	50	125	2.4	800	0.2
Silicio/Silicio	100	140	4.5	300	0.29
Silicio/Silicio	200	300	8	20	0.2
Silicio/Silicio	1000	1250	6	-	0.4
Silicio/Plastico	250	320	8	20	0.3
Silicio/Plastico	200	600	30(@ 820nm)	90	0.38
Silicio/Plastico	368	400	270(@ 790nm)	50	0.42

Tabla 2. Típicas fibras multimodo con índice escalonado.

FIBRAS MULTIMODO CON ÍNDICES DEGRADADOS

Una fibra con índice variable es mostrada en la figura 2. Aquí, el índice de refracción del núcleo varía conforme se vaya acercando a su centro. En el centro de la fibra se tiene n_1 , en la cubierta se tiene a n_2 y en medio se tiene $n(r)$, donde n es una función de un radio en particular (esto se puede ver en la figura 2(a)). En la figura 2(b) se simula el cambio del índice de refracción de una forma ingeniosa. Cada círculo concéntrico representa un índice de refracción diferente, decrementándose conforme se va acercando al centro de la fibra. Un rayo incidente en las fronteras entre $n_A - n_B$, $n_B - n_C$, etc., es refractado. Eventualmente, en el índice n_2 el rayo es lanzado en dirección opuesta y reflejado en su totalidad. Esta refracción continua produciendo el rayo trazado en la figura 2(c).

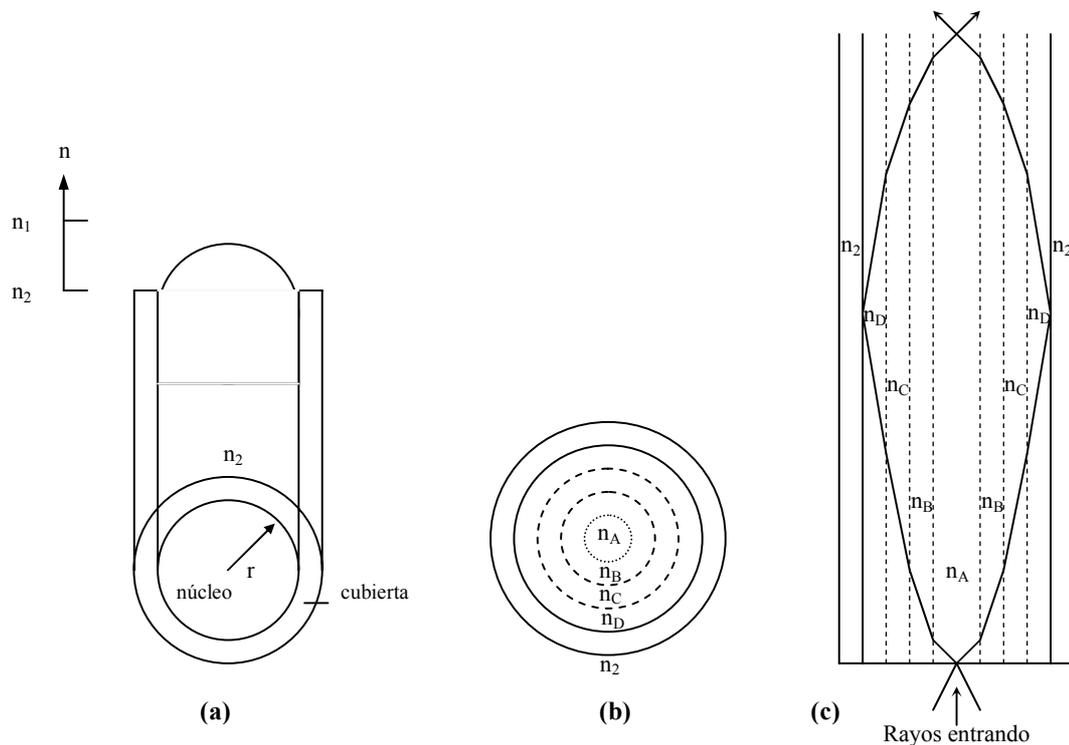


Figura 2. Fibra con índice degradado. (a) perfil del índice (b) perfil en capas del índice (c) trayectoria del rayo a través del perfil en capas del índice.

La descripción matemática de $n(r)$ es dada por:

$$n(r) = n_1 \times \{ 1 - [2 \times \Delta \times (r/\alpha)^2] \}^{1/2} \quad (9)$$

Donde α es el radio total del núcleo Δ depende únicamente de los índices de refracción en el centro del núcleo (n_1) y el borde exterior [44].

La apertura numérica N.A. de una fibra con índice degradado varía radialmente junto con r , puesto que n continuamente esta cambiando. Sin embargo, la apertura numérica N.A. de una fibra con índice degradado se calculará exactamente como para una fibra de índice escalonado $N.A. = (n_1^2 - n_2^2)^{1/2}$.

El número de modos en una fibra con índice degradado, es aproximadamente la mitad del número de modos de una fibra con índice escalonado.

$$M_N = V^2/4 \quad (10)$$

El bajo número de modos en una fibra con índice degradado produce una baja dispersión como la que es encontrada en la fibra de índice escalonado.

Para la fibra con índice degradado la dispersión esta dada por:

$$\Delta t = (L \times n_1 \times \Delta^2) / (8 \times c) \quad (11)$$

El tamaño de la fibra de índice degradado, es casi el mismo para la fibra con índice escalonado. Sin embargo su fabricación es más complicada.

El ángulo de aceptación efectivo de la fibra de índice degradado, es algo menor que el ángulo de aceptación de la fibra con índice escalonado. Esto hace que el acoplamiento de la fibra con la fuente de luz sea más difícil.

ÍNDICE ESCALONADO: FIBRAS DE MODO SIMPLE Ó MONOMODO

Como su nombre lo indica, la energía luminosa en una fibra de modo simple esta concentrada en un solo modo. Esto es llevado a cabo por la reducción de Δ y/o el diámetro del núcleo de tal forma que la fibra es diseñada para tener una frecuencia normalizada V entre 0 y 2.4. Este pequeño valor relativo significa que el radio de la fibra, y Δ la diferencia relativa de los índices de refracción deben ser pequeñas.

El valor común de Δ para una fibra multimodo es aproximadamente 2%, y la N.A. es aproximadamente 0.2. Para una fibra de modo simple tanto Δ como N.A son muy pequeñas. Una apertura numérica pequeña significa un ángulo de aceptación pequeño. Dando por resultado que el rayo incidente debe ser casi perpendicular al borde de la fibra.

En las fibras de monomodo a diferencia de las fibras multimodo, parte de la luz se propaga en la cubierta, entonces ésta debe tener una pérdida baja y ser relativamente más gruesa. Típicamente, el diámetro del núcleo es de 10 μm , y el de la cubierta es aproximadamente 120 μm . El tamaño completo de la fibra es aproximadamente el mismo que las otras fibras. El manejo y la manufacturación de la fibra de modo simple es mas difícil. Además este tipo de fibra es la que tiene una pérdida muy baja y una alta razón de transferencia de información, además de ser la más costosa. Sin embargo, su costo es considerado efectivo, en los casos en los que dicho costo esta relacionado con la razón de transferencia de información.

