



**AEFO**

**PROGRAMA PARA EL ANÁLISIS DE ENLACES  
PUNTO A PUNTO DE FIBRA ÓPTICA**

José Miguel Milán Rubio



## INDICE DE CONTENIDO

<b>DEDICATORIA</b> .....	6
<b>INTRODUCCIÓN</b> .....	7
<b>LENGUAJE DE PROGRAMACIÓN</b> .....	8
VISUAL BASIC 6.0 .....	9
DEFINICIÓN DE UN PROYECTO EN VISUAL BASIC .....	10
<b>FUNDAMENTOS DE LA PROGRAMACIÓN</b> .....	12
Como funciona una aplicación controlada por eventos .....	12
<b>CONCEPTOS BÁSICOS SOBRE FIBRA ÓPTICA</b> .....	14
<b>INTRODUCCIÓN</b> .....	14
<b>VENTAJAS Y LIMITACIONES DE LAS FIBRAS ÓPTICAS</b> .....	14
Ventajas .....	14
Limitaciones .....	15
<b>LAS TÉCNICAS DE TRANSMISIÓN BÁSICAS</b> .....	17
<b>TIPOS DE FIBRAS ÓPTICAS</b> .....	18
Distintos tipos de fibras ópticas de acuerdo con el índice de refracción ....	20
<b>PÉRDIDAS EN LAS FIBRAS ÓPTICAS</b> .....	22
Pérdidas por curvatura .....	23
Pérdidas por acoplamiento .....	24
Pérdidas intrínsecas .....	24
Reflexión de Fresnel .....	26
<b>DISPERSIÓN Y ANCHO DE BANDA DE LAS FIBRAS ÓPTICAS</b> .....	27
Dispersión modal .....	27
Dispersión material .....	28
Dispersión de guía de onda .....	30
Ancho de banda para una fibra óptica .....	31
<b>GENERALIDADES SOBRE LOS EMISORES Y DETECTORES ÓPTICOS</b> .....	32
<b>EMISORES ÓPTICOS USADOS EN SISTEMAS DE F.O.</b> .....	33
Diodos emisores de luz (LEDs) .....	33
Diodos Láser de inyección. (ILD) .....	34
Características de las fuentes .....	35
<b>DETECTORES ÓPTICOS USADOS EN SISTEMAS DE F.O.</b> .....	37
Fotodiodos PN .....	37
Fotodiodos PIN .....	38
Fotodiodos de avalancha. (APD) .....	38
Características de los Detectores .....	39
<b>COMPONENTES PARA FIBRAS ÓPTICAS</b> .....	44
Conexiones y empalmes para fibras ópticas .....	45
Las causas de las pérdidas en una interconexión óptica .....	45
Pérdidas por inserción .....	48
Requerimientos de los conectores para fibras ópticas .....	49
Tipos de conectores .....	49
Conectores multimodo .....	51
Conectores Monomodo .....	52



<b>CONSIDERACIONES Y PROCEDIMIENTOS GENERALES PARA EL DISEÑO DE UN SISTEMA DE COMUNICACIONES POR FIBRAS ÓPTICAS.....</b>	<b>53</b>
<b>DESARROLLO DE LA SOLUCIÓN. DIAGRAMAS DE FLUJO.....</b>	<b>66</b>
DISEÑO TOP-DOWN DE PROGRAMAS.....	67
BASE DE DATOS.....	68
COMPATIBILIDAD.....	69
BALANCE DE POTENCIA.....	70
BALANCE DE TIEMPO.....	71
CALCULO DEL ENLACE ÓPTIMO.....	72
CALCULO DEL COMPONENTE OPTIMO.....	73
DISEÑO DEL ENLACE.....	75
TIEMPO DE SUBIDA.....	77
MODULO BALANCE POTENCIA.....	78
MODULO PRESUPUESTO.....	78
MODULO BALANCE TIEMPO.....	79
<b>DESARROLLO DE LA SOLUCIÓN MEDIANTE VISUAL BASIC 6.0.....</b>	<b>80</b>
PÁGINA PRINCIPAL.....	80
Descripción del formulario.....	80
BASE DE DATOS EMISORES.....	81
Descripción del formulario.....	81
Controles utilizados en el formulario.....	82
Descripción del código fuente.....	85
BASE DE DATOS RECEPTORES.....	90
Descripción del formulario.....	90
Controles utilizados en el formulario.....	91
Descripción del código fuente.....	94
BASE DE DATOS FIBRAS OPTICAS.....	99
Descripción del formulario.....	99
Controles utilizados en el formulario.....	100
Descripción del código fuente.....	103
COMPATIBILIDAD.....	108
Descripción del formulario.....	108
Controles utilizados en el formulario.....	110
Descripción del código fuente.....	111
BALANCE DE POTENCIA.....	119
Descripción del formulario.....	119
Controles utilizados en el formulario.....	121
Descripción del código fuente.....	123
BALANCE DE TIEMPO.....	130
Descripción del formulario.....	130
Controles utilizados en el formulario.....	132
Descripción del código fuente.....	133
ANÁLISIS DEL ENLACE DE FIBRA ÓPTICA.....	141
Descripción del formulario.....	141
Controles utilizados en el formulario Análisis del Enlace de Fibra Optica.....	143
Descripción del código fuente.....	145
DISEÑO DEL ENLACE OPTIMO.....	154
Descripción del formulario.....	154



Controles utilizados en el formulario Diseño del Enlace Optimo.....	160
Controles utilizados en el formulario Seleccionar Enlace .....	161
Controles utilizados en el formulario Seleccionar Emisor.....	162
Controles utilizados en el formulario Seleccionar Receptor .....	164
Controles utilizados en el formulario Seleccionar Fibra Optica .....	166
Descripción del código fuente .....	168
<b>SELECCIÓN DEL COMPONENTE ÓPTIMO .....</b>	<b>189</b>
Descripción del formulario.....	189
Controles utilizados en el formulario Selección del Componente Optimo..	191
Descripción del código fuente .....	193
<b>CÁLCULO DEL TIEMPO DE SUBIDA PARA UNA FIBRA ÓPTICA .....</b>	<b>203</b>
Descripción del formulario.....	203
Controles utilizados en el formulario Tiempo de Subida para una Fibra Optica .....	205
Descripción del código fuente .....	206
<b>CASO PRÁCTICO .....</b>	<b>210</b>
EJERCICIO .....	210
SOLUCIÓN OBTENIDA APLICANDO LAS ECUACIONES .....	211
SOLUCIÓN 1 .....	213
OBTENIDA UTILIZANDO EL PROGRAMA: DISEÑO DEL ENLACE.....	213
SOLUCIÓN 2 .....	222
OBTENIDA UTILIZANDO EL PROGRAMA: ELECCIÓN DEL COMPONENTE OPTIMO .....	222
<b>MANUAL DE USUARIO .....</b>	<b>225</b>
PANTALLA PRINCIPAL: SELECCIÓN DE MENÚS .....	225
BASES DE DATOS .....	226
Creación de un nuevo elemento .....	227
Edición de un elemento existente.....	227
Deshacer los cambios realizados .....	228
Eliminar un elemento de la base de datos .....	228
Ordenar los elementos almacenados en la base de datos.....	228
Seleccionar un elemento de la base de datos .....	229
COMPATIBILIDAD .....	230
BALANCE DE POTENCIA .....	232
BALANCE DE TIEMPOS .....	234
ANÁLISIS DEL ENLACE DE FIBRA .....	236
DISEÑO DEL ENLACE DE FIBRA .....	237
SELECCIÓN DEL COMPONENTE ÓPTIMO .....	240
CÁLCULO DEL TIEMPO DE SUBIDA.....	241
<b>CÓDIGO FUENTE .....</b>	<b>242</b>
BDEmisores.frm .....	242
BDReceptores.frm.....	246
BDFibras.frm .....	250
Compatibilidad.frm .....	254
GuardarVBL.frm .....	257
ResultadosV.frm.....	258
BalancePotencia.frm.....	259
GuardarBPT.frm .....	264
ResultadosBP.frm .....	265
BalanceTiempo.frm.....	266
GuardarBTP.frm .....	271
ResultadosBT.frm .....	272
Enlace.frm.....	273



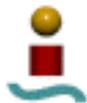
<b>GuardarENL.frm</b> .....	<b>280</b>
<b>ResultadosEnl.frm</b> .....	<b>281</b>
<b>ComponenteOptimo.frm</b> .....	<b>282</b>
<b>GuardarCOP.frm</b> .....	<b>291</b>
<b>ResultadosCOpt.frm</b> .....	<b>292</b>
<b>DiseñoEnl.frm</b> .....	<b>293</b>
<b>SeleccionarEnl.frm</b> .....	<b>299</b>
<b>SeleccionarTx.frm</b> .....	<b>300</b>
<b>SeleccionarRx.frm</b> .....	<b>303</b>
<b>SeleccionarFibras.frm</b> .....	<b>306</b>
<b>GuardarDEN.frm</b> .....	<b>309</b>
<b>Resultados.frm</b> .....	<b>310</b>
<b>Tsubida.frm</b> .....	<b>311</b>
<b>GuardarTSB.frm</b> .....	<b>314</b>
<b>ResultadosTs.frm</b> .....	<b>315</b>
<b>Abrir.frm</b> .....	<b>316</b>
<b>Buscar.frm</b> .....	<b>317</b>
<b>Menus.frm</b> .....	<b>318</b>
<b>Compatibilidad.frm</b> .....	<b>325</b>
<b>BalPotencia.bas</b> .....	<b>327</b>
<b>BalTiempos.frm</b> .....	<b>328</b>
<b>Presupuesto.frm</b> .....	<b>328</b>
<b>Redondear.frm</b> .....	<b>328</b>
<b>Resultado.frm</b> .....	<b>329</b>
<b>GestionUdl.frm</b> .....	<b>329</b>



## DEDICATORIA

De José Miguel:

Este Proyecto esta dedicado en su totalidad a mis padres, José Jesús y María de los Angeles, como reconocimiento a su esfuerzo para que sus hijos disfrutarán de la mejor educación posible, que es el mejor legado que unos padres te pueden dejar. Gracias.



## INTRODUCCIÓN

El trabajo desarrollado en este Proyecto pretende ser un complemento y un apoyo para los futuros ingenieros, interesados en el campo de las Comunicaciones Ópticas.

En el contenido del Proyecto se recogen las nociones básicas necesarias para el análisis de los enlaces punto a punto construidos mediante fibras ópticas, y aporta un software denominado AEFO, como herramienta de trabajo para la realización de este tipo de análisis.

Pienso que un software como complemento a la parte práctica de las asignaturas constituye un instrumento tremendamente útil para el desarrollo de la actividad académica del alumno, permitiéndole a éste disponer de una guía de consulta y una fuente para la comprobación de los resultados obtenidos.



## LENGUAJE DE PROGRAMACIÓN

Los estilos de programación han experimentado una continua evolución, al ritmo marcado por los cambios producidos en el diseño de las aplicaciones. De una programación secuencial se paso a una programación modular, para llegar a una programación estructurada. A partir de aquí los esfuerzos se centraron en potenciar la modularidad y la reutilización de código, lo que nos condujo a la programación orientada a objetos.

La programación orientada a objetos (POO) es una forma de programación que utiliza objetos, ligados mediante mensajes, para la resolución de problemas. Los mecanismos básicos de la programación orientada a objetos son: objetos, mensajes, métodos y clases.

### Objetos

Un programa tradicional se compone de procedimientos y de datos. Un programa orientado a objetos se compone solamente de objetos. Un objeto es una encapsulación genérica de datos y de procedimientos para que sean manipulados. Visto de otra manera, un objeto es una entidad con unos atributos particulares, las propiedades, y unas formas de operar sobre ellos, los métodos. Por lo tanto, un objeto contiene, por una parte, operaciones que definen su comportamiento, y por otra, variables manipuladas por esas operaciones que definen su estado.

### Mensajes

Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo, interpretando y respondiendo a mensajes de otros objetos.

### Métodos

Un método se implementa sobre una clase de objetos y determina cómo debe actuar el objeto cuando recibe un mensaje. En adición, las propiedades permitirán almacenar información para dicho objeto. Un método puede también enviar mensajes a otros objetos solicitando información o una acción. La estructura más interna de un objeto está oculta para otros usuarios y la única conexión que tiene con el exterior son los mensajes. Los datos que están dentro de un objeto solamente pueden ser manipulados por los métodos asociados al propio objeto.

### Clases

Una clase es un tipo de objeto definido por un usuario, equivalente a la generalización de un tipo específico de objetos.



Las características fundamentales de la programación orientada a objetos son: abstracción, encapsulamiento, herencia y polimorfismo.

### Abstracción

Nos permite generalizar y centrarnos en los aspectos que nos proporcionan una visión global del tema.

### Encapsulamiento

Permite ver un objeto como una caja negra en la que se ha metido de alguna manera toda la información relacionada con dicho objeto. Esto nos permitirá manipular los objetos como unidades básicas, permaneciendo oculta su estructura interna. La abstracción y la encapsulación están representadas por la clase. La clase es una abstracción porque en ella se definen las propiedades de un determinado conjunto de objetos con características comunes, y es una encapsulación, porque constituye una caja negra que encierra tantos los datos que almacenan los objetos como los métodos que permiten manipularlos.

### Herencia

Es el mecanismo para compartir automáticamente métodos y datos entre clases y subclases de objetos.

### Polimorfismo

Permite implementar múltiples formas de un método, dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto hace que se pueda acceder a una variedad de métodos distintos utilizando el mismo medio de acceso.

## **VISUAL BASIC 6.0**

La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente puede arrastrar y colocar objetos prefabricados en su lugar dentro de la pantalla.

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

El lenguaje de programación Visual Basic no es exclusivo de Visual Basic. La Edición para aplicaciones del sistema de programación de Visual Basic, incluida en Microsoft Excel, Microsoft Access y muchas otras aplicaciones Windows, utilizan el mismo lenguaje. El sistema de programación de Visual Basic, Scripting Edition (VBScript) para programar en Internet es un subconjunto del lenguaje Visual Basic.



Visual Basic dispone de las siguientes herramientas:

- Las características de acceso a datos permiten crear bases de datos y aplicaciones cliente para los formatos de las bases de datos más conocidas, incluidos Microsoft SQL Server y otras bases de datos de ámbito empresarial.
- Las tecnologías ActiveX™ permiten utilizar la funcionalidad proporcionada por otras aplicaciones, como el procesador de textos Microsoft Word, la hoja de cálculo Microsoft Excel y otras aplicaciones Windows. Puede incluso automatizar las aplicaciones y los objetos creados con la Edición profesional o la Edición empresarial de Visual Basic.
- Las capacidades de Internet facilitan el acceso a documentos y aplicaciones a través de Internet desde su propia aplicación.
- La aplicación terminada es un auténtico archivo .exe que utiliza una biblioteca de vínculos dinámicos (DLL) de tiempo de ejecución que puede distribuir con toda libertad.

Soporta la abstracción, la encapsulación, el polimorfismo y la reutilización del código.

Los objetos de Visual Basic están encapsulados; es decir, contienen su propio código y sus propios datos.

Los objetos de Visual Basic tienen propiedades, métodos y eventos. Las propiedades son los datos que describen un objeto. Los eventos son hechos que pueden ocurrir sobre un objeto. Un método agrupa el código que se ejecuta en respuesta a un evento.

Al conjunto de propiedades y métodos se le llama interfaz. Además de su interfaz predeterminada, los objetos pueden implementar interfaces adicionales para proporcionar polimorfismo, permitiéndonos manipular muchos tipos diferentes de objetos sin preocuparnos de su tipo.

La reutilización del código es la capacidad de trasladar características de un objeto a otro, lo que se logra con alguna forma de herencia.

## DEFINICIÓN DE UN PROYECTO EN VISUAL BASIC

Para crear una aplicación con Visual Basic se trabaja con proyectos. Un *proyecto* es una colección de archivos que se usan para generar una aplicación.

Al crear una aplicación probablemente se crearán nuevos formularios; también se puede volver a usar o modificar formularios creados en proyectos anteriores. Esto también se aplica a otros módulos o archivos que se incluyan en el proyecto. Los controles ActiveX y los objetos de otras aplicaciones también se pueden compartir entre proyectos.

Después de ensamblar todos los componentes de un proyecto y escribir el código, compilamos el proyecto para crear un archivo ejecutable.

Cuando se desarrolla una aplicación, se trabaja con un archivo de proyecto para administrar todos los diferentes archivos que se crean. Un proyecto consta de lo siguiente:

- Un archivo de proyecto que realiza el seguimiento de todos los componentes (.vbp)
- Un archivo para cada formulario (.frm)



- Un archivo de datos binario para cada formulario que contenga datos sobre propiedades de controles del formulario (.frm). Estos archivos no se pueden modificar y los genera automáticamente cualquier archivo .frm que tenga propiedades en formato binario, como Picture o Icon.
- Opcionalmente, un archivo para cada módulo de clase (.cls)
- Opcionalmente, un archivo para cada módulo estándar (.bas)
- Opcionalmente, uno o más archivos con controles ActiveX (.ocx)
- Opcionalmente, un único archivo de recursos (.res)

El *archivo de proyecto* es simplemente una lista de todos los archivos y objetos asociados con el proyecto, así como información sobre las opciones de entorno establecidas. Esta información se actualiza cada vez que guarda el proyecto. Todos los archivos y objetos también se pueden compartir con otros proyectos.

Cuando ha completado todos los archivos del proyecto puede convertir el proyecto en un archivo ejecutable (.exe) eligiendo en el menú Archivo el comando Generar *proyecto.exe*.

### Formularios

Un formulario es una ventana. La ventana Windows de cualquier aplicación. Podemos abrir tantas ventanas como queramos en nuestro proyecto, pero el nombre de las ventanas debe ser distinto. Por defecto la ventana que se abre en Visual Basic tiene el nombre de Form1.

Los módulos de formularios (extensión de nombre de archivo .frm) pueden contener descripciones en forma de texto del formulario y sus controles, incluyendo los valores de sus propiedades. También pueden contener declaraciones a nivel de formulario de constantes, variables y procedimientos externos, procedimientos de evento y procedimientos generales.

### Módulos de clase

Los módulos de clase (extensión de nombre de archivo .cls) son similares a los módulos de formulario, excepto en que no tiene interfaz de usuario visible.