OTRAS FIBRAS

En la actualidad han surgido nuevas creaciones de la fibra óptica, ya sea modificando el material de su cubierta, o variando sus índices de refracción. Estos diversos tipos de fibras están siendo ocupados en aplicaciones especiales como son sistemas para iluminación, sistemas experimentales, para sistemas de circuito cerrado y demás. Algunos de los nuevos tipos de fibra que existen en la actualidad son los siguientes: Fibras en paquetes, fibras monomodo con índices triangular y parabólico, fibras W, fibras de plástico, por mencionar entre otras [7,44]. Todas estas nuevas fibras tienen características similares a las fibras monomodo y multimodo con índice escalonado, así como también a las fibras con índice degradado, solo que algunas presentan características mejoradas con la finalidad de reducir la pérdida y la dispersión que presentan las fibras comunes al momento de la transmisión de la luz.

APÉNDICE B

CÓDIGO FUENTE

FUNCIONES DE LA SIMULACIÓN DE LA LEY DE REFRACCIÓN

Función para el cálculo de ángulo de refracción

```
anguloTemp:=strtofloat(edit1.text); // Almacena ángulo de incidencia
anguloIncidente:= 90-strtofloat(edit1.text);
n1:= strtofloat(edit3.text); // Almacena índices de refracción
n2:= strtofloat(edit4.text);
anguloCritico:=arcSin((n2/n1))*180/pi; // Calcula ángulo crítico
if anguloTemp > anguloCritico then // Decisión de Caso especial de la reflexión total interna
begin
  StatusBar1.SimpleText:= 'Caso especial del fenómeno de Reflexión Total Interna !!!';
  MnOgltimer1.Enabled:=False;
end
else
begin //Realiza cálculo de el ángulo de refracción.
  anguloRefraccion:=90-arcSin((n1/n2)*sin(anguloTemp*pi/180))*180/pi;
  str((90-anguloRefraccion):5:3,S); // Coloca ángulo de refracción en una cadena
  label10.caption:=S;
  MnOgltimer1.Enabled:=True; // Ejecuta la simulación
  x:=-3; // x,y,z coordenadas de la línea que representa la trayectoria del rayo
  y:=0;
  z:=-x*sin(anguloIncidente*pi/180)/cos(anguloIncidente*pi/180);
  z1:=-x*sin(anguloRefraccion*pi/180)/cos(anguloRefraccion*pi/180);
  Trayectoria.Visible:=true; // Dibuja la trayectoria
end;
```

Función de simulación de la ley de refracción

```
procedure TFormRefracc.MnOglTimer1timer(Sender: TObject);
VAR
  xyz : TXYZ; // Variable para manipular la posición de la esfera
begin
  MNOglCanvas1.Changing := TRUE;
  xyz := MNOglScene2.Translation; // Ejecuta la animación de la esfera que
  // recorre la trayectoria trazada por el rayo de luz
  xyz.x := xyz.x + 1; // Asigna valores a las coordenadas de la esfera
  xyz.z := -xyz.x*sin(anguloIncidente*pi/180)/cos(anguloIncidente*pi/180);
  xyz.y := 0;
  If xyz.x > 0 then // Inicia recorrido
  begin
    xyz.x := xyz.x + 1;
    xyz.z := -xyz.x*sin(anguloRefraccion*pi/180)/cos(anguloRefraccion*pi/180);
    // Regresa de donde al punto inicial...
    if xyz.x > 4 then
    begin
      xyz.x:= -4;
      xyz.z := -xyz.x*sin(anguloIncidente*pi/180)/cos(anguloIncidente*pi/180);
    end;
  end;
  MNOglScene2.Translation := xyz;
  MNOglCanvas1.Changing := FALSE;

end;
```

FUNCIONES DE LA SIMULACIÓN DE LA LEY DE REFLEXIÓN TOTAL INTERNA

Función de cálculo de ángulo crítico

```
n1:=strtofloat(edit3.text); // Almacena los índices de refracción
n2:=strtofloat(edit4.text);
if n1>n2 then // Para que existe la reflexión total interna n1 > n2
begin
  anguloCritico:=arcSin(n2/n1)*180/pi; // Realiza cálculo de ángulo crítico
  str(anguloCritico:5:3,S);
  edit5.text:= S; // Muestra el ángulo crítico en una cadena
end
```

Función de cálculo de ángulo de refracción ó reflexión

```

anguloIncidencia:=strtofloat(edit1.text); // Almacena ángulo de incidencia
x:=-3.5; // x,y,z coordenadas de la línea que representa la trayectoria del rayo
y:=0;
z:=-x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);

if anguloIncidencia < anguloCritico then // Ocurre refracción
begin
  StatusBar1.SimpleText:='El rayo es REFRACTADO debido a que el ángulo de
  incidencia es menor al ángulo crítico calculado !!';
  anguloRefraccion:=(arcSin((n1/n2)*sin(anguloIncidencia*pi/180))*180/pi);
  str(anguloRefraccion:5:3,S);
  edit2.text:= S;
  refractado:=1;
  z1:=-x*sin((90-anguloRefraccion)*pi/180)/cos((90-anguloRefraccion)*pi/180);
end;

if anguloIncidencia>=anguloCritico then // Ocurre el fenómeno de reflexión total interna
begin
  StatusBar1.SimpleText:='El rayo es REFLEJADO debido a que el ángulo de incidencia
  es mayor al ángulo crítico calculado !!';
  anguloRefraccion:=anguloIncidencia;
  str((90-anguloIncidencia)+90):5:3,S);
  edit2.text:=S;//floattostr((90-anguloIncidencia)+90);
  z1:=x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);
  refractado:=0;
end;
Trayectoria.Visible:=true; // Dibuja la trayectoria
MnOglTimer1.Enabled:=True // Ejecuta la simulación

```

Función de simulación de la ley de refracción

```

procedure TFormRTotal.MnOglTimer1timer(Sender: TObject);
VAR
  xyz : TXYZ;
begin
if (refractado=0) then // Rayo reflejado
begin
  MNOglCanvas1.Changing := TRUE;
  xyz := MNOglScene2.Translation;
  xyz.x := xyz.x + 1;
  xyz.z := -xyz.x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);
  xyz.y := 0;
  If xyz.x > 0 then
  begin
    xyz.x := xyz.x + 1;
    xyz.z := xyz.x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);
    // Regresa al punto inicial
    if xyz.x > 4 then
    begin
      xyz.x:= -4;
      xyz.z := -xyz.x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);
    end;
  end;
end;