### Módulos estándar

Un módulo es un archivo Visual Basic donde escribimos parte del código de nuestro programa (puede haber código en el formulario también). Las rutinas incluidas dentro de los módulos pueden ser ejecutadas desde los formularios de la aplicación.

Los módulos estándar (extensión de nombre de archivo .bas) pueden contener declaraciones públicas o a nivel de módulo de tipos, constantes, variables, procedimientos externos y procedimientos públicos.

### Archivos de Recursos

Los archivos de recursos (extensión de nombre de archivo .res) contienen mapas de bits, cadenas de texto y otros datos que puede modificar sin volver a modificar el código. Por ejemplo, si se piensa traducir la aplicación a un idioma extranjero, se pueden guardar todas las



cadena de texto de la interfaz de usuario y los mapas de bits en un archivo de recursos, y simplemente traducir el archivo de recursos en vez de la aplicación completa. Un proyecto sólo puede contener un archivo de recursos.

### Controles Active X

Los controles ActiveX (extensión de nombre de archivo .ocx) son controles opcionales que se pueden agregar al cuadro de herramientas y se pueden usar en formularios. Cuando instala Visual Basic, los archivos que contienen los controles incluidos en Visual Basic se copian a un directorio común (el subdirectorio \Windows\System en Windows 95). Existen controles ActiveX adicionales disponibles en diversas fuentes. También puede crear sus propios controles mediante las ediciones Profesional y Empresarial de Visual Basic.

### Controles estándar

Los controles estándar los proporciona Visual Basic. Los controles estándar, como CommandButton (botón de comando) o Frame (marco), siempre están incluidos en el cuadro de herramientas, al contrario de lo que ocurre con los controles ActiveX y los objetos insertables, que se pueden agregar y quitar del cuadro de herramientas.

## **FUNDAMENTOS DE LA PROGRAMACIÓN**

Cada módulo de formulario contiene *procedimientos de evento* (secciones de código donde se colocan las instrucciones que se ejecutarán como respuesta a eventos específicos). Los formularios pueden contener controles. Por cada control de un formulario, existe el correspondiente conjunto de procedimientos de evento en el módulo de formulario. Además de procedimientos de evento, los módulos de formulario pueden contener procedimientos generales que se ejecutan como respuesta a una llamada desde cualquier procedimiento de evento.

El código que no esté relacionado con un control o un formulario específico se puede colocar en un tipo diferente de módulo, un *módulo estándar* (.bas). Se deben colocar en un módulo estándar los procedimientos que se puedan utilizar como respuesta a eventos de diversos objetos, en lugar de duplicar el código en los procedimientos de evento de cada objeto.

### **Como funciona una aplicación controlada por eventos**

Un evento es una acción reconocida por un formulario o un control. Las aplicaciones controladas por eventos ejecutan código Basic como respuesta a un evento. Cada formulario y control de Visual Basic tiene un conjunto de eventos predefinidos. Si se produce uno de dichos eventos y el procedimiento de evento asociado tiene código, Visual Basic llama a ese código.

Aunque los objetos de Visual Basic reconocen automáticamente un conjunto predefinido de eventos, usted decide cuándo y cómo se responderá a un evento determinado. A cada evento le corresponde una sección de código (un procedimiento de evento). Cuando desea que un control responda a un evento, escribe código en el procedimiento de ese evento.



Los tipos de eventos reconocidos por un objeto varían, pero muchos tipos son comunes a la mayoría de los controles. Por ejemplo, la mayoría de los objetos reconocen el evento Click: si un usuario hace clic en un formulario, se ejecuta el código del procedimiento de evento Click del formulario; si un usuario hace clic en un botón de comando, se ejecuta el código del procedimiento de evento Click del botón. El código en cada caso será diferente.

He aquí una secuencia típica de eventos en una aplicación controlada por eventos:

1. Se inicia la aplicación y se carga y muestra un formulario.
2. El formulario (o un control del formulario) recibe un evento. El evento puede estar causado por el usuario (por ejemplo, por la pulsación de una tecla), por el sistema (por ejemplo, un evento de cronómetro) o, de forma indirecta, por el código (por ejemplo, un evento Load cuando el código carga un formulario).
3. Si hay código en el procedimiento de evento correspondiente, se ejecuta.
4. La aplicación espera al evento siguiente.



## CONCEPTOS BÁSICOS SOBRE FIBRA ÓPTICA

### INTRODUCCIÓN

Una fibra óptica es, en forma sencilla, un filamento muy fino de plástico o vidrio a través del cual la luz puede propagarse. Una fibra óptica puede ser estudiada fácilmente si se la considera como un "tubo transparente" circular en el cual los rayos de luz que entran por una punta pueden ser conducidos o guiados por el interior de la misma hasta su extremo opuesto. Las fibras son capaces de transportar señales ópticas y energía luminosa, y constituyen el equivalente óptico de un "par conductor" o un cable coaxial. El uso de fibras ópticas en lugar de estos medios tradicionales supone varias ventajas pero también se presentan algunas limitaciones.

Las fibras no transportan señales eléctricas, sino el equivalente óptico de las mismas. Como la mayoría de los sistemas electrónicos de comunicaciones funcionan con señales eléctricas, la información a transmitir debe ser primero convertida en una señal luminosa mediante el uso de algunos de los distintos tipos de transductores que existen, para luego ser enviada por la fibra al lugar deseado. En él, la recuperación de la señal original supone la existencia de otros transductores que actúan a la inversa de los primeros.

### VENTAJAS Y LIMITACIONES DE LAS FIBRAS ÓPTICAS

#### Ventajas

1- Gran ancho de banda. Una fibra óptica es el medio físico con mayor capacidad de transmitir información que existe en la actualidad. Esta mayor capacidad puede explicarse por dos razones, por un lado, el mayor ancho de banda que pueden soportar (lo cual posibilita el manejo simultáneo de gran cantidad de canales independientes), y por otro lado, la gran velocidad de propagación de la señal.

2- Bajas pérdidas. Las características de baja atenuación y baja dispersión de las modernas fibras ópticas permite lograr satisfactoriamente enlaces de hasta 50 km., que no necesitan estaciones repetidoras. Las estaciones repetidoras para sistemas de comunicaciones son habitualmente una de las partes más costosas del sistema.

3- Tamaño reducido. El reducido diámetro de una fibra hace que las mismas resulten muy apropiadas cuando el espacio utilizable para disponer un canal de comunicaciones es mínimo, por ejemplo:

- Aeronaves y Submarinos.
- Conductos de telefonía urbana subterráneos.
- Salas de computadoras, para las conexiones entre el equipamiento.



4- Peso reducido. La relación entre capacidad de canales respecto del peso en una red implementada con Fibras ópticas es sumamente conveniente.

5- Inmunidad a las interferencias electromagnéticas (EMI) y de radiofrecuencia (RFI). A diferencia de los conductores por los cuales circulan corrientes, las fibras ópticas (por las cuales solo se propaga luz) no irradian ni son afectadas o interferidas por ninguna forma de energía electromagnética.

La radiación electromagnética es una forma de contaminación ambiental que se ha tornado sumamente problemática en los últimos tiempos debido a la proliferación de equipos electrónicos en todos los ámbitos de la vida cotidiana. Como las fibras ópticas, no producen ni reciben radiación electromagnética, constituyen el medio de transmisión ideal en lo que concierne a la inmunidad a los ruidos ( EMI/RFI ). Además, como a través de la fibra propiamente dicha, no circulan corrientes, existe la ventaja adicional de que se constituyen en el medio más seguro de transmitir información en ambientes potencialmente explosivos, o entre equipamiento que trabaja con altas tensiones.

6- Seguridad: Sólo es posible interferir la señal que se propaga a través de una fibra interceptándola en forma individual, lo que normalmente es fácilmente detectable.

7- Flexibilidad de los sistemas: Los sistemas que se implementen en base a canales soportados por fibras ópticas seguramente podrán , con el tiempo, ser mejorados a medida que se presenten progresos en la tecnología de los emisores y los detectores usados.

## **Limitaciones**

1- Dificultades para la manipulación de las fibras. El equipamiento, y las herramientas que se necesitan para trabajar con fibras ópticas no es convencional y requiere de personal capacitado y entrenado para su manejo.

2- Baja eficiencia de los dispositivos que se usan como interfaz con el equipamiento electrónico.

3- La no linealidad de las fuentes ópticas y detectores, limitan las aplicaciones analógicas.

4- Sin polaridad. A diferencia de lo que ocurre con líneas conductoras, no es posible trabajar con señales de tres niveles ( Positivo - Negativo – Cero ).

5- Dependencia. Como la fibra óptica casi siempre está en interfaz con equipos y dispositivos electrónicos, casi siempre se termina necesitando una línea conductora eléctrica adicional.

6- Reparación complicada. Las instalaciones de cables con fibras ópticas son muy sensibles al daño intencional (sabotaje), y su reparación puede ser complicada y costosa.



Tabla de comparación de características principales entre los distintos medios

MEDIO	DISTANCIA (rango)	ANCHO DE BANDA	PERDIDAS (a 20MHz)	INMUNIDAD EMI/RFI	SEGURIDAD (Contra uso clandestino)	COSTO	COMENTARIOS
Cables de Fibras ópticas	2-10 km	>500 MHz	1-8 dB/km	Excelente	Muy buena	Elevado	Costos con tendencia a bajar rápidamente.
Línea telefónica (convencional, PBX)	---	< 50 KHz.	Elevado	Pobre	Pobre	Bajo	Bajos costos de instalación.
Cable multi-conductor plano.	< 100 m	<10 MHz.	Elevado	Pobre	Razonable	Elevado	Se puede conseguir blindado.
Par retorcido (blindado)	<1200 m	<100 KHz	Elevado	Muy buena	Razonable	Bajo	Económico para distancias limitadas.
Balanceado axial (Twinax)	< 1km	<200MHz	50/100 dB/km	Muy buena	Razonable	Elevado	Impedancia controlada
Cable coaxial	< 2km	<400MHz	6/60 dB/km	Razonable	Razonable	Bajo a muy elevado	
Radiación infrarroja	< 300 m	< 20 MHz	Elevado	Buena	Razonable	Habitualmente elevado. Potencialmente bajo	(Para espacio libre); sigue línea visual; rango reducido con niebla (depende de la humedad)
Microondas (radiación)	< 50 km horizontal)	< 100 MHz		Pobre	Razonable	Elevado	(Para espacio libre); sigue línea visual. Las pérdidas dependen del tamaño de la antena.

Tabla comparativa de las propiedades principales de los distintos tipos de cables.

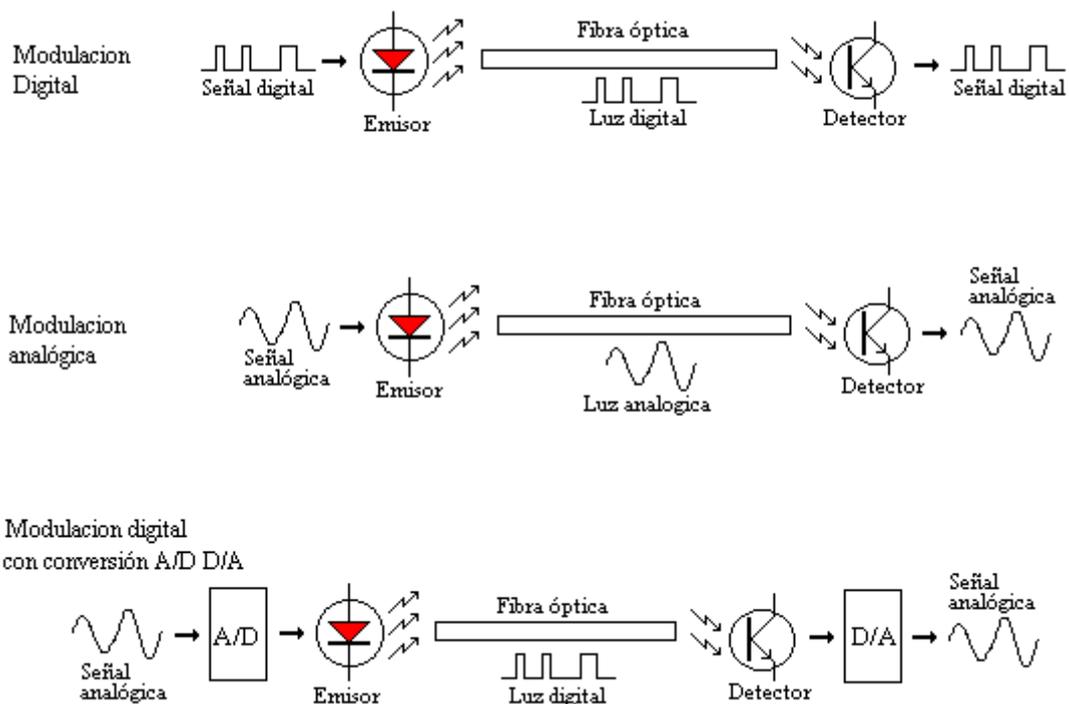
Propiedades	Fibras Ópticas	Cable Coaxil	Par Retorcido
Inmunidad a los ruidos (RFI/EMI)	Si	No	No
Aislamiento eléctrico total	Si	No	No
Elevada seguridad de transmisión.	Si	No	No
Baja modulación cruzada	Si	No	No
Peligro de chispas o fuego	No	Si	Si



## LAS TÉCNICAS DE TRANSMISIÓN BÁSICAS

Para que una determinada señal eléctrica pueda ser transmitida por una fibra óptica, esta debe usarse para modular una fuente de luz tal como un diodo emisor de luz (LED) o un láser. La luz modulada se acopla a la fibra en cuyo extremo opuesto hay un detector óptico que recibe la luz modulada y recupera la señal original de entrada.

La principal diferencia entre las distintas técnicas de transmisión radica en la manera como se efectúa la modulación de la luz. Desde este punto de vista se puede hacer una división en tres categorías básicas: modulación digital, modulación analógica, y modulación digital con conversión A/D y D/A.



Tres maneras de transmisión básica por fibra óptica

- La modulación digital simplemente cambia la secuencia de ceros y unos (que normalmente se corresponden a niveles de voltaje) en una secuencia idéntica de encendido y apagado de la fuente emisora de luz. Debido a que todas las comunicaciones entre computadoras se establecen en base a señales digitales, este tipo de modulación es adecuada para la transmisión de datos entre computadoras.

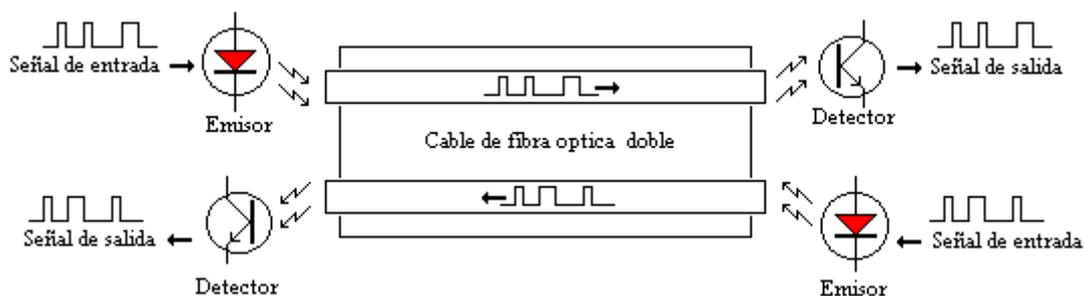
- Las señales de comunicación de tipo analógicas (voz o video) pueden usarse para hacer variar la intensidad de la fuente emisora de luz. Esta técnica es relativamente barata y fácil de aplicar pero adolece de la limitación impuesta, sobre todo, por la no linealidad de los dispositivos emisores y receptores ópticos (no así de la fibra propiamente dicha, que en principio es capaz de transmitir luz en un elevado y muy lineal rango dinámico).

-



- Para resolver el inconveniente que presenta la modulación directa con señales analógicas, pueden convertirse las mismas previamente a un formato digital utilizando un convertor A/D, modular la fuente emisora en forma digital, y recuperar en la salida la señal original mediante un convertor D/A. Existe además la ventaja adicional que usando equipamiento de multiplexado, puedan agregarse un gran número de señales adicionales para su transmisión simultánea.

La gran mayoría de los sistemas de transmisión de información, requieren una comunicación simultánea en ambos sentidos ( sistemas dúplex ), y aunque en principio, una fibra óptica es un elemento bidireccional y reversible ( es decir que cualquiera de los dos extremos de una fibra pueden usarse para emitir o recibir, y viceversa ), es habitual el método de disponer dos grupos idénticos de dispositivos de modulación y detección en sentido opuesto además de dos fibras ( una para la transmisión y otra para la recepción ) para formar un sistema bidireccional completamente funcional.



Comunicación bidireccional

## TIPOS DE FIBRAS ÓPTICAS

En este apartado se intentará hacer una clasificación básica de los tipos de fibras ópticas más comúnmente usadas en telecomunicaciones, es decir las de vidrio. Normalmente en aplicaciones prácticas reales, se utilizan conjuntos o haces de fibras ópticas, empaquetadas en un único envoltorio que se denomina "Cable de fibras ópticas". Suele ser muy común que el fabricante de Cables, no se a su vez quien produce la fibra propiamente dicha. Una discusión y clasificación más detallada de los distintos tipos de cables se efectuara más adelante. Las fibras ópticas se diferencian entre si de acuerdo con las dimensiones de la misma y en relación con el índice de refracción del material empleado. La tabla siguiente muestra cuales son las dimensiones de los principales tipos de fibras utilizadas en telecomunicaciones.

El lenguaje habitual para designar una fibra es en correspondencia con el tamaño de la misma, y normalmente se especifica con la relación "Núcleo/Revestimiento". Por ejemplo, el tipo II se conoce como fibra 50/125.



Tipo	Núcleo	Revestimiento	Recubrimiento	Tubo o vaina
I	4 a 10 (típicamente 8)	125	250 a 500	900 a 2000
II	50	125	250 a 500	900 a 2000
III	62,5	125	250 a 500	900 a 2000
IV	85	125	250 a 500	900 a 2000
V	100	140	250 a 500	900 a 2000

Diámetros de una fibra óptica y de su protección. ( en  $\mu\text{m}$ )

### Tipo I

Las fibras 4/125 a 10/125 son normalmente monomodo. Toleran un elevado ancho de banda, son de baja dispersión y atenuación, pero son las más difíciles de manipular y requieren el uso de transductores de gran calidad y conectores de alta precisión. Se utilizan frecuentemente para aplicaciones de transmisión de datos a alta velocidad o para largas distancias. Si bien la fibra en sí es más barata que las del tipo multimodo, el equipamiento que se requiere es mucho más caro.

### Tipo II

Todas las fibras cuyo núcleo tiene un diámetro mayor a  $50\mu\text{m}$  son del tipo multimodo. Las fibras 50/125 fueron las primeras fibras para telecomunicaciones que se vendieron masivamente. Son de pequeña apertura numérica y por consiguiente la relación entre la potencia que se acopla a la fibra y la emitida por la fuente es normalmente reducida. Sin embargo, de todas las fibras multimodo, es la que tiene mayor ancho de banda.

### Tipo III

Las fibras 62,5/125, es en la actualidad la más popular de las fibras multimodo. Tiene un ancho de banda ligeramente menor que la 50/125, pero su apertura numérica es mayor y posibilita un mejor acoplamiento de las fuentes. Posee también menos pérdidas por microcurvaturas.

### Tipo IV

El tamaño 80/125 se usa principalmente en Europa, sus características son similares a la de tipo II, y como el diámetro del revestimiento es idéntico (  $125 \mu\text{m}$ ) se pueden usar los mismos conectores y técnicas de empalme.

### Tipo V

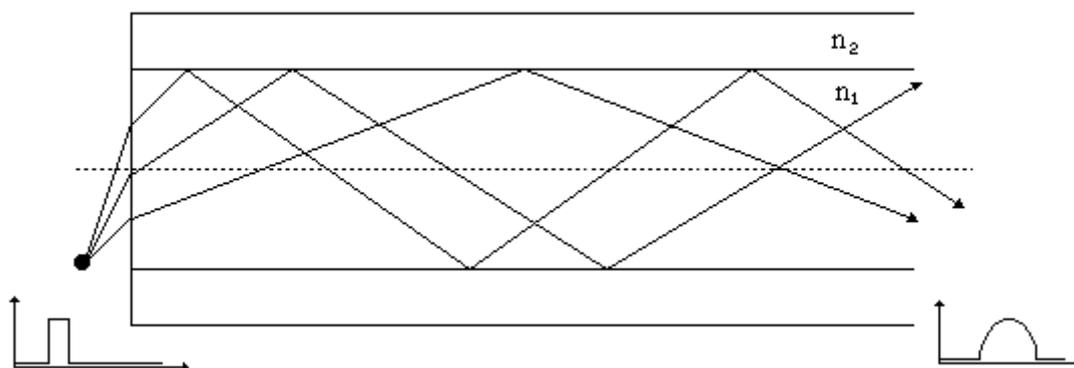
El diámetro del núcleo de la fibra 100/140, la convierte en la fibra más fácil de manipular, es menos sensible a las tolerancias de los conectores, y a la acumulación de suciedad en los mismos. Acopla la mayor cantidad de luz de las fuentes, pero tiene un ancho de banda más reducido que las de tamaño más pequeño. Por esto se las utiliza principalmente en sistemas de comunicaciones de reducida longitud, con varios conectores y que no requieren elevada velocidad de transferencia de datos ( por ejemplo: en instalaciones dentro de edificios ). No es muy común y suele ser difícil de conseguir.



## Distintos tipos de fibras ópticas de acuerdo con el índice de refracción

### Fibras de índice escalonado

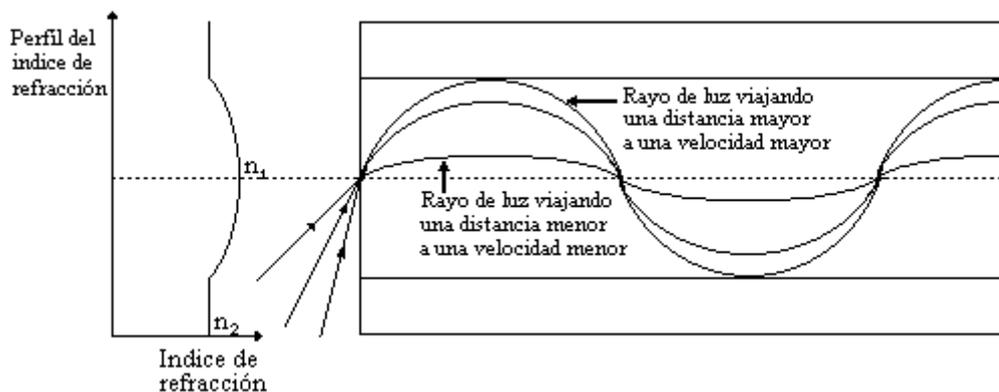
Respecto del índice de refracción del material empleado, existen básicamente dos tipos de fibras ópticas. Por un lado están aquellas denominadas de índice escalonado; en este tipo de fibra, el núcleo de la misma, posee un índice de refracción homogéneo en toda su sección, y lo mismo ocurre con el índice de refracción del revestimiento



En el dibujo precedente corresponde a un esquema de una fibra multimodo de índice escalonado donde, como puede verse, los rayos de luz que parten al mismo tiempo de una fuente acoplada a un extremo de la fibra, pueden llegar a la otra punta en instantes diferentes en razón de que las distancias recorridas también son diferentes. Esto trae como consecuencia que un pulso transmitido se termine ensanchando en el tiempo. Por este motivo, las fibras multimodo de índice escalonado, que fueron las primeras fibras en fabricarse (y durante un tiempo las únicas disponibles para propósitos prácticos), han ido cayendo poco a poco en desuso, siendo sustituidas por un tipo más moderno de fibras denominadas de índice gradual; en cambio no ha sucedido así con las fibras monomodo de índice escalonado, que se continúan fabricando y utilizando.

### Fibras de índice gradual

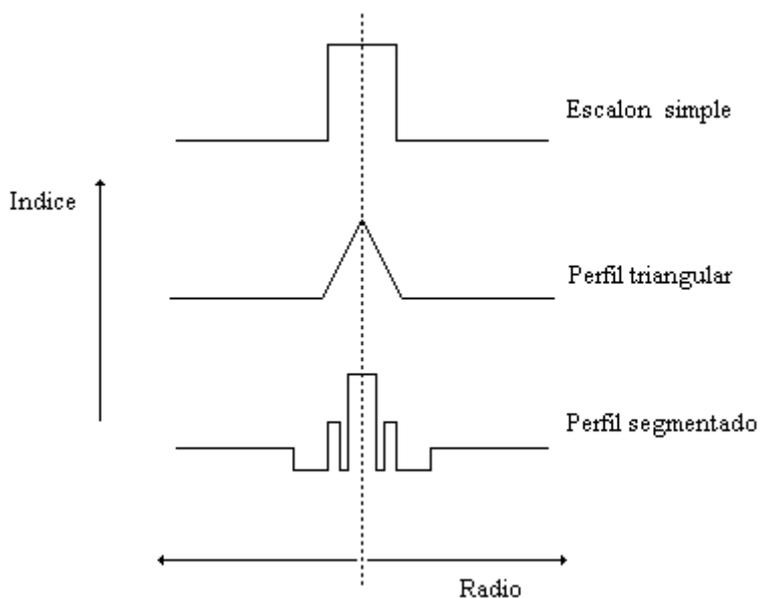
El índice de refracción del núcleo de una fibra de índice gradual no es de un valor homogéneo en toda la sección del mismo, sino que tiene un valor máximo en el centro que luego va decreciendo con el radio hasta hacerse igual al del revestimiento en la frontera con este. Como consecuencia de esto, los rayos de luz siguen una trayectoria sinuosa en vez de un camino en zigzag, además, los caminos más largos son más rápidos, (porque el índice de refracción decrece al acercarse al revestimiento) y son más lentos en el centro del núcleo (donde el índice de refracción es más elevado). Si el perfil puede ajustarse de manera que la variación de la longitud de los caminos, sean compensados por la correspondiente variación de la velocidad, el efecto de ensanchamiento de un pulso de luz que viaja por la fibra, se ve notablemente minimizado.



Las fibras de índice gradual ( que son normalmente multimodo), aparecieron como una solución al problema del reducido ancho de banda soportado por las fibras multimodo de índice escalonado.

### Fibras con perfiles especiales

Hasta ahora las fibras monomodo han sido siempre de índice escalonado, sin embargo desde hace poco tiempo se han comenzado a producir también fibras monomodo con perfiles que no pueden ser catalogados estrictamente dentro de los escalonados. Se busca de esta forma lograr ciertas mejoras como por ejemplo, reducir las pérdidas por curvatura, aumentar la apertura numérica y reducir la dispersión cromática, (que ocurre como resultado de las diferentes velocidades de propagación de las componentes espectrales de una fuente de luz que no emite en una única longitud de onda).



Algunos perfiles especiales de fibras monomodos de última generación.



Resumiendo:

Tipos de fibras ópticas	Perfil	Características principales.	Comentarios
Multimodo	<ul style="list-style-type: none"><li>De índice escalonado</li></ul>	Gran NA. Reducido ancho de banda.	Actualmente su utilización es mínima, (prácticamente nula en telecomunicaciones)
	<ul style="list-style-type: none"><li>De índice gradual</li></ul>	Gran NA Mejor ancho de banda.	El equipo óptico que se requiere es económico.
Monomodo	<ul style="list-style-type: none"><li>De índice escalonado</li></ul>	Muy pequeña NA. Gran ancho de banda	Requieren equipo óptico y de conectorización de precisión
	<ul style="list-style-type: none"><li>De perfiles especiales</li></ul>	Menor pérdidas por curvatura Menor dispersión cromática.	Se busca reducir requerimientos del equipo óptico.

## PÉRDIDAS EN LAS FIBRAS ÓPTICAS

Cuando una fuente óptica se acopla a la punta de una fibra, la cantidad de luz que llega al extremo opuesto es en general menor que la original; lógicamente esto se debe a que en el camino se producen una serie de pérdidas. Por un lado es evidente que no toda la energía emitida por la fuente, puede acoplarse a la fibra, pero además, una vez que un rayo de luz ha penetrado en la misma, también puede ir sufriendo una disminución de su intensidad con la distancia a medida que viaja por el núcleo.

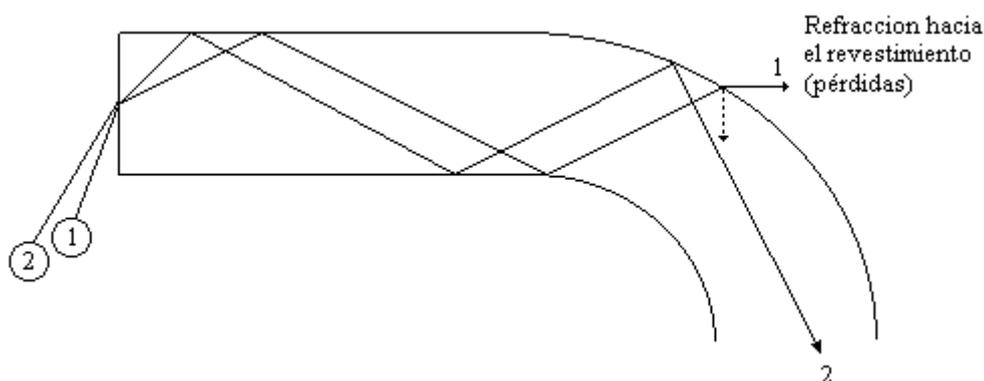
Las pérdidas pueden clasificarse, según las causas que las provocan, en pérdidas debidas a factores externos, y pérdidas por motivos intrínsecos de las fibras. Entre las primeras están por un lado las que se deben a los empalmes y conexiones entre diferentes tramos de un sistema, y por el otro lado las que se producen debido a que al efectuar el tendido de un cable de fibras ópticas, es imposible seguir siempre trayectorias rectas y en algún lugar el cable debe forzosamente curvarse.

Las pérdidas por causas intrínsecas son: la atenuación por absorción del vidrio; las pérdidas que se derivan de imperfecciones o defectos del vidrio y que se conocen como dispersión de Rayleigh; las pérdidas por defectos mecánicos que se producen en el proceso de fabricación ( variación del diámetro del núcleo, presencia de contaminantes, etc. ), y la reflexión de Fresnel en la frontera entre el vidrio y el aire que ocurre en ambos extremos de la fibra.



## Pérdidas por curvatura

El valor de apertura numérica de una fibra óptica corresponde al que se obtiene suponiendo que la misma se mantiene derecha. En el dibujo siguiente puede verse como se produce un efecto de reducción de la apertura numérica al curvar la fibra.



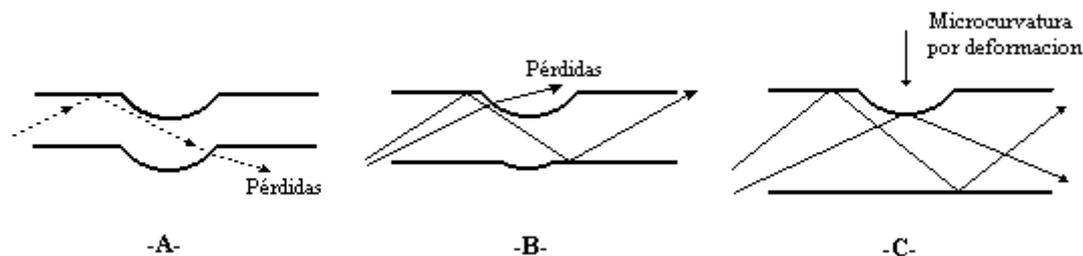
En condiciones normales, tanto el rayo 1 como el 2 se propagan por el núcleo. Sin embargo, cuando la fibra se tuerce, en esa zona el rayo 1 sufre una refracción hacia el revestimiento, ( es decir que parte de la energía del mismo se pierde y no continua su camino por la fibra ). Es evidente que la magnitud de la disminución de la NA, se corresponderá con el radio de la curvatura de la fibra. También el efecto puede entenderse como una anulación de los modos de propagación de orden elevado en fibras multimodo, de lo que también puede deducirse que a medida que el diámetro del núcleo es menor, el efecto de la curvatura se va reduciendo. Desde ambos puntos de vista, lo que se reduce es la capacidad de la fibra para acoplar luz del dispositivo que se use como emisor.

Si la fibra es monomodo, la curvatura casi no produce variación en la apertura numérica ( que siempre es muy baja ) sino que finalmente implica una disminución del ancho de banda de la misma.

En realidad, existe un valor mínimo para el radio de la curva, por encima del cual, las pérdidas son prácticamente despreciable, y por lo tanto ignoradas. Este radio de curvatura mínimo depende, entre otras cosas, principalmente del diámetro del núcleo y al respecto se puede decir que en general las fibras pueden ser mas curvadas a medida que el núcleo es más pequeño.

Sin embargo, corresponde también diferenciar entre las curvaturas grandes ( que son las que normalmente deben producirse por requerimientos de la instalación ) y las "microcurvaturas", que se producen por ejemplo, por un aplastamiento o deformación de la fibra, lo cual puede ser un defecto muy común al efectuar el montaje de conectores ya que los mismos se sujetan a la fibra mediante presión (en términos corrientes se "crimpean"), por ello se debe prestar especial atención a las herramientas utilizadas.

Estas microcurvas pueden afectar la propagación de manera distinta dependiendo de la forma de la misma, y del diámetro del núcleo de la fibra.



En la figura A se muestra el efecto de una microcurvatura en una fibra de pequeño diámetro donde la deformación es igual en toda la sección transversal y puede llegar el caso en que la pérdida sea total. La figura B muestra el caso de una fibra algo más gruesa que la anterior en la cual la deformación no es igual; algunos modos resultan atenuados, pero otros pueden propagarse igualmente. El caso C muestra un caso donde una fibra de sección relativamente grande comparada con las anteriores sufre una alteración mínima ya que la deformación solo está presente en uno de los lados.

### Pérdidas por acoplamiento

Cuando una fibra óptica se acopla a un emisor de luz, un detector de luz u otra fibra, la unión entre la fibra y la otra entidad no es perfecta, por lo que ocurre una pérdida en la frontera. Cualquier energía de luz que entre en la fibra fuera del cono de aceptación se pierde a través del revestimiento. Como luego se verá es posible definir valores de apertura normal también para fuentes emisoras y detectores de luz. Cuando una fuente de luz tiene un cono de aceptación estrecho respecto del que tiene la fibra receptora, no existe pérdida alguna, siempre que la energía luminosa no esté fuera de dicho cono. En cambio si una fibra fuente tiene un cono de aceptación amplio respecto de un detector o de una fibra receptora si hay pérdidas, y estas se calculan de la siguiente manera:

$$\text{Pérdidas} = 20 \log \text{ NA}_2/\text{NA}_1$$

( Donde  $\text{NA}_1$  es el valor de apertura numérica más grande y  $\text{NA}_2$  el valor más chico ).

De lo expresado, puede deducirse que en general no es una buena idea intentar acoplar o empalmar fibras ópticas de distinto tamaño, y aunque este fuera el mismo, aun así habría que comprobar si los valores de apertura numérica coinciden ( lo cual puede no ser así tratándose de distintos fabricantes ).

### Pérdidas intrínsecas

Cuando la luz atraviesa un medio transparente como el vidrio, la perturbación ondulatoria hace que los electrones de las moléculas, oscilen periódicamente en respuesta al vector campo eléctrico. Es evidente que durante este proceso, parte de la potencia contenida en la onda luminosa, es cedida a la molécula, y transformada en otro tipo de energía ( por ejemplo calor ).



Visto desde el exterior, en el vidrio se produce una "absorción" de energía, la magnitud de la cual guarda cierta relación con la frecuencia de resonancia molecular y con la longitud de onda de la luz considerada. Para el vidrio de sílice puro, la absorción responde aproximadamente a la siguiente expresión:

$$\text{Absorción} \propto C \cdot 10^{1/\lambda}$$

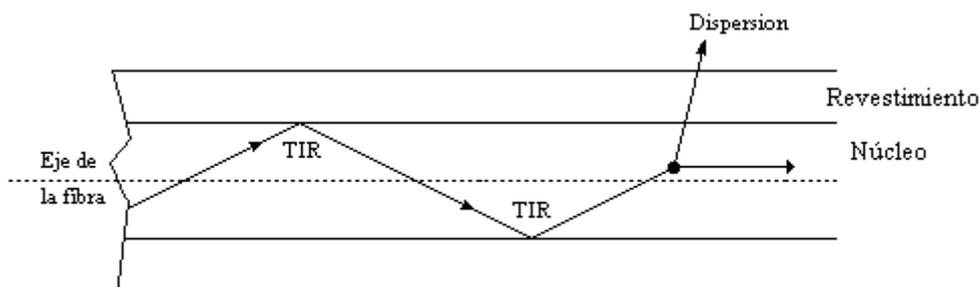
Donde  $C$  es una constante y  $\lambda$  es la longitud de onda de la luz considerada.