```

```

    end;
end;
MNOglScene2.Translation := xyz;
MNOglCanvas1.Changing := FALSE;
end; //del if refractado=0

if (refractado=1) then // Rayo refractado
begin
MNOglCanvas1.Changing := TRUE;
xyz := MNOglScene2.Translation;
xyz.x := xyz.x + 1;
xyz.z := -xyz.x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);
xyz.y := 0;
If xyz.x > 0 then
begin
xyz.x := xyz.x + 1;
xyz.z := -xyz.x*sin((90-anguloRefraccion)*pi/180)/cos((90-anguloRefraccion)*pi/180);
//****regresa de donde partio...
if xyz.x > 4 then
begin
xyz.x:=- 4;
xyz.z := -xyz.x*sin((90-anguloIncidencia)*pi/180)/cos((90-anguloIncidencia)*pi/180);
end;
end;
MNOglScene2.Translation := xyz;
MNOglCanvas1.Changing := FALSE;
end; // Fin del rayo refractado

end;

```

FUNCIONES DE LA SIMULACIÓN DE LA FIBRA ÓPTICA

Función de cálculo para algunos ángulos, apertura numérica, y el radio del cono de aceptación

```

n1x:=strtofloat(edit1.text); // Índices de refracción
n2x:=strtofloat(edit2.text);

// Calculamos ángulos y apertura numérica...
AngCP:=arcCos(n2/n1)*180/pi; // Ángulo crítico de propagación
AngC :=arcSin(n2/n1)*180/pi; // Ángulo crítico
AperN:=sqrt(n1*n1 - n2*n2); // Apertura numérica
AngA :=2*(arcSin(sqrt(n1*n1 - n2*n2))*180/pi); // Ángulo de aceptación

// Calculamos el radio del cono de aceptación...
R:=4*tan((AngA/2)*180/pi);
// Ahora con dicho radio se crea el cono...
Formfo.ConoAceptancia.Visible:=true;
Formfo.ConoAceptancia.TopRadius:=R;

```

Función de determinación si el rayo entra ó sale de la fibra.

```

AngInci:=strtofloat(Edit1.text); // Almacena ángulo de incidencia
angTeta:=arcSin((1/n1)*Sin(AngInci*pi/180))*180/pi;
str(angTeta:5:3,A1);
label28.Caption:=A1;
angAlfa:=90-angTeta;
str(angAlfa:5:3,A3);
label30.Caption:=A3;
if angAlfa > AngC then // El rayo continua dentro de la Fibra Óptica
begin
    // Como anguloAlfaX=angAlfa > anguloAlfaOriginal -> Existe reflexión total interna
    FormFO.StatusBar1.SimpleText:='El rayo se propaga dentro de fibra óptica...';

    // Calculamos el ángulo con el cual va ha ser reflejado el rayo dentro de la fibra
    angY:=90-angAlfa;
    banderaEntrar:=1;

end

else // El rayo abandona de la fibra óptica
begin
    FormFO.StatusBar1.SimpleText:='El rayo SALE de la fibra óptica...';
    // Calculamos el ángulo al cual va ha salir de la fibra.. primera parte
    angX:=90-(arcSin((n1/n2)*sin(angAlfa*pi/180))*180/pi);
    banderaEntrar:=0;
end;

```

Función de simulación de la fibra óptica

```

procedure TFormFO.MnOglTimer4timer(Sender: TObject);
VAR
    xyz : TXYZ;
    ztemp:Real;
begin
    MNOglCanvas1.Changing := TRUE;
    xyz := MNOglSphere1.Translation;

    if i=1 then
    begin
        xyz.x:=xyzTemp.x;
        xyz.y:=2.0;
        xyz.z:=xyzTemp.z;
        i:=0;
        ztemp:=0;
    end;

    if banderaEntrar=1 then // El rayo se refleja internamente con un angulo angY
    begin
        xyz.x:= 0;
        xyz.z := xyz.z - 0.5;
        ztemp:=ztemp+0.5;
        xyz.y := xyz.y - ztemp*sin(angY*pi/180)/cos(angY*pi/180);
    end;

```

```

    if xyz.y<-2.0 then
    begin
        MnOgltimer4.Enabled:=false; // Regresamos al punto inicial es decir se repite la
animación
        xyz.z :=5; //para regresar al inicio...
        xyz.y := -xyz.z*sin(anglnci*pi/180)/cos(anglnci*pi/180);
        MnOgltimer3.Enabled:=true;
    end;
end
else // El rayo sale de la fibra...con un ángulo angx
begin
    xyz.x:= 0;
    xyz.z := xyz.z -0.5;
    ztemp:=ztemp +0.5;
    xyz.y := xyz.y + ztemp*sin(angX*pi/180)/cos(angX*pi/180);
    if xyz.y>4 then
    begin
        MnOgltimer4.Enabled:=false; // Regresamos al punto inicial es decir se repite la
animación
        xyz.z :=5; // Coordenadas iniciales...
        xyz.y := -xyz.z*sin(anglnci*pi/180)/cos(anglnci*pi/180);
        xyz.x := 0;
        MnOgltimer3.Enabled:=true;
    end;
end;
MNOglSphere1.Translation := xyz;
MNOglCanvas1.Changing := FALSE;

end;

```

FUNCIONES PARA LA OBTENCIÓN DE CÁLCULOS TÉCNICOS

Función de cálculo de la apertura numérica

```

n1:=strtofloat(edit1.text); // Índices de refracción
n2:=strtofloat(edit2.text);
// Calculamos la apertura numérica
NA:=sqrt(sqr(n1)-sqr(n2));

```

Funciones de selección de longitud de onda

```

procedure TFormTerceraParte.N820880nm1Click(Sender: TObject);
begin
    perdidaLongitudOnda:=2.2;
    // Longitud de onda promedio de 820-880 nm...>850nm
    LongitudOnda:=0.00000085;
end;

procedure TFormTerceraParte.N12001320nm1Click(Sender: TObject);

```

```

begin
    perdidaLongitudOnda:=0.6;
    // Longitud de onda promedio de 1200-1320 nm...>1260nm
    LongitudOnda:=0.00000126;

end;

procedure TFormTerceraParte.N15501610nm1Click(Sender: TObject);
begin
    perdidaLongitudOnda:=0.2;
    // Longitud de onda promedio de 1550-1610 nm...>1580nm
    LongitudOnda:=0.00000158;
end;

```

Función de cálculo del número de modos

```

radioNucleo:=strtofloat(edit1.text); // Almacena radio del núcleo
// Una vez teniendo N.A, y longitud de onda y el radio del núcleo
// Calculamos la frecuencia normalizada...V
V:=((2*pi*radioNucleo*0.000001)/longitudOnda)*NA;
// Y por ultimo calculamos el numero de modos...MN
MN:=trunc(sqr(V)/2);

```

Función de cálculo de la dispersión intermodal

```

longitud:=strtofloat(formY.edit1.text); // Almacena la longitud de la fibra óptica
// Ahora teniendo: N.A, longitud de onda, el radio del núcleo...y longitud de la FO
// Calculamos...la Dispersión Intermodal
dispersionIntermodal:=((longitud*1000*sqr(NA))/(2*n1*300000000))*1000000000;

```

Función de cálculo de la potencia de salida

```

perdidaTotal:=numConectores*perdidaxConector + numEmpalmes*perdidaxEmpalme +
    longitud*perdidaLongitudOnda;
perdidaTotal:=perdidaTotal*-1;
logInverso:=power(10,perdidaTotal);
Pout:=Pin*logInverso;
formPout.Edit1.Text:=floattostr(Pout);

```

APÉNDICE C

MANUAL DE USUARIO

SOFTWARE DE APOYO PARA EL ESTUDIO DE LOS EFECTOS ÓPTICOS
EN UNA FIBRA

ÍNDICE

Introducción	106
Requerimientos de Hardware y Software.....	107
Acerca de la instalación.....	107
Guía del usuario.....	108
a) Pantalla Principal.....	108
b) Sección de Simulación de Leyes Elementales	109
c) Sección de Simulación de la Fibra Óptica	115
d) Sección de Cálculos Técnicos	119

I. Introducción

El software que se describe en este manual, es un sistema que permite llevar a cabo el estudio de los efectos ópticos en una fibra. Abarcando, la simulación de las leyes elementales de la propagación de la luz (ley de reflexión, ley de refracción y reflexión total interna), la simulación del confinamiento de la luz dentro de la fibra óptica, y la posibilidad de realizar algunos cálculos técnicos, útiles en la construcción de sistemas de comunicación por este medio.

Así pues, este software se divide en tres secciones:

- Sección de simulación de leyes elementales
- Sección de simulación de la fibra óptica
- Sección de cálculos técnicos

Las dos primeras secciones tienen como principal objetivo la simulación del fenómeno en cuestión, en forma gráfica. Esto, representado en escenarios tridimensionales con gráficos animados. Mientras que la última de las secciones brinda únicamente la oportunidad de realizar cierto tipo de cálculos, cuyos resultados son valores numéricos de tipo técnico.

II. Requerimientos de Hardware y Software

Para que este software funcione correctamente, es necesario que cuente con los siguientes requerimientos:

Hardware:

- Una PC convencional
- Procesador Pentium o Superior
- Un monitor VGA o Superior
- Memoria RAM 64 Mb o Superior
- Espacio mínimo libre en disco duro de 10 Mb

Software:

- Sistema operativo Windows Millennium, Windows 2000 ó Windows XP.

III. Acerca de la Instalación y Ejecución

Para instalar el software “Sophy I” es necesario seguir los pasos descritos a continuación:

- 1) Insertar CD del software
- 2) Mostrar el contenido del CD
- 3) Ejecutar la aplicación <<SETUP>>
- 4) Seguir al asistente para elegir el destino de la instalación

Una vez instalado el software, aparecerá un icono del simulador en el escritorio de Windows, así como en la sección de programas del botón inicio.

Para ejecutar el software puede hacerlo con un doble clic con el botón izquierdo del ratón o simplemente posicionando el puntero del ratón sobre el icono del software “Sophy I” y oprimir la tecla ENTER.

III. Guía de Usuario

a) Pantalla Principal

Después de haber instalado el Software “Sophy I” y ejecutarlo lo primero que aparecerá en su monitor es la pantalla principal de bienvenida (figura 1).



Figura 1. Pantalla de bienvenida.

Para poder continuar y presentar el menú principal del software “Sophy I”, lo único que se debe de hacer es dar un click con el botón izquierdo en cualquier parte de la pantalla de bienvenida. Después de esto, el software mostrará la Pantalla del Menú principal, la cual es mostrada en la figura 2.



Figura 2. Menú principal..

El menú principal está compuesto por tres grupos de imágenes, las cuales representan la ejecución de cada una de las secciones que conforman el sistema. Además, de incluir en la parte inferior derecha de la pantalla, una etiqueta llamada "Salir", con la que se puede terminar la ejecución del software "Sophy I".

Para la ejecución de alguna de las secciones, lo que se debe hacer es, colocar el puntero del ratón sobre algún grupo de imágenes representativas de la sección, y realizar un solo click sobre ésta.

b) Sección de Simulación de Leyes Elementales

Cuando el usuario ejecute la primera sección del menú principal del software, se presentará la siguiente pantalla (Figura 3). En ésta se muestra una interfaz muy similar a la del menú principal, consistiendo básicamente en representar cada una de las simulaciones de las leyes elementales, en forma de iconos con imágenes representativas afines a la ley.

En esta pantalla se presenta en forma de iconos la simulación de la ley de reflexión, de refracción y, del fenómeno de la reflexión total interna. Para entrar a realizar alguna simulación, basta solo posicionar el puntero del ratón sobre la imagen representativa de la ley, y dar un clic sobre esta con el botón izquierdo. En esta pantalla se muestra una etiqueta con la leyenda "REGRESAR", con la cual al hacer un clic con el botón izquierdo del ratón, nos permite regresar al menú principal.

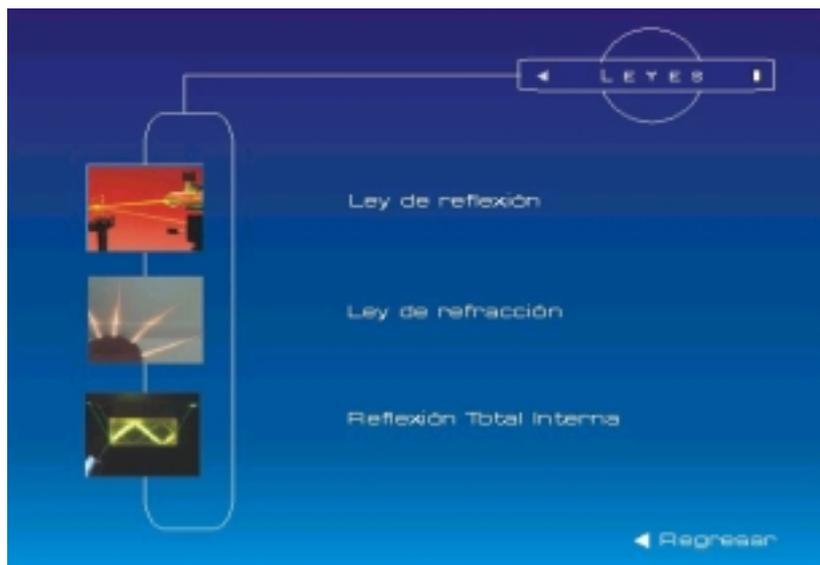


Figura 3. Menú para la simulación de alguna ley elemental.

SIMULACIÓN LEY DE REFLEXIÓN

La interfaz implementada para la simulación de la ley de reflexión, se muestra en la figura 4.