En el vidrio, hay además distintas impurezas o contaminantes, que colaboran en aumentar la absorción. Por ello la curva de absorción en función de la longitud de onda no tiene un desarrollo suave sino que presenta irregularidades que determinan ciertas bandas de absorción selectivas, entre las cuales pueden citarse:

- Banda del Ultra violeta.
- Banda del Infra rojo.
- Bandas del radical  $\text{OH}^-$
- Bandas de iones metálicos.

De todas estas, las bandas de absorción más importantes son las que se producen por la presencia de los radicales  $\text{OH}^-$ , que provienen del agua, y que se difunden a través del vidrio durante el proceso de fabricación, y a medida que la fibra envejece.

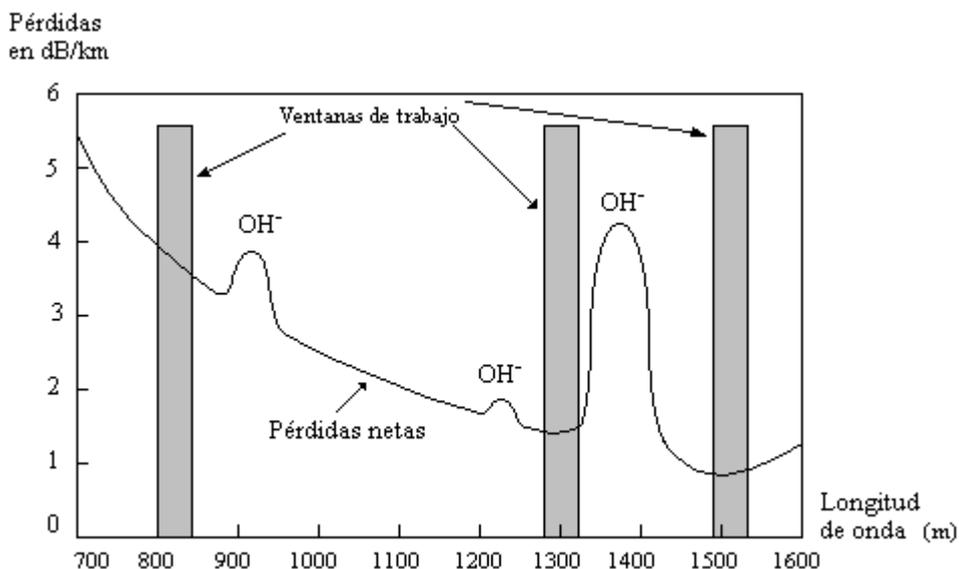
Otro mecanismo por el cual se produce atenuación es la dispersión de Rayleigh que se debe a defectos tales como microburbujas, concentraciones importantes de impurezas, o variaciones locales de la densidad del vidrio. Por ello los rayos de luz que encuentran en su trayectoria estos defectos, pueden cambiar de dirección, y perderse hacia el exterior a través del revestimiento de la fibra.



Lógicamente, la absorción también depende de la longitud de la fibra considerada, por ello normalmente los fabricantes de fibras ópticas acostumbran a especificar las pérdidas por absorción de las mismas en decibelios por unidad de longitud (habitualmente la unidad de longitud es el km.).



El gráfico siguiente muestra como varia la atenuación neta de una fibra óptica de vidrio de sílice típica.



Pérdidas por absorción de una fibra óptica de vidrio.

La figura anterior sugiere también la existencia de "ventanas" de trabajo (zonas grises) en las cuales es más oportuno operar. Todas ellas están en la región invisible del espectro (infrarrojo). Para transmisiones a larga distancia se utilizan las ventanas de 1300 o 1500 nm. La ventana de 800 nm (la más próxima a la luz visible) se usa en instalaciones de distancias más cortas y de costos más bajos.

### Reflexión de Fresnel

La reflexión de Fresnel ocurre siempre en cualquier frontera entre dos medios cuyos índices de refracción son distintos. Cuando la luz pasa de un medio al otro, parte de la energía del rayo incidente es reflejada de nuevo hacia el primer medio. La reflexión de Fresnel se pone en evidencia particularmente en cualquiera de los extremos de una fibra óptica o en todas las partes de la misma donde exista un conector, ya que indefectiblemente se forma una frontera entre el vidrio del núcleo y el aire. Por ello sucede que al intentar acoplar un emisor a una fibra, parte de la energía de la luz es devuelta hacia el aire, y en el caso del acoplamiento con el detector parte de la energía que viaja por la fibra, es reflejada de nuevo hacia la fuente. La cantidad de luz que se refleja en la frontera depende principalmente de la diferencia entre los índices de refracción de los dos medios, y viene dada en forma aproximada por la siguiente expresión:

$$\text{Luz reflejada (\%)} = 100 \cdot (n_1 - n_2)^2 / (n_1 + n_2)^2$$

Donde  $n_1$ ,  $n_2$  son, respectivamente, los índices de refracción de cada medio.



Hay que tener en cuenta que en el caso del aire, su índice de refracción puede sufrir variaciones de acuerdo con la temperatura, el porcentaje de humedad, y la presencia de partículas en suspensión. A veces, cuando la terminación de la fibra no está suficientemente protegida de los agentes externos, pueden depositarse sobre la misma, partículas de polvo o se puede condensar la humedad. Todos estos factores van a influir en el porcentaje de luz reflejada. De ahí que es muy importante la limpieza y protección de los distintos tipos de conectores que se utilizan. Para efectuar la correspondiente limpieza, se deben usar las técnicas y materiales apropiados, (que se estudiarán más adelante).

Resumen:

- Las pérdidas por curvatura afectan en mayor magnitud a las fibras a medida que mayor es el diámetro del núcleo.
- Las pérdidas por microcurvaturas afectan en mayor medida a las fibras monomodo
- La regla práctica es : Cuidado con las macrocurvaturas al efectuar el tendido de cables de fibras ópticas. Cuidado con las microcurvaturas al crimpear los conectores.
- Para el empalme de fibras ópticas se requiere que las mismas correspondan tanto en tamaño como en valores de apertura numérica.
- La limpieza de conectores, emisores y detectores es de fundamental importancia y es un aspecto que no debe tomarse a la ligera.

## DISPERSIÓN Y ANCHO DE BANDA DE LAS FIBRAS ÓPTICAS

Un pulso de luz que se acopla a la punta de una fibra óptica, llega al otro extremo de la misma “desparramado” o “ensanchado” en el tiempo, esto es debido al efecto conocido como Dispersión de las fibras ópticas.

La deformación de un pulso rectangular que viaja por una fibra ocurre como resultado de uno o más de las siguientes formas o maneras de dispersión.

- Dispersión modal.
- Dispersión material.
- Dispersión de guía de onda.

La velocidad con la cual puede transmitirse una serie de pulsos digitales (lo que en inglés se conoce como “bit rate”) por una F.O. dependerá en forma muy directa de la dispersión que se produzca, por lo cual resulta importante estudiar cuáles son los mecanismos por los cuales se produce la misma.

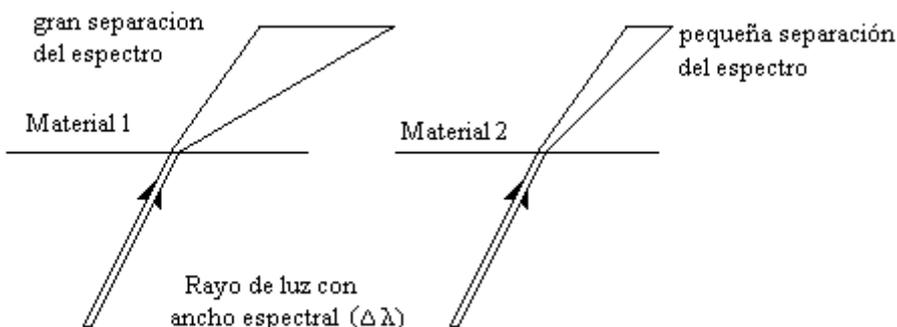
### Dispersión modal

Se presenta principalmente en las fibras multimodo, y ocurre porque que los rayos de luz que parten a la vez de un emisor no llegan, sin embargo, al mismo tiempo al extremo final de la fibra, debido a que recorren diferentes caminos, (siendo más largos los correspondientes a los modos de orden más elevados). El efecto es muy notable en las fibras multimodo de índice escalonado, pero afecta bastante menos a las fibras multimodo de índice gradual. En cuanto a las fibras monomodo, en principio podría decirse que no hay dispersión de este tipo (Luego se verá que en realidad también hay dispersión modal, pero como resultado de otro fenómeno).



## Dispersión material

Es un efecto conocido que el índice de refracción de cada material varía con la longitud de onda considerada, por eso es posible separar las componentes espectrales de la luz blanca del sol mediante un prisma. La magnitud de la variación del índice de refracción en función de la longitud de onda es una propiedad característica de las moléculas de cada material.



En el dibujo precedente se representan los distintos resultados que producen dos materiales distintos sobre la separación de las componentes espectrales de un rayo de luz. Este efecto depende, como ya se ha dicho, de las características de los materiales, pero es evidente que también será consecuencia del ancho espectral del rayo de luz incidente.

Las fuentes ópticas utilizadas en los sistemas de F.O. emiten luz no solamente en una longitud de onda específica, sino que también lo hacen en un pequeño rango alrededor de una determinada longitud de onda central. Este rango de longitudes de onda inyectadas a una fibra se denomina “ancho espectral” de la fuente de luz.

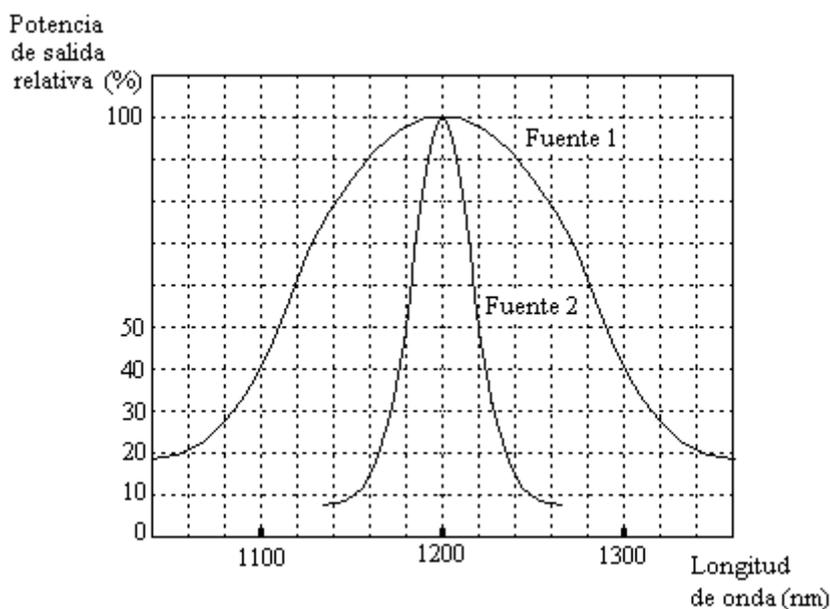
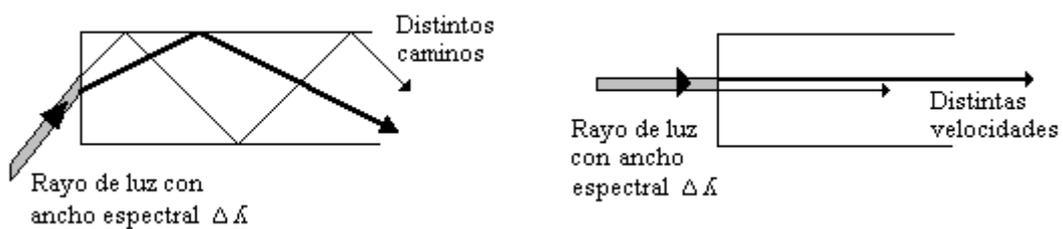


Figura 2 ( Gráfica típica de distribución espectral de dos fuentes ópticas distintas)



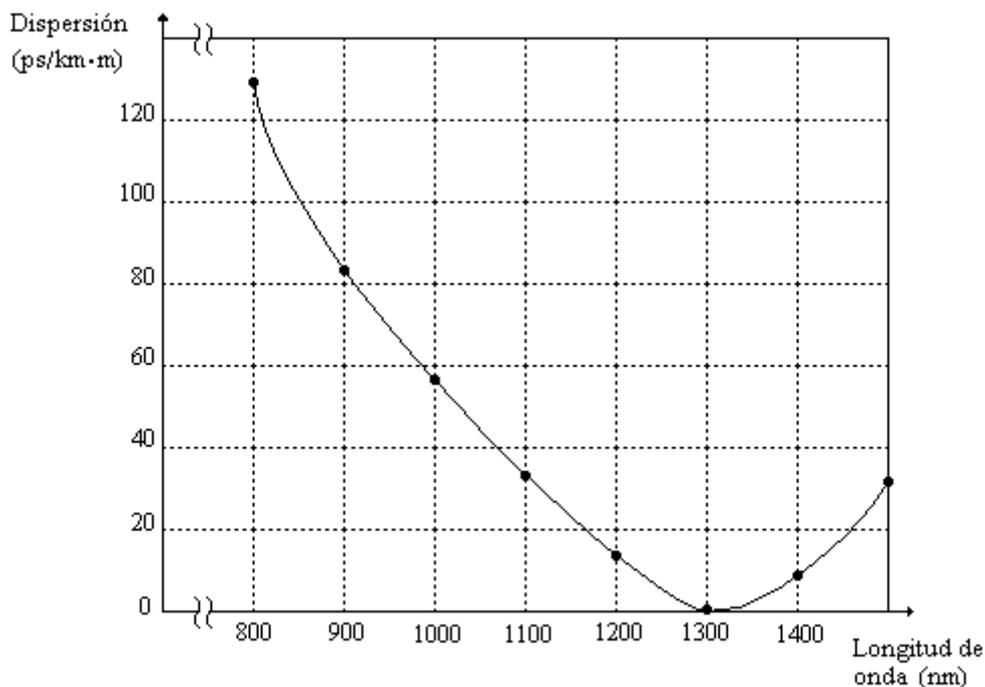
Dado que el índice de refracción del núcleo es distinto para cada una de las componentes de un pulso de luz emitido por una fuente, se originará un ensanchamiento del pulso transmitido a causa de los dos siguientes efectos:

- Las distintas componentes espectrales de la fuente, tomarán distintos caminos en razón de, la variación del índice de refracción en función de la longitud de onda distinta con un efecto similar al de la dispersión modal.
- Las distintas componentes viajarán además a velocidades diferentes, y como consecuencia de ello llegarán a destino en instantes distintos.



Cualquiera de estos dos efectos se verá incrementado a medida que mayor es el ancho espectral de la fuente. Por otra parte es evidente que, ya sea en el caso de la dispersión modal, como en la dispersión material, la magnitud del efecto dependerá también directamente de la longitud física de la fibra. Por esto los fabricantes especifican la dispersión de cada tipo de fibra en forma de un coeficiente normalizado, es decir con referencia a la unidad de longitud. Por razones que se explican un poco más adelante, en las fibras ópticas monomodo este parámetro se da en ps/km . nm . Así, por ejemplo, un valor tal como 300 ps/km.nm significa que un pulso transmitido sufrirá un ensanchamiento de 300 ps por cada nm de ancho espectral de la fuente usada luego de recorrer una distancia de 1 km por la fibra .

Afortunadamente para el vidrio de sílice que se usa en las fibras ópticas, la variación relativa del índice de refracción respecto de la longitud de onda no es constante, por lo cual la dispersión material puede reducirse e incluso anularse si se elige convenientemente la longitud de onda de trabajo, el gráfico siguiente muestra la dispersión típica para una fibra monomodo como función de la longitud de onda.



El agregado de ciertas impurezas (dopado) permite obtener variantes de vidrios que posibilitan la fabricación de fibras monomodo con baja dispersión en un margen comprendido entre 1300nm y 1550nm ( se conocen como fibras “dispersion flattened”).

### Dispersión de guía de onda

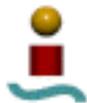
Ocurre principalmente en las fibras monomodo, y sobre todo cuando el diámetro del núcleo es sumamente pequeño ( y por lo tanto comparable a la longitud de onda de la luz). En estas circunstancias no puede evitarse que parte de la energía de la luz se desborde del núcleo y viaje por el revestimiento (lo que se conoce como efecto túnel). Como además, los índices de refracción del núcleo y del revestimiento son de valores distintos (aunque muy próximos) hay una diferencia en las velocidades que finalmente es causa de un efecto de dispersión. La dispersión de guía de onda es comparativamente bastante menor que las dispersiones modal y material, y solo se vuelve importante cuando estas últimas tienden a cero.

En todas las fibras ópticas la dispersión total de las mismas será el resultado de la superposición de los efectos producidos por cada uno de los mecanismos estudiados. Por ejemplo si en una determinado tipo de fibra, hay dispersión modal y dispersión material, el ensanchamiento total será:

$$\Delta t(total) = \sqrt{\Delta t(modal)^2 + \Delta t(material)^2}$$

Sin embargo en cada tipo de fibra hay una forma dominante que prevalece por sobre las demás. La dispersión modal afecta grandemente a las fibras multimodo de índice escalonado, y un poco menos a las de índice gradual, donde comienza a ser importante la dispersión material (también llamada dispersión cromática).

Esta ultima a su vez afecta a las fibras monomodo pero puede reducirse eligiendo apropiadamente la longitud de onda de trabajo y utilizando emisores ópticos altamente coherentes (con reducido ancho espectral), a partir de este punto comienza a tornarse importante la dispersión de guía de onda.



## Ancho de banda para una fibra óptica

Hemos aprendido que la dispersión es uno de los principales factores que condicionan la transmisión por una fibra óptica, ya que el ensanchamiento en el tiempo ( $\Delta t$ ), de los pulsos que forman el tipo de señal digital que habitualmente se transporta por un sistema de FO impone un límite a la separación mínima que debe haber entre dos pulsos sucesivos.

El ancho de banda del sistema estará en relación directa con la capacidad del mismo para poder distinguir entre dos pulsos seguidos. Como la información concerniente al cambio de estado de una señal digital, esta en cada uno de los flancos de la misma, si se considera un hipotético pulso cuya duración es cero, se requerirá al menos el doble del tiempo de ensanchamiento  $\Delta t$  para transmitir la información completa (se invierte un tiempo para el flanco de subida, y otro idéntico para el de bajada), de manera que el ancho de banda máximo tolerable del sistema será aproximadamente:

$$AB \approx \frac{1}{2 \cdot \Delta t} = \frac{0,5}{\Delta t}$$

La ecuación anterior sirve para la estimación del ancho de banda en fibras ópticas multimodo, y también podría aplicarse para las fibras monomodo (donde prácticamente no hay dispersión modal) si el espectro del emisor óptico utilizado fuese parejo (la misma amplitud para todas las componentes).

En cambio si la fuente usada tiene una distribución gaussiana (lo cuales muy común para emisores tipo LED), el ancho de banda debe calcularse a partir de la siguiente ecuación:

$$AB = 0,187 / D \cdot L \cdot \Delta\lambda$$

Donde  $D$  es la dispersión,  $L$  es la longitud de la fibra, y  $\Delta\lambda$  es el ancho espectral de la fuente evaluado entre los puntos de potencia mitad.

Por todo lo expuesto, se puede comprender ahora porque los fabricantes especifican de manera distinta la dispersión según se trate de fibras multimodo o monomodo.

Para fibras multimodo, donde la dispersión dominante es modal, la dispersión dependerá casi exclusivamente de los parámetros propios de la fibra, y se acostumbra a dar como especificación el "producto Distancia - Ancho de banda". Así, un valor para este parámetro de 500 Mhz· km significa que:

- Una señal de 500 Mhz. puede ser transmitida a 1 km
- Una señal de 250 Mhz. puede ser transmitida a 2 km
- Una señal de 1000 Mhz. puede ser transmitida a 0,5 km

En cambio para las fibras monomodo, donde la dispersión depende (además de los parámetros propios de la fibra) del ancho espectral de la fuente óptica, se la normaliza también respecto de la unidad de longitud de onda, tal como se ha ejemplificado previamente.



Tabla de especificaciones (para algunos tipos de fibras)

Tipo	Diámetro del núcleo (μm)	Diámetro del revestimiento (μm)	Apertura Normal	Ancho de banda (Mhz . km)	Atenuación (dB/km)
Monomodo	8	125		6 ps/km.nm (*)	0,5 - 1300 nm
	5	125		4 ps/km.nm (*)	0,4 - 1300 nm
Multimodo de índice gradual	50	125	0,20	400	4 - 850 nm
	63	125	0,29	250	7 - 850 nm
	85	125	0,26	200	6 - 850 nm
	100	125	0,30	20	5 - 850 nm
Multimodo de índice escalón	200	380	0,27	25	6 - 850 nm
	300	440	0,27	20	6 - 850 nm
Multimodo de plástico	200	350	0,30	20	10 - 790 nm
	400	550	0,30	15	10 - 790 nm 400 - 650 nm

(\*) Dispersión para fibras monomodo.

## GENERALIDADES SOBRE LOS EMISORES Y DETECTORES ÓPTICOS

Los emisores y los detectores ópticos, son respectivamente los principales componentes de los transmisores y receptores usados en los sistemas de transmisión por F.O. Las propiedades optoelectronicas (como transductores) y ópticas propiamente dichas de estos elementos son las que en definitiva determinan en última instancia el comportamiento de un sistema de comunicaciones por fibras ópticas.

Las características que deben poseer los emisores y detectores están definidas por las especificaciones del sistema, es decir, ante todo por la longitud de los enlaces, y por la velocidad de transmisión requeridas.

En particular esto significa para los emisores que:

- La emisión de los impulsos luminosos, tiene que tener lugar en el margen espectral para el cual la fibra óptica adoptada, presente la mínima absorción y/o dispersión.
- La potencia radiada acoplada a la fibra óptica debe ser la mayor posible. Esto significa no solamente un rendimiento de transducción optoelectrica elevado sino también un reducido valor de pérdidas por acoplamiento.
- La emisión óptica debe ser modulable en forma sencilla por la señal transmitida.



Para los detectores se pueden enumerar también de manera análoga:

- La sensibilidad de recepción debe ser lo mayor posible manteniendo las mejores condiciones de ruido. De esta forma, incluso con una frecuencia de error binaria predeterminada es detectable todavía una potencia óptica mínima.
- Para la velocidad de transmisión deseada, la velocidad umbral tiene que ser lo suficientemente grande.

## EMISORES ÓPTICOS USADOS EN SISTEMAS DE F.O.

### Diodos emisores de luz (LEDs)

Un LED es un dispositivo formado por una juntura semiconductor PN, que emite luz por el mecanismo de emisión espontánea. Existen en la actualidad, dos tipos de estructuras básicas de LEDs que se adecuan a los requerimientos de los sistemas de FO. 1) Las estructuras en Homojunturas, y 2) Las estructuras en Heterojunturas. Los LEDs de homojuntura (también conocidos como de emisión superficial) están formados por un material semiconductor simple, en tanto que los de heterojuntura (conocidos como de emisión lateral) usan más de un material, lo cual permite un mejor confinamiento de la luz emitida además de un mejor comportamiento frente a las variaciones de temperatura.

Los materiales usados en la construcción de los LEDs tiene influencia en varios de los principales parámetros de interés, que son:

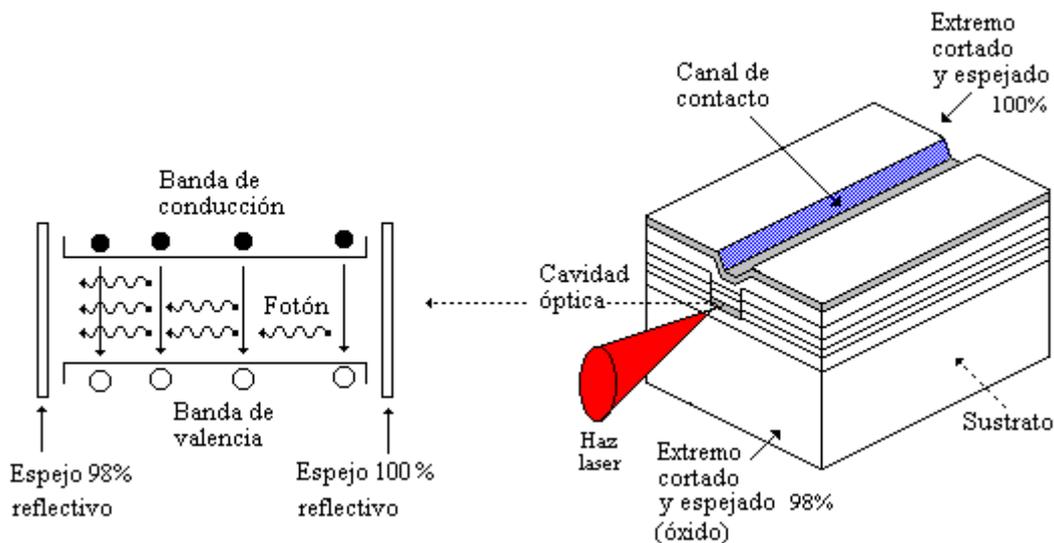
- La longitud de onda de la luz emitida ( $\lambda$ ). Los LEDs que emiten en longitudes de onda comprendidas entre 820 y 850 nm se fabrican mediante el dopado del arseniuro de galio con aluminio (Ga Al As), en tanto que para conseguir longitudes de onda de 1300nm se usan como impurezas el indio y el fósforo (Ga As In P).
- El rendimiento de potencia externo, es decir la relación entre la potencia de la luz emitida externamente, y la potencia eléctrica consumida, que por lo general es sensiblemente menor que 1. Para dar una idea, se pueden obtener valores absolutos de potencias de luz emitidas del orden de -10 dBm con una relación lineal entre corriente y potencia o incluso magnitudes más elevadas como 0 dBm, aunque con un comportamiento muy alineal.
- El ancho espectral ( $\Delta\lambda$ ), el cual es aproximadamente proporcional al cuadrado de la longitud de onda, es decir que se incrementa sensiblemente para mayores longitudes de onda, lo cual tiene gran importancia en relación con los fenómenos de dispersión. (Un orden de magnitud típica puede ser: para  $\lambda \approx 800\text{nm}$ , un  $\Delta\lambda = 100\text{ nm}$ ).
- La frecuencia límite de modulación, que reviste especial importancia en relación con las telecomunicaciones. La vida media de los portadores de carga en exceso (es decir el tiempo que los electrones permanecen en la banda de conducción antes de caer espontáneamente a la banda de valencia) fija un límite por debajo del cual la emisión de luz no sigue las variaciones de la corriente que atraviesa la juntura. Dichos valores de vida media están situados por lo general alrededor de algunos nanosegundos, lo que posibilita frecuencias límite de modulación de hasta algunos cientos de MHz.



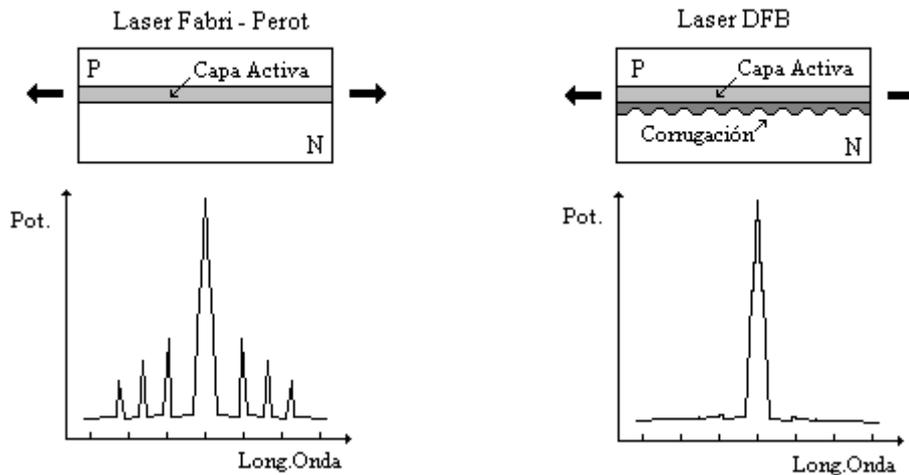
## Diodos Láser de inyección. (ILD)

Al igual que los diodos LED, un diodo Láser esta formado por una juntura semiconductor, por lo cual son validas las mismas consideraciones efectuadas respecto de los parámetros de interés, pero a diferencia de aquellos tiene asociado un elemento selectivo de longitud de onda (una cavidad resonante) que hace que la luz se genere principalmente por emisión estimulada.

En el tipo mas sencillo de diodo Láser el resonador óptico, denominado de Fabry - Perot, consiste en un pequeño bloque compuesto por varias capas del semiconductor (que forman la juntura). Este chip posee una hendidura y se encuentra entre dos superficies paralelas y reflectoras, una de ellas levemente translúcida. El conjunto se comporta como una cavidad óptica y si las dimensiones de la misma son las adecuadas, la emisión espontanea de un fotón estimula a su vez la emisión de otros, de igual fase y longitud de onda, que se reflejan sucesivamente en los espejos aumentando aun mas el efecto inicial. Parte de los fotones generados pueden salir hacia el exterior a través del espejo translucido en forma de un haz de luz altamente direccional, coherente y de gran intensidad. Para que este efecto se produzca, es necesario un valor mínimo de corriente de inyección (umbral de disparo), a través de la juntura. La magnitud de la corriente esta en el orden de los 100 mA y es bastante dependiente de la temperatura. El orden de magnitud de la potencia óptica de salida es de alrededor de 10 dBm.



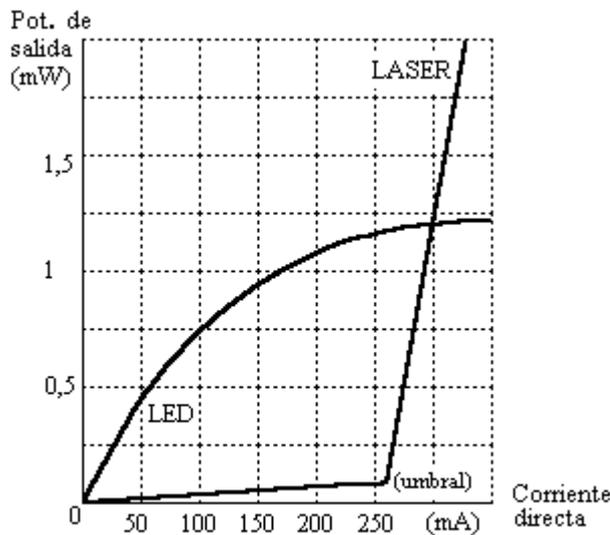
Aunque el ancho espectral de los Láser Fabry -Perot es bastante angosto, lo es aun más en los diodos Láser de Realimentacion Distribuida (DFB Láser), en los que se consigue una emisión en una longitud de onda prácticamente única merced a una forma especial de la hendidura, una de cuyas caras es corrugada, y a la mejora en el proceso de espejado de los extremos de la cavidad.



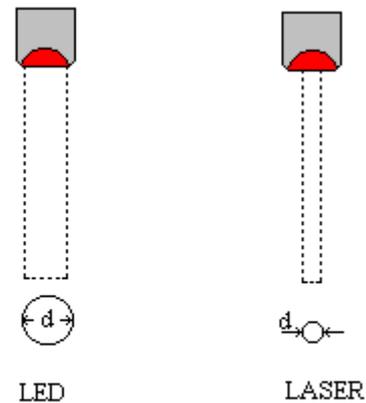
### Características de las fuentes

La potencia óptica de salida de los diodos LEDs y Láser es proporcional a la corriente aplicada a la juntura. En general, la potencia de salida de los Láser es comparativamente mayor que la de los LEDs, salvo cuando en los primeros se opera con corrientes de inyección menores que la de disparo.

El diámetro del haz así como la forma del lóbulo de la luz emitida por una fuente óptica es de fundamental importancia para la técnica de las fibras ópticas, ya que el mismo estará directamente ligado con la apertura numérica (concepto estudiado en lecciones previas). Una buena fuente óptica para los propósitos de un sistema de FO, es aquella que exhibe un buen rendimiento de potencia externo aunado a valores reducidos de apertura numérica y diámetro de emisión.



(A) Curva de corriente vs. potencia de salida



(B) Diametro del haz



Al igual que la diferencia entre los valores de apertura numérica de dos componentes que se acoplan produce pérdidas, la diferencia entre el diámetro del haz emitido por una fuente, y el diámetro del núcleo de una fibra que se acoplan disminuye la cantidad de luz que puede ser transferida del emisor a la fibra en una relación que viene dada por la siguiente expresión.

$$\text{Pérdidas (por diferencia de diámetro)} = 20 \log D_f/D_e$$

Donde  $D_f$  es el diámetro del haz emitido por la fuente, y  $D_e$  es el diámetro del núcleo de la fibra.

Habitualmente, los LEDs que son más fáciles de usar y que además presentan una zona de trabajo lineal donde a pesar de los bajos valores absolutos de potencia emitida, comparativamente tienen más rendimiento que los Lasers, no se usan como fuentes ópticas para fibras monomodo a causa de los elevados valores de apertura normal y diámetro de emisión. En cambio los diodos Láser, que en principio pueden usarse con cualquier tipo de fibra óptica son más alineales, su comportamiento varía con la temperatura, y son más difíciles de usar, se prefieren para sistemas con fibras monomodo, (es decir donde hay requerimientos simultáneos de las largas distancias con velocidades elevadas que no pueden ser satisfechos con fibras multimodos).

Los emisores Laser poseen anchos espectrales sensiblemente menores que los LEDs, por ejemplo para valores comparables de potencia emitida si el  $\Delta\lambda$  de un diodo LED es 100 nm, el correspondiente valor para un Laser Fabry - Perot puede ser 10 nm, e incluso bastante menor (p.ej 1 nm) para un Láser DFB.

Respecto de la frecuencia límite de modulación de un diodo LED o un Láser, la misma estará en relación con los tiempos de crecimiento o de caída (se toma el mayor de los dos) de cada dispositivo. Una regla aproximada para el cálculo de esa relación es:

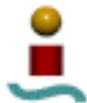
$$AB = 0,35/tr$$

Otro aspecto importante a tener en cuenta, es el "Tiempo de vida" de las fuentes que está definido como el tiempo que transcurre desde que el dispositivo se usa por primera vez hasta que la potencia emitida baja al 50% del valor original. Por ejemplo si un LED emite una potencia óptica de 1 mW al utilizarlo inicialmente, su tiempo de vida habrá llegado al final cuando la potencia emitida sea solamente 0,5W.

La forma de la cápsula que contiene al emisor, así como las técnicas de montaje, son otro de los aspectos importantes a tener en cuenta, ya que tendrán una directa influencia en la eficiencia con que la luz se acopla a la fibra.

Resumiendo:

- En un sistema de transmisión por fibras ópticas, hay tres componentes principales del mismo cuyas características definen el comportamiento del sistema, ellos son: Emisores - Fibras - Detectores. Pero en realidad para el estado actual de la técnica las limitaciones de un sistema pasan principalmente por las limitaciones de los emisores y de los receptores.
- Hay dos tipos de dispositivos emisores principales, con sus correspondientes variantes, ellos son los diodos LED, y los diodos LASER.



### COMPARACIÓN ENTRE DIODOS LED Y LASER.

Características	LED	Diodos LASER
Potencia de salida	Baja	Elevada
Velocidad	Lentos	Rápidos
Patrón de salida (AN)	Grande	Reducido
Ancho espectral	Amplio	Angosto
Compatibilidad con FO monomodo	No	Si
Dependencia de la temperatura	Baja	Elevada
Aplicación	Fácil	Difícil
Costo	Bajo	Elevado

### DETECTORES ÓPTICOS USADOS EN SISTEMAS DE F.O.

Las señales ópticas que se transmiten por una fibra, deben convertirse nuevamente en señales eléctricas en el equipo receptor. Aunque existen varios tipos de dispositivos que podrían usarse en principio como transductores "optoelectronicos" (por ejemplo: las fotoceldas resistivas, las celdas fotovoltaicas, y los fototransistores), en los sistemas de comunicaciones por FO, se utilizan principalmente los fotodiodos.