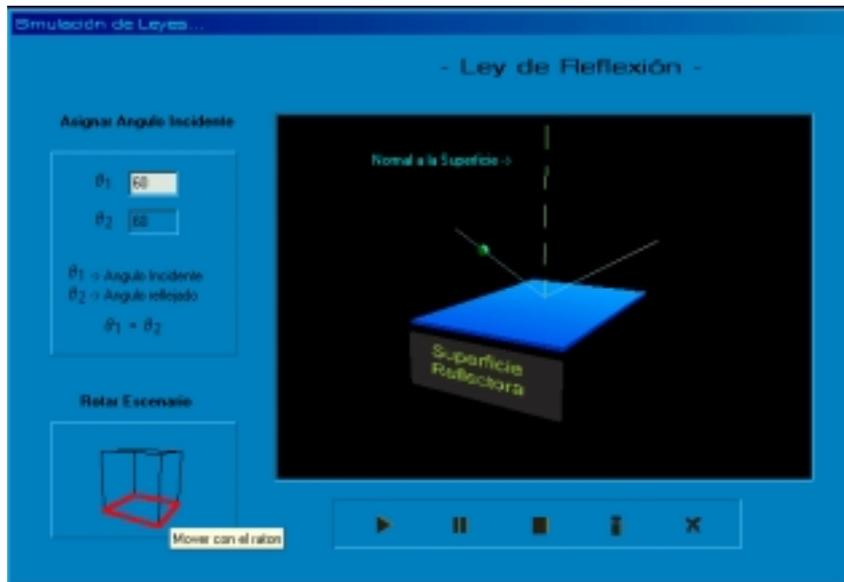


Figura 4. Simulación de la ley de reflexión.

En esta, como en las interfaces de las otras simulaciones, existen partes muy comunes entre ellas. En la simulación de cada una de las leyes elementales, existe una parte de la pantalla, en la cual se introducen los parámetros necesarios para llevar a cabo dicha simulación, así como también, presenta algunos valores numéricos como resultados. Para la simulación de la ley de reflexión tenemos esta parte mostrada en la figura 5.

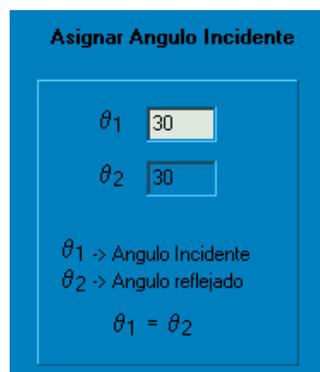


Figura 5. División de la pantalla para asignar parámetros, y mostrar resultados.

Como se puede observar para la simulación de la ley de reflexión, únicamente se necesita introducir el ángulo de incidencia θ_1 , donde el valor para este ángulo debe de estar en el rango de 0 a 90 grados. Se debe tomar en cuenta, que el ángulo introducido toma como referencia la normal a la superficie.

Si el valor asignado al ángulo de incidencia es incorrecto, el software enviará un mensaje de error como se muestra en la figura 6.



Figura 6. Mensaje de error producido por la validación del ángulo de incidencia.

Si el ángulo es correcto, se podrá ejecutar la simulación ocupando uno de los botones que forman parte del juego de componentes para controlar la simulación, los cuales son mostrados en la figura 7.

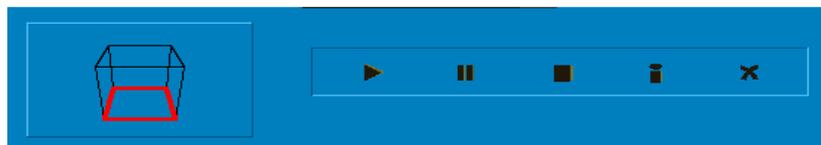


Figura 7. Juego de componentes para controlar la simulación.

Este juego esta formado por los siguientes componentes, nombrados de izquierda a derecha:

- Componente para rotar el escenario donde se muestra la simulación gráficamente
- Botones para:
 - Ejecutar la simulación
 - Pausar la animación de la simulación
 - Detener la simulación actual y preparar una nueva simulación
 - Mostrar información acerca de la ley que se esta analizando
 - Cancelar o Terminar la Simulación actual.

En caso, de que se oprima el botón de mostrar información en alguna de las simulaciones, el software presentará una ventana, como la que observa en la figura 8.

REFLEXIÓN

Ley de reflexión

Cuando un rayo de luz incide sobre una superficie reflectora, el rayo rebota hacia atrás como una pelota lanzada a una pared.

La ley de reflexión establece que el ángulo de incidencia es igual a el ángulo de reflexión. En la figura mostrada el rayo de incidencia es la línea AO, el rayo reflejado es OB. Los ángulos θ_1 y θ_2 son el ángulo de incidencia y el ángulo de reflexión respectivamente.

Aceptar

Figura 8. Información referente a la ley de reflexión.

SIMULACIÓN LEY DE REFRACCIÓN

La interfaz implementada para la simulación de la ley de refracción es muy similar a la simulación anterior, y ésta es mostrada en la figura 9.

Simulación de Leyes...

- Ley de Refracción -

Asignar Índices y Áng. Incidente

n_1 n_2

θ_1 θ_2

θ_1 → Ángulo Incidente
 θ_2 → Ángulo refractado

Rotar Escenario

Normal a la superficie →

Superficie de reflexión →

n_1

n_2

▶ || ■ ⓘ ✕

Figura 9. Simulación de la ley de refracción.

Para la simulación de esta ley, se necesitan introducir los índices de refracción n_1 y n_2 , de los medios por donde viaja el rayo de luz, y el ángulo de incidencia θ_1 . Si los valores numéricos asignados son correctos, se puede ejecutar la simulación oprimiendo el botón *Ejecutar*, del grupo de componentes para controlar la simulación.

Ahora bien, puede existir un caso, en donde, dado los índices de refracción y el ángulo de incidencia, no se permita ejecutar la simulación, mostrando un mensaje en la barra de estado como el que se muestra en la figura 10.

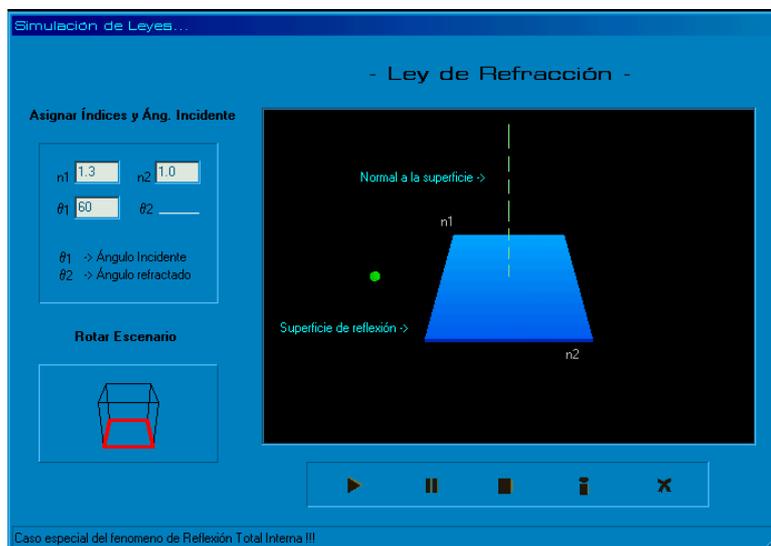


Figura 10. Ejemplo de cuando la ley de refracción no se puede simular.

Esto sucede, debido a que el software calcula automáticamente un ángulo llamado crítico, con los índices de refracción dados. Y si el ángulo de incidencia que se introduzca es mayor al ángulo crítico, no se puede ejecutar la simulación, pues este es un caso de la reflexión total interna, lo que es analizado en la siguiente simulación.