En el estado actual de la técnica, existen tres tipos de fotodiodos cuyo uso está difundido en los sistemas de FO. Estos tipos son: 1) Los fotodiodos PN. 2) Los fotodiodos PIN. 3) Los fotodiodos de avalancha.

#### Fotodiodos PN

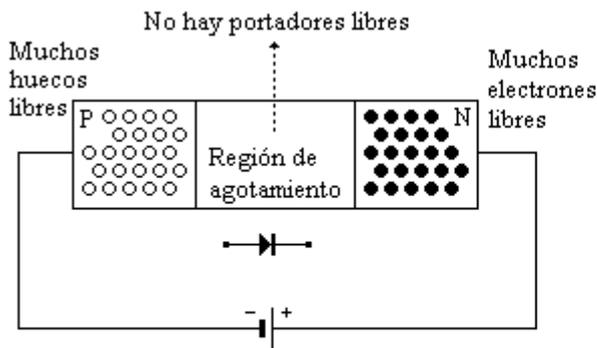
El más simple de los dispositivos optoelectronicos de detección es el fotodiodo de juntura PN cuyo principio de funcionamiento se ajusta casi exactamente a lo descrito en el párrafo previo. Podría decirse, en pocas palabras, que el diodo se polariza inversamente y que la corriente inversa, que normalmente es muy pequeña, puede controlarse mediante el mecanismo de absorción que se describió en la lección anterior. Dicha corriente inversa varía en forma más o menos proporcional a la cantidad de fotones que inciden en la juntura.

Los fotodiodos PN presentan serias restricciones en lo referente a sus características, lo cual hace que su uso esté algo limitado para aplicaciones en sistemas de FO. Dichas limitaciones se originan en las siguientes causas:

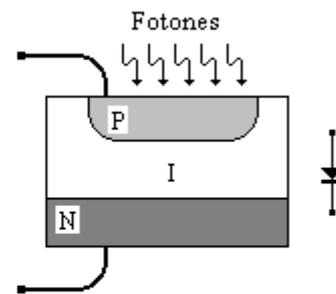
1. Los fotones incidentes liberan electrones de la banda de valencia no solamente en la zona de agotamiento, sino también en regiones fuera de la misma. Si un electrón es liberado fuera de la zona de agotamiento, por ejemplo dentro del material tipo P de la juntura, es más probable que se recombine rápidamente con algún hueco, antes que contribuya a aumentar la corriente inversa. Los fotodiodos PN tienen normalmente, una zona de agotamiento bastante estrecha, y por lo tanto se requiere un haz de luz altamente colimado para crear portadores libres dentro de la misma o en su defecto gran cantidad de fotones. En otras palabras la eficiencia (relación entre el número de fotones y el número de portadores generados) es sumamente baja.



2. Los electrones que saltan a la banda de conducción fuera de la zona de agotamiento, y no llegan a recombinarse, finalmente contribuirán a aumentar la corriente a través de la juntura, pero existirá cierto retardo entre la aniquilación del fotón y el aumento efectivo de la corriente. Esta demora en la respuesta termina produciendo una reducción en la velocidad de operación del fotodiodo.



(A) Juntura polarizada inversamente



(B) Diodo PIN

### Fotodiodos PIN

Los fotodiodos PIN, fueron ideados para superar las dificultades que presenta el fotodiodo PN como detector. La designación de este tipo de diodo se corresponde con las tres capas de semiconductores que se usan para su fabricación: Capa tipo P, Capa de material Intrínseco, y Capa tipo N. (figura B).

Un semiconductor intrínseco (aquel que no ha sido dopado con impurezas) se intercala entre las dos capas P y N del diodo, con lo que se produce el efecto de ensanchar la zona de agotamiento. Con esto se consigue incrementar la eficiencia del diodo como detector al aumentar la relación entre el número de fotones y el número de portadores generados. Sin embargo, un aumento excesivo del tamaño de la zona de agotamiento tendrá un efecto pernicioso sobre la velocidad de respuesta del dispositivo.

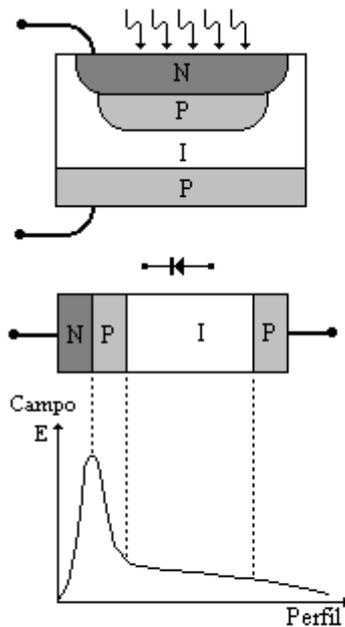
Los Fotodiodos PIN se utilizan ampliamente en los sistemas de FO, y están diseñados específicamente para estas aplicaciones. El espesor de la capa intrínseca se fija buscando, en cada caso, una relación de compromiso entre la eficiencia, y la velocidad de respuesta necesarias.

### Fotodiodos de avalancha. (APD)

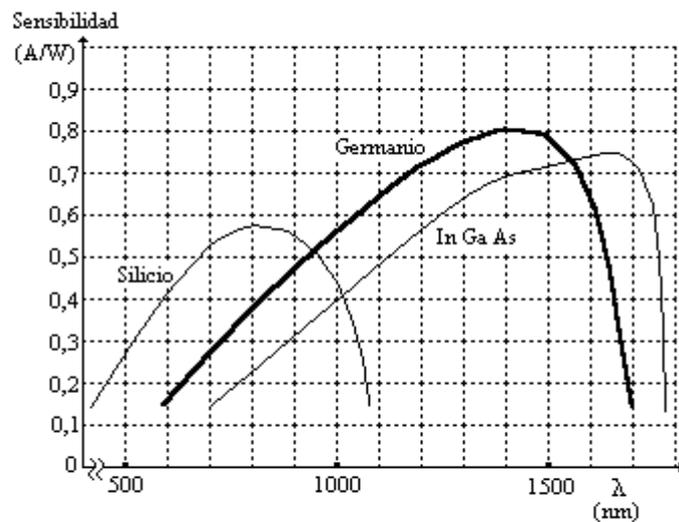
Los fotodiodos de avalancha presentan una elevada eficiencia en la relación de conversión entre fotones y portadores debido a que su operación se basa en el fenómeno de "foto multiplicación". Los diodos APD se fabrican con una estructura que genera un campo eléctrico muy fuerte en una porción de la región de agotamiento (figura C).



Los portadores que se crean por la incidencia de fotones (llamados portadores primarios) son acelerados por este campo y colisionan con los electrones de los átomos de la región intrínseca. La energía cinética que acumulan los electrones al acelerarse debido al campo eléctrico, es cedida a los electrones de la banda de valencia de los átomos neutros causando su salto a la banda de conducción ( estos son los portadores secundarios), lo cual incrementa la corriente a través del diodo. En pocas palabras, un único fotón incidente puede generar varios portadores, que a su vez aumentan la corriente. Por este motivo, un diodo APD es bastante mas eficiente que un diodo PIN. El numero de electrones que se transforman en portadores depende del factor de multiplicación del fotodiodo. Un factor de multiplicación de 50 significa que por cada fotón incidente se generan 50 electrones de conducción.



(C) Estructura básica de un diodo de avalancha



(D) Sensibilidad (Responsivity)

## Características de los Detectores

Las principales características de los fotodiodos dependen tanto del tipo de dispositivo que se trate como del de semiconductor que se usa para su fabricación, y en menor medida de las características de los circuitos asociados. También son importantes, desde luego, las características que tiene que ver con el encapsulado o montaje de los detectores para su utilización y aplicación.

### Sensibilidad espectral ( $S_\lambda$ )

La sensibilidad espectral de un fotodiodo es una de sus características mas importante. Viene definida por la relación entre la corriente que se induce en el detector y la potencia de la luz incidente, y se mide en A/W.



Por ejemplo, una sensibilidad de 1,2 A/W , significa que un rayo de luz incidente cuya potencia óptica es 1 W producirá una corriente de 1,2 A. La sensibilidad espectral de un fotodiodo depende del material empleado para su fabricación y cambia con la longitud de onda de la luz incidente. En la figura (D) pueden observarse tres gráficas distintas de sensibilidad en función de la longitud de onda para tres tipos de semiconductores. Allí puede notarse que los fotodiodos de silicio no son apropiados para trabajar con longitudes de ondas elevadas. Como se ha estudiado previamente, en las fibras ópticas de vidrio, la atenuación de material es mínima alrededor de 1500 nm, en tanto que la dispersión material se reduce prácticamente a cero para 1300 nm. Para estos valores de longitud de onda los fotodiodos fabricados con germanio o con arseniuro de galio, son mas convenientes.

### **Eficiencia cuántica ( $\eta$ ).**

En los párrafos previos se ha hecho referencia a la relación entre fotones incidentes y portadores generados, esta relación es justamente la "eficiencia cuántica" del fotodiodo, que puede darse en términos de relación entre la cantidad de fotones que pueden crear un par electrón -hueco y los que no producen tal efecto. Su expresión es simplemente:

$$\eta = N^{\circ} \text{ de fotones que crean un par electrón - hueco} / N^{\circ} \text{ de fotones incidentes}$$

Una eficiencia cuántica de 1 (o 100 %) significa que cada fotón que logra incidir sobre el diodo, generara un portador. Una eficiencia cuántica de 0,5 significa que solo la mitad de los fotones que inciden generan un portador cada uno. La eficiencia cuántica de un fotodiodo esta relacionada con la Sensibilidad espectral ( $S_{\lambda}$ ) y también puede expresarse así:

$$\eta = (S_{\lambda}) \cdot h \cdot c / q \lambda \quad (1)$$

Donde:

- $\eta$  : eficiencia cuántica
- $S_{\lambda}$  : Sensibilidad espectral
- $h$  : Constante de Planck =  $6,63 \cdot 10^{-34}$  J.s
- $c$  : velocidad de la luz =  $3 \cdot 10^8$  m/s
- $q$  : Carga eléctrica de un electrón =  $1,6 \cdot 10^{-19}$  Coulomb
- $\lambda$  : Longitud de onda de los fotones incidentes

Las expresiones anteriores son válidas para cualquiera de los tipos de detectores estudiados, pero en el caso de los fotodiodos de avalancha se debe hacer la salvedad que solo indica la eficiencia en la generación de portadores primarios.

### **Voltaje de polarización (Bias)**

Los fotodiodos necesitan un voltaje de polarización inversa para la creación del campo eléctrico en la zona de agotamiento de la juntura (recordar que todos estos dispositivos se polarizan inversamente) . Dicho voltaje de polarización esta comprendido entre 3 V y 15 V para los fotodiodos PN y PIN, en cambio los fotodiodos de avalancha requieren valores de voltajes bastante mas elevados pudiéndose llegar en algunos casos al orden de los cientos de volts. Esta polarización inversa se aplica a través de una resistencia de carga (RI) cuya función es además transformar las variaciones de corriente en variaciones de voltaje. El valor de la resistencia de carga se fija teniendo en cuenta varios factores, entre los cuales una de los mas importantes es el que tiene en cuenta el ruido térmico que se genera sobre la misma.



## Ruido

El propósito de un detector es producir una corriente eléctrica en respuesta a la excitación generada por los fotones incidentes. Sin embargo siempre existe un pequeño monto de corriente no deseada que se genera aunque no exista luz incidente. Esta corriente indeseada se denomina ruido, y aunque existen muchos tipos de ruido, en los fotodiodos que se usan en los sistemas de F.O. los dos principales tipos son: El ruido de "granalla" (Shot noise), y el ruido térmico (Thermal noise).

El ruido de granalla se produce debido a la naturaleza discontinua de la luz y de los portadores generados por la misma. Debido a ello la corriente inducida sufre pequeñas fluctuaciones de valor impredecible. Además cuando la luz que incide sobre el fotodiodo se corta abruptamente, aun así puede seguir circulando una pequeña corriente debida a los pares electrón - hueco generados previamente que se denomina "corriente de oscuridad" (Dark current,  $I_R$ ). La expresión matemática para el ruido de granalla es:

$$I_{sn} = \sqrt{2 \cdot q \cdot i \cdot B}$$

Donde:

$I_{sn}$  : Corriente de ruido de granalla  
 $q$  : Carga eléctrica del electrón =  $1,6 \cdot 10^{-19}$  Coulomb  
 $i$  : Corriente promedio (Incluye la corriente de señal y la corriente de oscuridad)  
 $B$  : Ancho de banda del receptor (al que esta conectado el detector).

El ruido térmico se debe a las fluctuaciones de la resistencia de carga del detector. Los electrones se mueve debido al campo eléctrico, pero también se agitan debido a la temperatura, y como resultado de este movimiento aleatorio se produce una corriente que se superpone a la corriente de la señal produciendo una distorsión en la misma. La corriente de ruido térmico es:

$$I_{tn} = \frac{4 \cdot K \cdot T \cdot B}{\sqrt{RI}}$$

Donde:

$I_{tn}$  : corriente de ruido térmico.  
 $K$  : Constante de Boltzman =  $1,38 \cdot 10^{-23}$  J/°K  
 $T$  : Temperatura en °K  
 $B$  : Ancho de banda del receptor (al que esta conectado el detector).  
 $RI$  : Resistencia de carga.

## Relación Señal/ Ruido. (SNR)

La relación señal - ruido (signal - noise) de un detector, es una medida de la calidad de la señal que el mismo entrega. Se expresa simplemente como la relación, en decibeles, entre el valor promedio de la potencia de la señal disponible en relación con el promedio de la potencia de ruido generado.

$$SNR = 10 \log (S/N)$$

La relación señal ruido se usa principalmente para caracterizar la bondad de los sistemas analógicos.

---

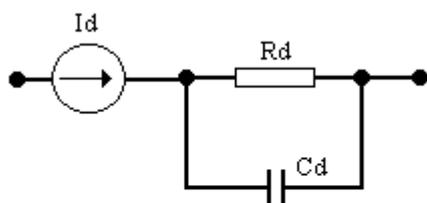


### Tasa de error digital. (*Bit error rate : BER*)

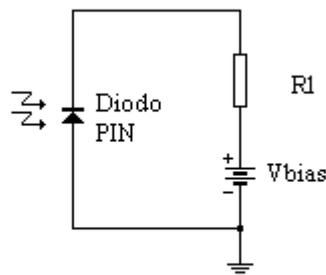
En un sistema de comunicaciones digital, el BER se usa para determinar la calidad del sistema. El BER es la relación entre la cantidad de bits correctamente recibidos y la cantidad de bits recibidos con error. Un valor de BER de  $10^{-9}$  significa que por cada  $10^9$  bits transmitidos por un sistema, 1 bit es recibido con error. Un error de esta magnitud, o incluso de algunos ordenes mas elevados, puede ser insignificante si se esta considerando la transmisión de la voz por un sistema telefónico, pero en cambio puede ser muy importante en la transmisión de datos entre computadoras. El BER y la relación señal ruido guarda estrecha relación entre si; un valor grande de SNR se traduce en un mejoramiento del BER. La tasa de error digital depende de varios y casi todos los componentes de un sistema de comunicaciones. Los fotodiodos constituyen una de las partes principales de un sistema de FO, y por ello es muy importante tener en cuenta su contribución al valor total del BER.

### Velocidad

La velocidad de operación de un fotodiodo, esta determinada por el tiempo de crecimiento del dispositivo. El modelo eléctrico de un fotodiodo PIN se muestra en la siguiente figura.



Modelo equivalente eléctrico de un diodo PIN



Circuito de polarización externo

Cuando un diodo se polariza inversamente, se comporta eléctricamente como un capacitor, por eso el modelo usado consiste en un capacitor ( $C_d$ ) en paralelo con una resistencia ( $R_d$ ). El valor de la capacidad interna del diodo es un dato sumamente importante porque su valor afectará directamente al tiempo de crecimiento del dispositivo:

$$T_{Cd} = 2,19 \cdot R_i \cdot C_d$$

Donde:

- $T_{Cd}$  : tiempo de crecimiento del fotodiodo.
- $R_i$  : Resistencia de carga exterior (Debido a que normalmente  $R_i \ll R_d$ )
- $C_d$  : capacidad interna del diodo.



El ancho de banda de un fotodiodo puede aproximarse con la siguiente expresión:

$$AB = 0,35/Tcd$$

Desde luego es posible encontrar una relación entre el ancho de banda de un fotodiodo y la constante de tiempo determinada por su capacidad y la resistencia de carga.

$$AB = 1 / ( 2.\pi . RI Cd)$$

### **Montaje o encapsulado**

Al igual que los emisores, los detectores para fibras ópticas son suministrados por los fabricantes en varios tipos de montajes y encapsulados (Packaging), siendo los mas populares el "pigtail" y el receptáculo con microlentes. También, de manera análoga, estos encapsulados presentan perdidas de acoplamiento que ocurren debido a diferencias de diámetro y apertura normal, que en principio pueden ser calculadas mediante las ecuaciones estudiadas en las lecciones anteriores. Sin embargo, normalmente los fotodiodos se fabrican con valores de AN y "área efectiva" comparativamente grandes respecto de las fuentes y fibras ópticas más populares, de manera que las perdidas por acoplamiento son sensiblemente menores que las que se presentan con los emisores.

El tipo de encapsulado o montaje también tiene influencia en las "perdidas por retorno" que ocurren debido al efecto de reflexión de Fresnel (estudiado previamente). Las perdidas por retorno se especifican dando la relación entre la potencia óptica que llega al detector y la cantidad que se devuelven a la fibra en dB y se dan para un valor de longitud de onda específica, (p.ej. : un valor de -20 dB de perdidas por retorno significa que el 1 % de al potencia incidente es devuelta a la fibra).

Una especificación también importante, y que no suele ser tomada en cuenta, es el tiempo y la temperatura máxima de soldadura, que es particularmente crítico en los montajes en Pigtail, (una temperatura o un tiempo excesivos pueden alterar la unión entre chip y la fibra de manera irreversible).

Resumen:

- Los transductores optoelectronicos mas comúnmente usados en sistemas de fibras ópticas son los fotodiodos.
- Los fotodiodos PN se usan en aplicaciones de bajos costos y prestaciones (distancias cortas, reducidos anchos de banda). Para aplicaciones de media y larga distancia, los detectores mas populares son los fotodiodos PIN y los fotodiodos de Avalancha (APD).
- Para que un fotodiodo trabaje como detector debe ser polarizado inversamente.
- Los fotodiodos APD poseen una mayor sensibilidad espectral debido a que operan en base al fenómeno de fotomultiplicacion.
- Las principales especificaciones a tener en cuenta para la selección de un fotodiodo para un propósito determinado son: Sensibilidad espectral. Eficiencia cuántica. Voltaje de polarización. Ruido. Relación señal ruido (o BER). Velocidad. Montaje o encapsulado.



- Las dos causas principales de generación de ruido en los fotodiodos son: El ruido térmico. El ruido de granalla.
- La velocidad de un fotodiodo, esta limitada principalmente por su capacidad interna.
- El ancho de banda y el tiempo de crecimiento de un fotodiodo, están relacionados entre si por ecuaciones sencillas.
- Las pérdidas de potencia óptica en un fotodiodo son siempre comparativamente menores que en los emisores.

## COMPONENTES PARA FIBRAS ÓPTICAS

Para la instalación de un sistema con FO, se utilizan algunos pocos elementos y técnicas que son idénticos a los usados en sistemas con conductores de cobre, Hay otro grupo que tiene ciertas similitudes pero que requieren un tratamiento y manipulación especial, en tanto que existe un tercer conjunto de componentes que son decididamente diferentes en su concepción, fabricación e instalación. El tema general de la conectorización juega un rol muy importante en el comportamiento final de un sistema de transmisión por FO, y por lo tanto debe ser tenido en cuenta durante el diseño del mismo. Las fibras se conectan a las fuentes, detectores, y otras fibras mediante conectores y empalmes.

Los conectores se usan principalmente en lugares donde se requiere la posibilidad de desconectar y conectar fácil y rápidamente un determinado componente o sistema, como puede ser los extremos finales de una línea con los detectores o emisores, en tanto que los empalmes se destinan a la unión de tramos de fibras en reparaciones o en tendidos de varios kilómetros de largo (los fabricantes suministran la fibra óptica en rollos de longitudes determinadas que casi nunca coinciden exactamente con la longitud física necesaria para cada caso en particular).

Los empalmes pueden hacerse básicamente de dos maneras distintas, por un lado están los empalmes por fusión, en los cuales se requiere de una herramienta o maquina especial que suelda las fibras mediante el uso de un arco eléctrico, y por el otro lado existen los empalmes mecánicos. Ambos tipos de uniones están destinadas a ser permanentes, si bien algunos fabricantes ofrecen también cierto tipo de empalmes mecánicos que pueden desconectarse y volver a usarse, aunque, desde luego, un número reducido de veces.

También existen como componentes usados para distribución de señales ópticas en sistemas de múltiples fibras, el equivalente óptico de acopladores, llaves selectoras y distribuidores.



## Conexiones y empalmes para fibras ópticas

La conectorización en el campo de las fibras ópticas es bastante más complicada que en sistemas tradicionales que usan conductores de cobre, ya que se requiere no solamente la unión física de las fibras, sino el perfecto alineamiento óptico de los componentes a fin de que se produzca una mínima pérdida de potencia óptica.

En la actualidad existen varios tipos de conectores y empalmes cuyo diseño se ha efectuado teniendo en cuenta las principales causas que ocasionan pérdidas con el fin de minimizarlas.

En la elección apropiada de un tipo de conector, o una técnica de empalme adecuada para cada caso en particular intervienen varios factores, entre los cuales hay aspectos técnicos y económicos. Respecto de los aspectos técnicos el tema pasa por el conocimiento de cuales son los factores que influyen en los efectos que las interconexiones producen sobre un sistema. El principal efecto es, desde luego, la atenuación; pero también hay otras consecuencias que se derivan, y que se verán seguidamente.

### Las causas de las pérdidas en una interconexión óptica

Podría decirse que en general las pérdidas de potencia óptica en la interconexión de fibras o componentes, obedecen a tres diferentes factores:

1. Factores intrínsecos.
2. Factores extrínsecos.
3. Factores del sistema.

Los factores intrínsecos: son principalmente aquellos que se derivan de las pequeñas variaciones en la geometría y composición de una fibra que producen alguno de los efectos que se muestran a continuación.

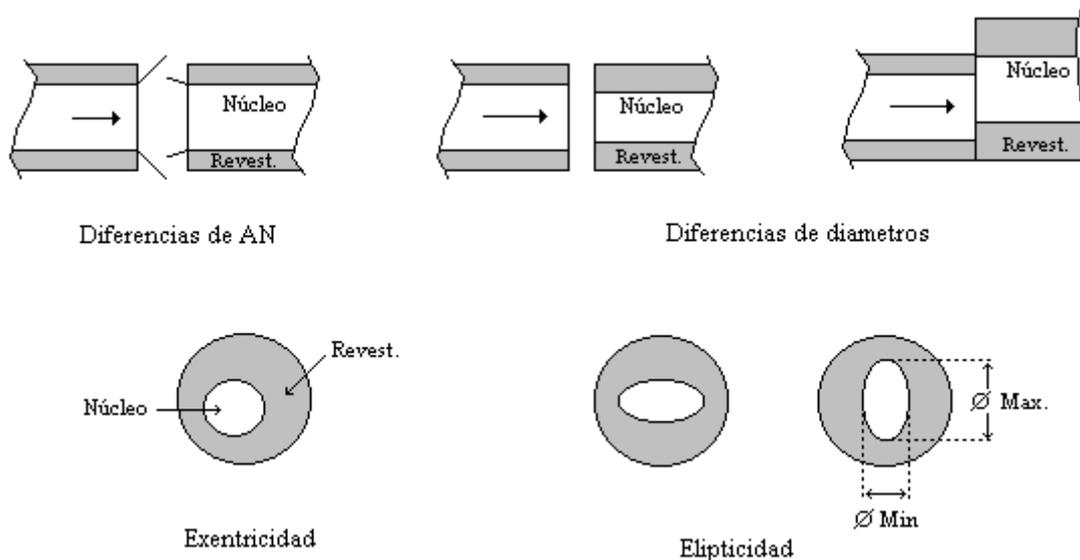


Figura 9-1



- Las pérdidas por diferencias de apertura numérica y diferencias de diámetros: responden esencialmente a la misma causa estudiada previamente en lecciones anteriores al considerar los detectores y emisores.
- Las pérdidas por excentricidad: ocurren cuando el núcleo de la fibra no está perfectamente centrado en el revestimiento. Idealmente los ejes del núcleo y el revestimiento deberían ser los mismos, pero siempre existe una pequeña desviación que depende de la calidad de la fibra.
- Las pérdidas por elipticidad: se deben a que las secciones del núcleo y/o el revestimiento no son perfectamente circulares sino que presentan cierta deformación. Por este motivo el monto de las pérdidas que pueden darse en la interconexión dependerá de la orientación entre sí de las fibras.
- Las pérdidas por diferencias en el diámetro del revestimiento: se producen cuando el diámetro externo de este último es diferente entre las dos fibras que se acoplan.

Los factores intrínsecos que causan pérdidas en las interconexiones de fibras ópticas dependen sobre todo de las tolerancias en las dimensiones mecánicas de las mismas. Estas tolerancias quedan definidas principalmente durante el proceso de fabricación empleado y en menor medida por las condiciones de almacenamiento, y manipulación posterior de la fibra. La siguiente tabla muestra algunos valores típicos de tolerancias que influyen en los factores intrínsecos para las fibras ópticas actuales.

Tabla 1. Tolerancias de factores intrínsecos.

Factor intrínseco	Tolerancia
Diámetro del núcleo (50 $\mu\text{m}$ )	$\pm 3 \mu\text{m}$
Diámetro del revestimiento (125 $\mu\text{m}$ )	$\pm 3 \mu\text{m}$
AN (0,26)	$\pm 0,015$
Excentricidad (*)	$\leq 3 \mu\text{m}$
Elipticidad del núcleo (**)	$\leq 0,98$
Elipticidad del revestimiento (**)	$\leq 0,98$

(\*) La excentricidad es la diferencia entre el eje del revestimiento y el eje del núcleo

(\*\*) La elipticidad del núcleo, o del revestimiento está definida por la relación entre los valores del menor diámetro y el mayor diámetro de las secciones respectivas.

Los factores extrínsecos: son debidos a las variaciones de orientación o separación de la fibra dentro de un conector o un empalme. Siempre que dos componentes de un sistema de FO que se conectan no están perfectamente alineados ocurren pérdidas de potencia óptica. La siguiente figura ilustra sobre los distintos factores extrínsecos de pérdidas.

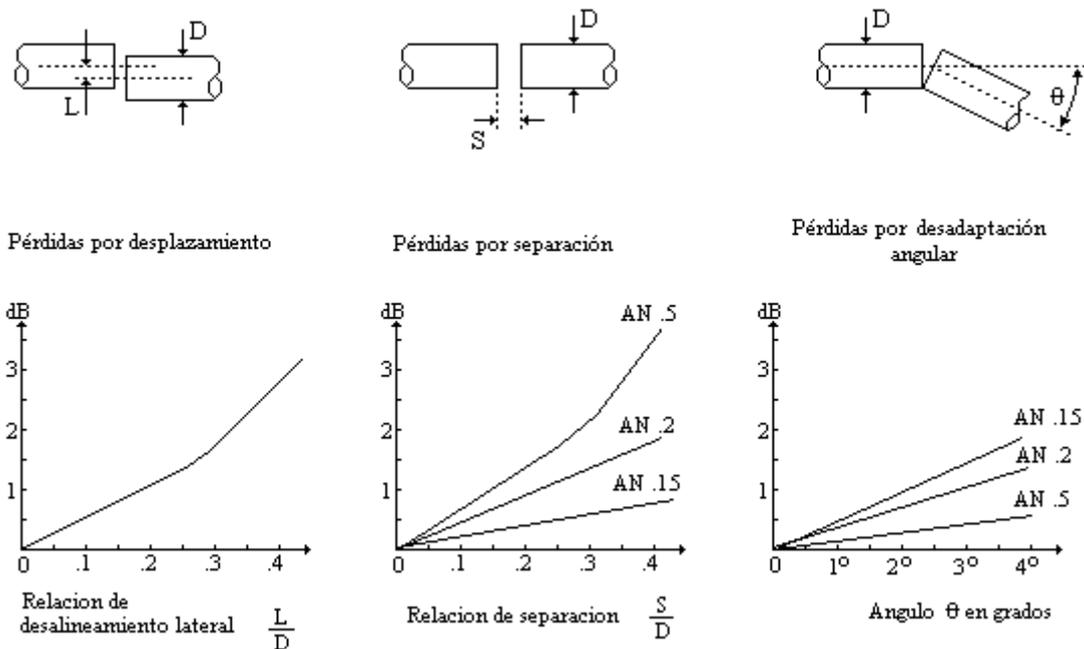


Figura 9-2

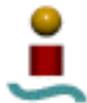
Las pérdidas debidas a desplazamiento lateral ocurren, por ejemplo, cuando en un conector no existe una perfecta alineación de los ejes centrales de los dos componentes a unir. (El efecto que se produce es esencialmente el mismo de las pérdidas por excentricidad). La relación entre el valor del desplazamiento lateral entre los dos ejes de las fibras a unir, y el diámetro del núcleo de las mismas, se conoce como "Relación de desalineamiento lateral"

$$\text{Relación de desalineamiento lateral} = L/D$$

L: desplazamiento lateral  
D: diámetro del núcleo

La separación entre bordes de dos fibras produce pérdidas debido principalmente a dos causas:

- Algunos de los rayos luminosos se reflejan de vuelta hacia la fibra de salida en la frontera núcleo-aire de la misma, otros se reflejan en la frontera aire-núcleo de la fibra de entrada. Este efecto fue estudiado en lecciones previas y se conoce "pérdidas por reflexión de Fresnel".
- En el caso de fibras multimodo, los rayos correspondientes a los modos de orden superior, simplemente no alcanzan a entrar en la fibra de entrada debido a que se pierden en el espacio de separación. La magnitud de las pérdidas por separación dependen del valor de apertura numérica de la fibra y se reducen notablemente a medida que la AN es menor.



Si el espacio de separación está ocupada por aire, el valor de las pérdidas por reflexión de Fresnel que se produce es de alrededor de 0,34 dB. Este valor de pérdidas puede parecer despreciable, pero para un sistema en el que existan varias uniones (conectores o empalmes) puede terminar transformándose en un dolor de cabezas, por este motivo se suele usar un "fluido de adaptación de índice" para rellenar el espacio y reducir así las pérdidas. El fluido de adaptación es normalmente un gel transparente cuyo índice de refracción es el mismo o está muy próximo al del núcleo de la fibra, y también contribuye a reducir las pérdidas de modos de orden superior en fibras multimodo.

La reflexión de Fresnel también causa que parte de la energía que viaja por la fibra sea devuelta hacia el emisor y si la potencia reflejada es considerable puede llegar a interferir con el funcionamiento del mismo, esto puede ser particularmente grave en las fibras monomodo, donde se usan emisores Láser. Un recurso usado en este caso para disminuir la potencia reflejada es el que se pone en práctica en los conectores llamados de contacto físico (PC).

La desadaptación angular entre dos fibras también introduce pérdidas al sistema. Las terminaciones de dos fibras que van a ser acopladas deben ser perpendiculares al eje de las mismas y paralelas entre sí. Por este motivo, el pulido y acabado plano de la sección de las fibras que se cortan para conectarse es de fundamental importancia en orden a minimizar la pérdida de potencia óptica.

Desde luego que la limpieza de las superficies de los cortes es otro de los factores extrínsecos que más contribuyen a las pérdidas en los conectores. Existen técnicas de limpieza así como solventes apropiados para tal fin cuya utilización es corriente y debe formar parte del juego de herramientas del operario calificado para esta tarea.

Los factores del sistema también contribuyen sobre las pérdidas que una unión o interconexión producen en el propio sistema, ya que el valor de las mismas depende en cierta medida del lugar donde la interconexión está efectuada. Por ejemplo, supongamos que en un determinado sistema se usa una técnica de empalmes que genera modos fugados (que se propagan a través del revestimiento). Si un empalme se efectúa cerca del receptor, aumentará la dispersión debido a la presencia de dichos modos fugados. El mismo empalme hecho cerca del emisor no tendrá el mismo efecto, ya que los modos fugados se atenuarán rápidamente con la distancia y casi con seguridad no alcanzarán a llegar al detector. Por otro lado, las pérdidas por retorno serán más perjudiciales si el empalme o conector que las generan están cerca del emisor, ya que en caso contrario (si están alejados del mismo) la potencia reflejada se atenuará naturalmente debido al trayecto.

### **Pérdidas por inserción**

La pérdida de potencia que se produce al intercalar un conector o un empalme en un tramo de fibra óptica determinado se conoce como "pérdidas de inserción". Para determinar las pérdidas por inserción de una unión, los fabricantes usan técnicas y métodos de medición normalizados, siendo la norma más comúnmente usada, la fijada por La "Electronic Industries Association" (EIA). En forma simplificada la medición se efectúa determinando la diferencia entre los niveles de potencia obtenidos en el extremo de una fibra óptica de longitud determinada antes y después de efectuar un corte en la misma e intercalar el conector o la unión correspondiente.

$$\text{Pérdidas por inserción} = 10 \log (P1/P2)$$

P1: potencia medida con la fibra entera

P2: la potencia medida con el conector intercalado



Pero debe tenerse en claro que la pérdida especificada es válida para fibras idénticas, de lo contrario deben agregarse las pérdidas por diferencias de diámetro y de apertura numérica (mediante las expresiones estudiadas en las lecciones previas). También hay que tener en cuenta que puede existir un valor adicional de pérdidas por reflexión de Fresnel.

Los valores típicos de pérdida de inserción de un buen conector moderno para fibras ópticas de vidrio están en torno a los 0,3dB, lo cual está bastante por debajo del valor de 0,75dB permitido como máximo por varias normas de interconexión de redes. En cuanto a las pérdidas que se producen en los empalmes, estas varían de acuerdo a si el mismo es mecánico (pérdidas no menores a 0,1 dB) o si es por fusión (menores de 0,1 dB).

El reducido valor de pérdidas por inserción que se logran hoy en día se ha dado como resultado de una mejora permanente en la tecnología de fabricación. Inicialmente, cuando aparecieron las primeras fibras ópticas comerciales, se consideraban razonables pérdidas de hasta 1,5dB, paulatinamente las mejoras permitieron superar la barrera de 1dB hasta llegar a los valores actuales de alrededor de 0,3dB.

Se agrega además que en la actualidad existen varios tipos de conectores, pero estos diferentes tipos, que son de uso popular en Europa, Japón y los Estados Unidos, normalmente difieren entre sí, y en un país como el nuestro, que no genera su propia tecnología sino que la adopta de acuerdo a la conveniencia, no es descabellado pensar que el ingeniero proyectista o de mantenimiento, deba enfrentar situaciones donde haya una mezcla de normas, y la elección acertada para una aplicación específica dependerá en gran medida del conocimiento de las características principales de cada tipo.

## Requerimientos de los conectores para fibras ópticas

Se pueden listar por lo menos seis características principales que son deseables y deben ser tenidas en cuenta a la hora de seleccionar un conector para una aplicación determinada. Dichas características o requerimientos son:

- Bajo valor de pérdidas de inserción.
- Buena estabilidad ambiental.
- Facilidad de instalación.
- Mínima dispersión de las tolerancias (Mínima variación del valor de las pérdidas entre dos conectores del mismo tipo).
- Alta repetibilidad (Número de veces que un conector puede ser desconectado y vuelto a conectar sin degradarse).
- Bajo costo.

## Tipos de conectores

En líneas generales, casi todos los conectores usados para fibras ópticas consisten en:

- Un “casquillo” o férula, cuya función es fijar y alinear la fibra propiamente dicha. La fibra se inserta mediante la técnica apropiada en la férula, y el conjunto se asegura mediante una gota de resina epoxi, lo cual permite el pulido y acabado del borde frontal.
- Un “cuerpo” o cápsula, con su correspondiente sistema de encastre o rosca que sirve para mantener el conector en su sitio.
- Un “manguito” o refuerzo para la descarga de tensiones mecánicas.