SIMULACIÓN DEL FENÓMENO DE REFLEXIÓN TOTAL INTERNA

La interfaz implementada para la simulación del fenómeno de ley de reflexión total interna es muy similar a las simulaciones anteriores, y ésta es mostrada en la figura 11.

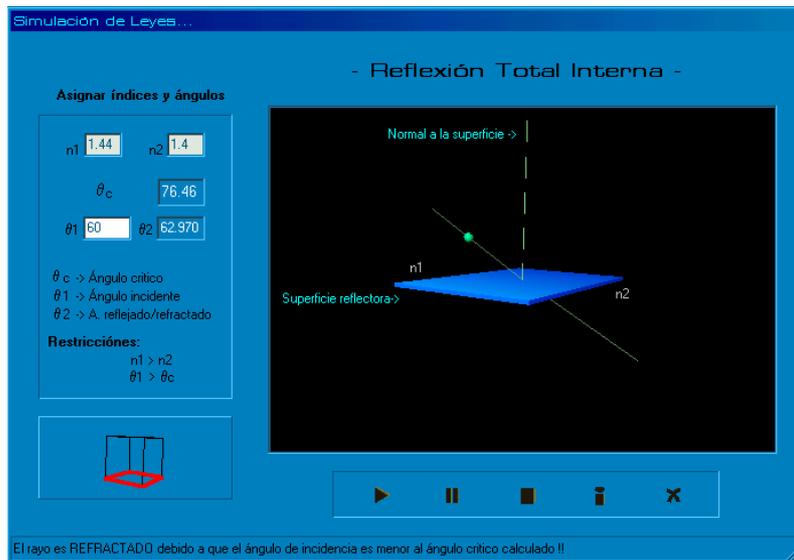


Figura 11. Interfaz de la simulación del fenómeno de reflexión total interna.

Para poder llevar a cabo esta simulación, se requiere que el usuario introduzca los índices de refracción n_1 y n_2 , en donde se debe tomar en cuenta la restricción que el índice de refracción n_1 , debe ser mayor a n_2 . A continuación tendrá que calcular el ángulo crítico, éste último se calcula oprimiendo el botón que se encuentra en la parte de introducir parámetros, etiquetado con el nombre θ_c .

El ángulo crítico, es un valor fundamental para la simulación de este fenómeno, ya que dependiendo de este ángulo se debe de asignar un valor al ángulo de incidencia. Y esto consiste en que, si el ángulo de incidencia es menor al ángulo crítico, el rayo se refractara y ocurrirá únicamente una refracción como se muestra en la figura 9. Pero si el ángulo de incidencia es mayor al ángulo crítico, el rayo se reflejara ocurriendo el fenómeno de reflexión total interna, esto último es mostrado en la figura 12.

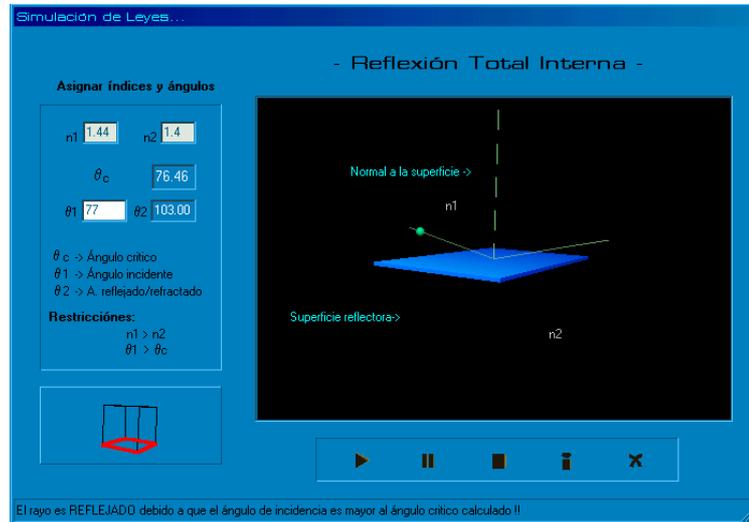


Figura 12. Interfaz de la simulación del fenómeno de reflexión total interna.

c) Sección de Simulación de la Fibra Óptica

La interfaz utilizada para la simulación de la fibra óptica, es la que se muestra en la figura 13. Como se puede observar tiene partes muy similares a las simulaciones anteriores. Lo único que varía en ésta, es un menú de opciones incluido en la parte superior de la pantalla. Este menú tiene ciertas funciones específicas para la simulación de esta sección.

El menú tiene las siguientes opciones:

- Información. En ésta, se encuentran algunas definiciones de conceptos claves que se utilizan en esta sección, dentro de los que se encuentran una definición de fibra óptica, tipos de fibra (monomodo y multimodo), tipos de rayos (meridionales y skew), y el tipo de fibra que se está analizando.
- Parámetros. Con esta opción podemos introducir los parámetros requeridos para llevar a cabo la simulación como los índices de refracción del núcleo y la cubierta de la fibra óptica, así como también, mostrar algunos valores calculados automáticamente.
- Simular. En esta opción se encuentran comandos para controlar la simulación, ya sea para ejecutarla o detenerla.
- Salir. Y una última opción, la cual como su nombre lo dice, se ocupa para terminar la simulación de la fibra óptica y regresar al menú principal.

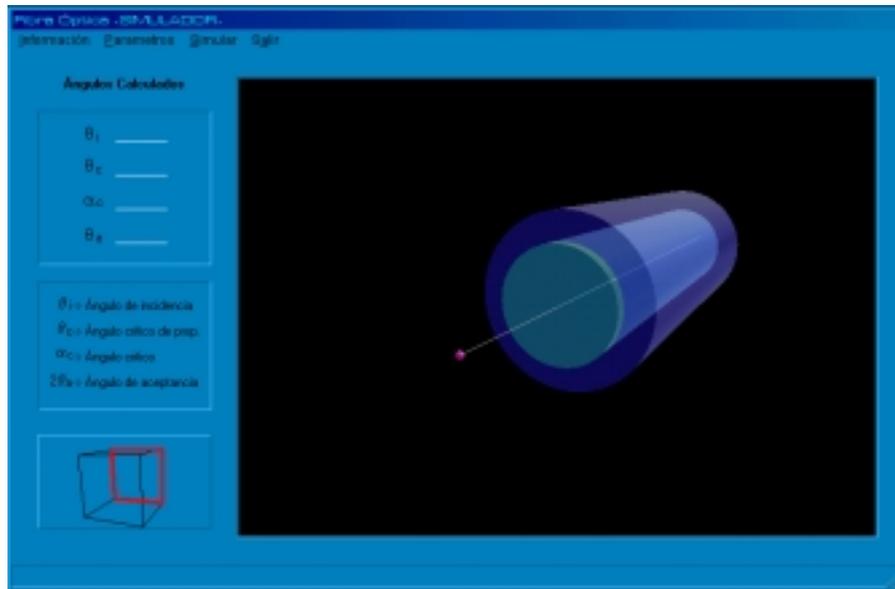


Figura 13. Interfaz de la simulación del fenómeno de reflexión total interna.

Para iniciar el proceso de simulación el usuario debe introducir los índices de refracción del núcleo y la cubierta de la fibra óptica. Esto con ayuda de la opción *Parámetros* del Menú antes mencionado. Al seleccionar y ejecutar **Establecer índices** de la opción *Parámetros*, se muestra una pantalla para introducir los índices como la que se observa en la figura 14.

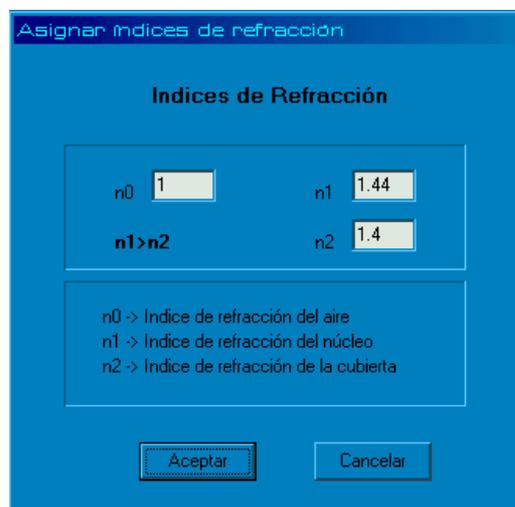


Figura 14. Captura de índices de refracción.

Una vez asignado los índices de refracción el software calcula automáticamente algunos valores importantes, los cuales pueden ser visualizados ejecutando **Mostrar Valores Calculados**, de la opción *Parámetros* del Menú “Simulador de Fibra Óptica”. La ventana en la que se muestran dichos resultados es mostrada en la figura 15.

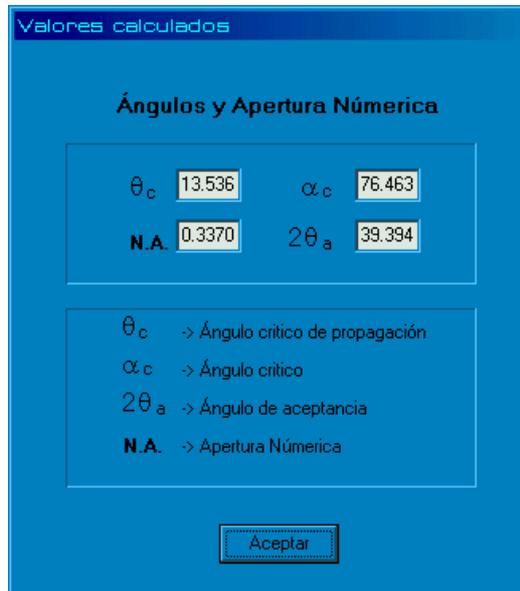


Figura 15. Ventana de consulta de los valores calculados automáticamente.

Una vez que se asignan los índices de refracción, el software calcula el ángulo de aceptación $2\theta_a$, y lo representa gráficamente en tercera dimensión como un cono de aceptación. Como se puede visualizar en el la figura 16.

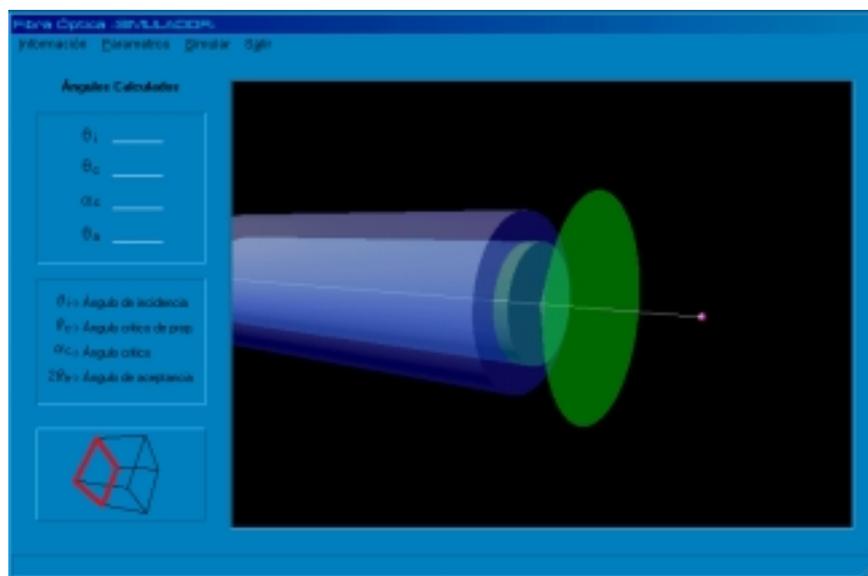


Figura 16. Representación del ángulo de aceptación en forma de cono.

Teniendo presente el ángulo de aceptación, se procede a asignar el ángulo de incidencia. Esto con el comando **Asignar ángulo de incidencia**, en la opción *Parámetros* del Menú ya

presentado. La ventana que nos permite asignar un valor al ángulo de incidencia es la que se observa en la figura 17. El valor de este ángulo debe estar dentro del rango del ángulo de aceptación.

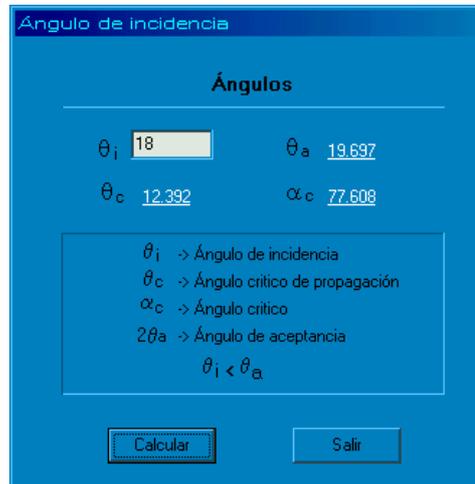


Figura 17. Asignar ángulo de incidencia.

Por último, se ejecuta la simulación eligiendo el comando **Ejecutar** contenido en la opción *Parámetros* del menú que contiene la interfaz. La simulación un rayo de luz que inciden en el centro de la fibra se visualizará, y dependiendo del ángulo de incidencia que se le haya asignado al rayo, éste se propagará dentro de la fibra óptica o se saldrá de ésta. Esto sucede así, si el ángulo de incidencia entra dentro del rango marcado por el ángulo de aceptación, el rayo continua su trayectoria dentro de la fibra óptica, pero si al ángulo de incidencia esta fuera del rango del ángulo de aceptación, el rayo saldrá de ésta como se observa en el figura 18.

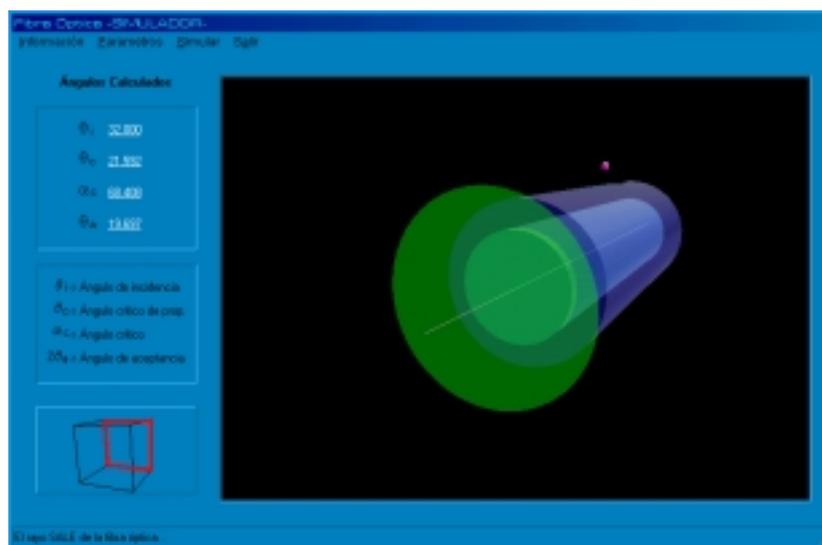


Figura 18. Simulación del rayo de luz saliendo de la fibra óptica.

d) Sección de Cálculos Técnicos

La presente sección es utilizada para realizar algunos cálculos de tipo técnico, los cuales son utilizados para la construcción de sistemas de comunicación vía fibra óptica.

Esta sección hace uso de la interfaz mostrada en la figura 19. La cual contiene únicamente un menú de opciones, en el que cada una de éstas se utilizan para introducir parámetros y obtener resultados.

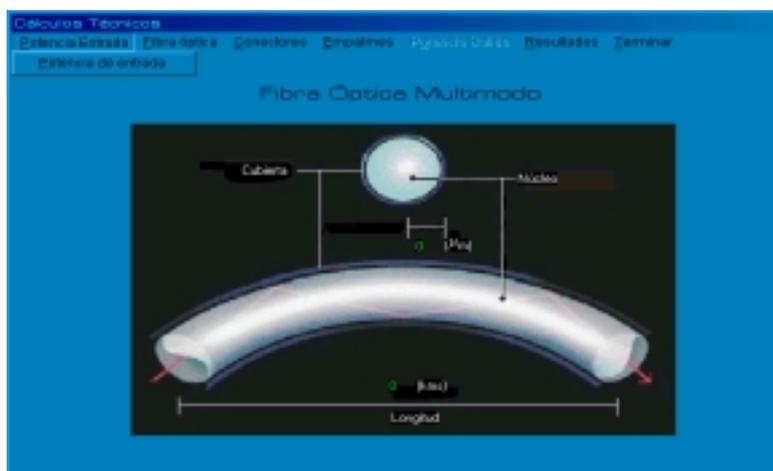


Figura 19. Interfaz de la sección cálculos técnicos.

Algunos parámetros deben ser introducidos en orden, principalmente los parámetros que se encuentran en la opción *Fibra Óptica* del Menú principal de esta interfaz, dentro de estos se encuentran los **índices de refracción**, como lo muestra la ventana que se obtiene ejecutando este comando visualizada en la figura 20, enseguida se debe de elegir la **Longitud de onda** como nos muestra la figura 21, para después continuar con la asignación del **Radio del núcleo** de la fibra óptica, y una vez teniendo todo lo anterior, se introduce la **Longitud** de la Fibra Óptica.



Figura 20. Asignando índices de refracción.

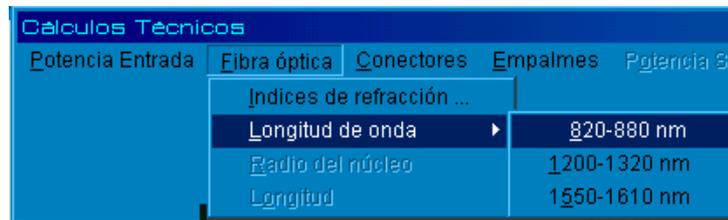


Figura 21. Seleccionando longitud de onda.

Las unidades de los parámetros que maneja este software están dados en:

- La longitud de onda en nanómetros (nm)
- El radio del núcleo en micrómetros (μm)
- La longitud de la fibra óptica en kilómetros (km)
- La potencia de entrada y salida, en microWatts (μW)
- Las pérdidas presentadas por los conectores y empalmes, en decibeles por kilómetro (dB/km)
- Y las pérdidas totales en decibeles (dB).

Para la asignación de potencia de entrada, se debe de elegir la opción Potencia entrada del menú principal de esta interfaz como se muestra en la figura 22.

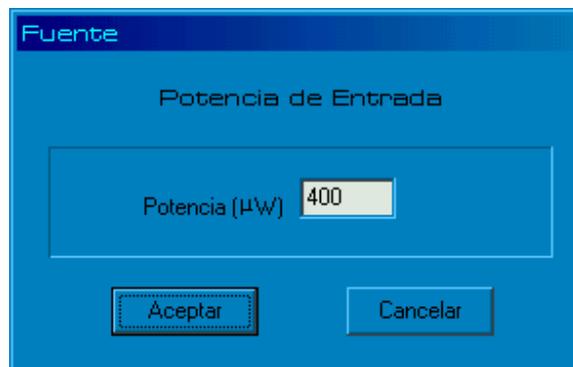


Figura 22. Seleccionando potencia de entrada.

Los parámetros que se necesitan para esta simulación son solicitados de la misma forma que el parámetro antes visto, la potencia de entrada. Utilizando el menú de opciones de esta interfaz se asignan valores, al número de conectores y empalmes, así como a la pérdida que presentan cada uno de estos en el uso.

Ahora, los resultados son mostrados en ventanas, por ejemplo para mostrar las pérdidas totales del sistema en construcción, se debe de elegir del menú *Resultados* la opción

Pérdidas, donde este resultado se mostrará en una ventana como la que se observa en la figura 23.

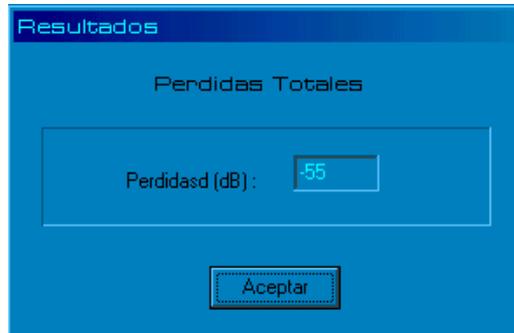


Figura 23. Ejemplo de la forma en que se muestran los resultados.

De esta misma manera, son mostrados los siguientes resultados:

- Potencia de entrada
- Número de modos
- Dispersión intermodal
- Y las pérdidas totales del sistema.

Para terminar esta sección únicamente se debe de elegir la opción *Terminar*.

Hasta este punto ya ha sido explicada cada una de las secciones que forman el sistema, y como se pudo observar la finalizar cada una de éstas secciones en forma individual, se tiene la oportunidad de regresar al menú principal del sistema, en donde se tiene presente la entrada a cualquiera de las tres secciones.