Sin embargo la múltiple variedad de conectores que existen son, fuera de estas similitudes, radicalmente diferentes, por lo cual se hace necesario establecer una clasificación a fin de diferenciarlos. Pueden seguirse varios criterios para establecer dicha clasificación y no hay una regla que sea mejor que la otra.

Por una parte puede establecerse una diferenciación teniendo en cuenta el material que se emplea para su fabricación; así se tienen por un lado los conectores de "plástico moldeados" y por el otro lado los de "metal maquinado". (Los conectores de plástico son mas baratos que los de metal).

También podría clasificarse los conectores teniendo en cuenta el material de que esta construida la férula o casquillo, desde este punto de vista los hay de cerámica, de acero, y de plástico. (La férula de acero es la que permite mayor repetibilidad, le sigue la de cerámica y por ultimo la de plástico).

Otra pauta podría ser dividirlos, de acuerdo al tipo de fibra para el que están fabricado, en conectores multimodo, y conectores monomodo. En este texto se seguirá este criterio.

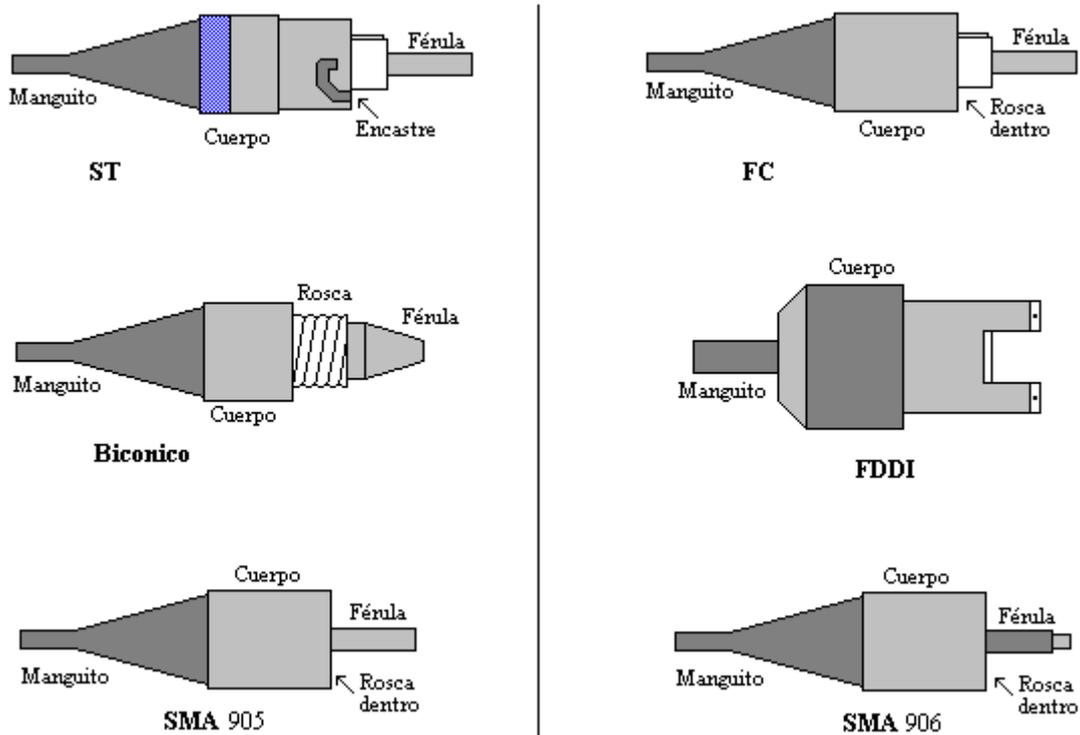


Figura 9-3 (Conectores típicos para fibras ópticas)



## Conectores multimodo

Tipo SMA: Hasta el presente, aproximadamente el 80 % de los sistemas implementados en base a fibras ópticas multimodo instalados en los EE.UU. se han realizado con un tipo de conectores denominados SMA (Sub miniature Style A) que fueron desarrollados por la firma Amphenol alrededor del año 1970. Tiene altas pérdidas, aproximadamente 0,7 dB, y por esta razón paulatinamente va siendo reemplazado por tipos más modernos. No es recomendable para la implementación de sistemas nuevos (salvo circunstancias que lo justifiquen). El tipo de conectores SMA se consigue en varios estilos, siendo los más populares:

1. Estilo 905.
2. Estilo 906.
3. Estilo SMA cerámico.

El estilo 905 fue ideado para ser usado con un tipo de fibra de núcleo más grande que las actuales cuya difusión finalmente no prosperó y se usan principalmente en sistemas de comunicaciones punto a punto para pequeñas distancias, es decir donde no existen empalmes ni repetidores.

El estilo 906 se diseñó para fibras multimodo (Tipos III, IV y V). Tiene mejor comportamiento que el estilo 905 pero es ligeramente más costoso. La principal diferencia respecto de aquel es que el casquillo es escalonado en lugar de ser recto.

El estilo SMA Cerámico se diferencia de los anteriores precisamente porque el casquillo es de material cerámico, lo cual le confiere un comportamiento muy superior con pérdidas menores a 0,5 dB pudiendo llegar a ser tan pequeñas como 0,1 dB. (La técnica de alineamiento mediante casquillos cerámicos se usa corrientemente para fibras monomodo). Es, desde luego, más caro que los tipos 905 y 906.

Tipo Biconico: Es un tipo de conector algo anticuado que fue desarrollado en Japón. Tiene una repetibilidad pobre, es susceptible a las vibraciones y tiene altas pérdidas (valores mínimos 1dB). No se recomienda su utilización.

Tipo ST: Es un tipo de conector más moderno que ha ido sustituyendo al tipo SMA. Se usa también para conexiones con fibras monomodo. Tiene pérdidas promedio menores que 0,5 dB. El cuerpo tiene un sistema de encastrado con medio giro y resorte (similar a los conectores BNC para coaxiales) que es bastante bueno para ambientes con vibraciones. Es fabricado por y para varias firmas (AMP, 3M AT&T) con algunas diferencias físicas entre sí, (Férula de acero y/o de cerámica) pero sin perder la compatibilidad.

Tipos de plástico: Otro tipo de conectores usados para fibras multimodo son los de férula plástica simple y los DNP (Dry No Polish). En los primeros el conector es completamente de plástico y la fibra se asegura a la férula mediante resina epoxi y un sistema de sujeción mecánica (crimpeado). En cambio el conector DNP no requiere el uso de pegamento y la fijación de la fibra se hace solo mediante presión. Son conectores de muy bajo costo, pero tienen elevadas pérdidas (entre 2 y 4 dB). Se usan principalmente con fibras multimodos de plástico, y son apropiados para enlaces de longitudes no mayores a 50 m y con velocidades por debajo de los 30 Mbits/seg.



## Conectores Monomodo

La principal diferencia entre los conectores monomodo y los multimodo, son las tolerancias permitidas, que son mucho más estrictas en los primeros. En realidad hay algunos tipos de conectores que pueden ser usados indistintamente para uno u otro caso en función precisamente de las tolerancias especificadas, como por ejemplo el Tipo ST, o alguno de los siguientes:

Tipo FC: También conocido como tipo FCN, fue desarrollado por la NTT de Japón. Tiene, por ejemplo, tolerancias menores al 1% para la excentricidad, y pérdidas de inserción por debajo de 0,4 dB. Es muy común en la industria de la televisión por cable, y existen versiones denominadas de contacto físico que se identifican por el agregado de la sigla PC (Physical Contact). La técnica del contacto físico consiste básicamente en un acabado redondeado del borde frontal de la férula y fibra, con esto se consiguen dos efectos. Primero los dos elementos (fibras o transductores) pueden entrar en contacto de manera más efectiva. Segundo El borde redondeado hace que la luz reflejada de vuelta hacia el núcleo de la fibra lo haga en una dirección distinta de la incidente con lo cual se termina dispersando a través de la cubierta, (lo cual es muy conveniente a fin de no saturar el emisor).

Tipo SC: Es un nuevo tipo de conector modular de plástico de alta densidad con sistema de encastre que permite el adosado de dos conectores en un conjunto que solo puede conectarse en un sentido, lo cual lo hace muy útil para el caso de un dúplex (cada fibra lleva información en un único sentido). Tiene bajas pérdidas, debajo de 0,5 dB.

Tipo FDDI: Es un conector duplex (doble) con llave de uso estándar en sistemas full duplex. Es muy similar al conector tipo ESCON usado por la firma IBM para sus sistemas. No son compatibles entre sí.

Resumiendo:

- Los conectores para fibras ópticas se diseñan para facilitar la conexión y desconexión rápida el mayor número de veces con las menores pérdidas posibles.
- Un empalme se usa para conectar dos fibras en forma permanente.
- La principal especificación de conectores y empalmes son las "Pérdidas por inserción".
- El material empleado para la férula influye en la repetibilidad del conector.
- Las férulas de cerámica son las que permiten menores tolerancias mecánicas.
- Las férulas de acero son las que permiten mayor repetibilidad.
- El encastre tipo bayoneta soporta mejor las vibraciones que el roscado.
- Los factores que causan pérdidas en las interconexiones de fibras ópticas son:

Factores de pérdidas.		
Factores intrínsecos	Factores extrínsecos	Factores del sistema
• Diferencias de AN	• Desplazamiento lateral	• Posición de la unión.
• Diferencias de diámetros	• Desalineado angular	
• Excentricidad.	• Separación entre bordes	
• Elipticidad.	• Acabado de la superficie	



## CONSIDERACIONES Y PROCEDIMIENTOS GENERALES PARA EL DISEÑO DE UN SISTEMA DE COMUNICACIONES POR FIBRAS ÓPTICAS.

Por lo general, el diseño de un sistema de comunicaciones por fibras ópticas, se hace buscando satisfacer dos condiciones básicas:

- El detector debe poder recibir un nivel de señal suficientemente grande como para asegurar que la señal recuperada en el receptor sea exactamente la misma que la emitida en el transmisor.
- La totalidad del sistema debe operar a la velocidad suficiente para satisfacer el ancho de banda en un sistema analógico o su equivalente en un sistema digital.

Para dar cumplimiento a estos requerimientos, el proceso de diseño puede dividirse básicamente en dos fases principales:

- El análisis a partir del balance de las potencias.
- El análisis desde el punto de vista de los tiempos de crecimiento.

El diseñador normalmente debe conocer los siguientes parámetros, que corresponden al sistema que desea implementarse:

1. El tipo de señal a transmitir (Vídeo, Telefonía, Datos, etc).
2. El ancho de banda, para sistemas analógicos, o su equivalente para sistemas digitales (la cantidad de bits por segundo).
3. La longitud del enlace (en kilómetros).
4. La relación Señal - Ruido (SNR) o la Tasa de Error Digital (BER).
5. Las condiciones ambientales del sistema (Tendido externo, interno etc..).

Partiendo del conocimiento de estos parámetros, el diseño de un sistema consiste en la elección de los componentes apropiados para satisfacer los requerimientos del proyecto. Lo que sigue es una secuencia de pasos a cumplir que servirá como guía para el trabajo. En lo que respecta al orden a seguir, parece ser más conveniente comenzar teniendo en cuenta lo concerniente al balance de las potencias, y luego verificar si los componentes elegidos cumplen con el aspecto del análisis de tiempos.



Diseño del sistema partiendo desde el punto de vista del balance de las potencias.

Se supone inicialmente que se debe encarar un diseño con base en las siguientes especificaciones y/o requerimientos.

- Longitud del enlace : 14 km.
- Tipo de sistema: Transmisión / recepción digital.
- Promedio máximo de bits (Bit Rate) : 4 Mbps. (4 Mega bits por segundo).
- Tasa de error digital permitida :  $\leq 10^{-9}$

Paso 1:

Las especificaciones fijan que se trata de un sistema de tipo digital.

Paso 2:

Por tratarse de un sistema digital, no se trabaja con el concepto de ancho de banda.

Paso 3:

Por la misma razón que en el paso anterior, ahora se debe tomar nota del valor de promedio de bits ( en nuestro caso 4 Mbps).

Paso 4:

Se debe tomar nota de la longitud del enlace, que en este caso es directamente el valor especificado igual a 14 km.

Paso 5:

En este paso, se debe determinar si se trata de un sistema que va a usarse en un ambiente interior (dentro de un edificio) o exterior. Los requerimientos no especifican literalmente cual es el caso, pero dada la longitud del enlace es de suponer que se trata de un sistema para exteriores.

Paso 6:

Como el sistema que se debe diseñar es digital no se trabaja con el concepto de relación señal ruido.

Paso 7:

Por la misma razón que en el paso anterior, ahora se debe tomar nota del valor de tasa de error digital, ( en nuestro caso  $BER \leq 10^{-9}$  ).

Paso 8:

En este paso vamos a elegir el mejor tipo de código de modulación, para nuestro sistema. Esto es posible de hacer porque los requerimientos iniciales no especifican nada al respecto. A veces el diseñador se encuentra con que por las condiciones de contorno del diseño, el tipo de modulación ya esta impuesto.

Para elegir el mejor tipo de modulación un criterio podría ser optar por el sistema que proporcione un valor mas reducido de tasa de error digital. Se sabe que el BER es mínimo cuando el "factor de ciclo de trabajo" es 50 %.

Los códigos Manchester y Bifase - M tiene un factor de ciclo de trabajo de 50 %. Sin embargo, su implementacion es muy compleja y además aumenta al doble el ancho de banda necesario ya que se requieren dos símbolos por cada bit (Baud).



Para un sistema relativamente simple como el que se está planteando, sería más conveniente optar por un sistema de codificación más sencillo como puede ser el Código sin retorno a cero (NRZ) que tiene un ciclo de trabajo cuyo valor puede variar entre 0 y 100 % y, aunque necesitan una relación señal ruido más alta, no requieren una electrónica tan compleja.

La decisión definitiva sobre el código a adoptar se tomará finalmente al llegar a una primera conclusión del trabajo. Si el sistema cumple con el valor de BER requerido, se acepta, de lo contrario, se debe elegir otro tipo de código. Entonces, inicialmente escogemos NRZ, y a partir de esta decisión el ancho de banda es equivalente al valor de promedio de bits, es decir 4 Mbps.

Paso 9:

En este punto del trabajo, se debe estimar el valor mínimo de potencia óptica requerida por el sistema. Esta será la Sensibilidad del receptor ó  $P_{\min}$  y se calcula en dos (2) partes: En la primera parte se usará la ecuación b-1:

$$P_{\min} = 2 e F K (\Delta f) / R \quad (b-1)$$

El valor de F depende del tipo de detector a utilizar. Dado que los diodos APD de silicio y de AsGa se usan normalmente solo para aplicaciones especiales, (que no es este caso), se optará por un fotodiodo PIN, en cuyo caso  $F=1$ .

El valor de K depende de la relación señal ruido del sistema. Si se tratara de un sistema analógico, dicho valor podría calcularse tomando el antilogaritmo de la relación SR. mediante la siguiente expresión:

$$K = \text{Log}^{-1} ( SR/10) \quad (b-2)$$

En el caso de tratarse de un sistema digital se usa el BER, y aunque en principio sería posible obtener analíticamente el valor equivalente de relación SR a partir del BER, dicha operación sería muy engorrosa ya que la correspondencia obedece a una múltiple variedad de factores que tienen que ver con el diseño del circuito. Para un propósito práctico, es posible obtener el valor equivalente de relación SR a partir del BER y viceversa, mediante el uso de ábacos. Este gráfico está trazado sobre la base de considerar sistemas de codificación con ciclos de trabajo de 50 %. En caso de usar códigos con ciclo de trabajo entre 0 y 100 % se debe adicionar 6 dB al valor conseguido. Una vez que se ha obtenido y corregido el equivalente se introduce en la ecuación b2 :

$$K = \text{Log}^{-1} ( 21,4 \text{ dB}/10) = 138$$

El valor de  $\Delta f$  es el ancho de banda para un sistema analógico o el valor del promedio de símbolos para un sistema digital, que en este caso, y de acuerdo a lo establecido en el paso 8 vale:

$$\Delta f = 4 \text{ Mbps}$$

El valor de R es la responsividad del fotodiodo a usar. Como ya se ha optado por el tipo de detector a usar, se fija inicialmente este valor en  $R = 0,5 \text{ A/W}$  que es un valor típico para un fotodiodo PIN.

Ahora que se tienen todos los valores de las variables de la ecuación b-1 puede calcularse el valor de potencia óptica mínima:

$$P_{\min} = 3,53 \cdot 10^{-10} \text{ W}$$



La segunda parte del paso consiste en pasar del valor de potencia óptica en W a dBm , lo cual se hace para facilitar el cálculo:

$$\text{dBm} = 10 \log (P_{\min} / 1 \text{ mW}) = - 64,5 \text{ dBm}$$

Paso 10:

Ahora que se ha calculado el valor de sensibilidad de recepción, se debe proceder a elegir el detector apropiado, para lo cual deben consultarse manuales de componentes y hojas de datos. Un listado de fotodetectores con un resumen de las especificaciones principales puede encontrarse al final de los pasos. Para este problema el detector que mas se adecua al valor calculado de sensibilidad resulta ser el diodo PIN tipo DO1.

Una vez que se ha elegido un detector, se deben asumir todas las características del mismo, lo que incluye: tiempo de crecimiento, Longitud de onda de máxima sensibilidad, el encapsulado, etc.

Paso 11:

Se debe determinar a continuación que tipo de fibra óptica va a usarse para la implementacion del enlace. Para ello se parte de dos interrogantes básicos:

1. ¿Cual es la distancia de transmisión?
2. ¿Cual es la velocidad de transmisión?

Respecto de la distancia o longitud, un buen criterio puede ser fijar un valor de 1 km como punto de separación entre largas y cortas distancias. En relación con la velocidad de transmisión, para el estado actual de la técnica el valor de producto *ancho de banda por longitud* que se considera como punto de separación entre velocidades elevadas y bajas es:

- 1 Ghz . km (para sistemas analógicos)
- 1 Gbits . km (para sistemas digitales)

Utilizando los valores de Promedio de bits, y longitud especificados, se puede calcular el valor del producto *ancho de banda por longitud*, que en este caso es:

$$4 \text{ Mbps} . 14 \text{ km} = 56 \text{ Mbits} . \text{ km}$$

De lo expuesto se deduce que el sistema que debe diseñarse, es de larga distancia y de baja velocidad.

Normalmente el criterio para optar entre tipos de fibras ópticas, puede ser resumido en la siguiente tabla:

	Larga distancia	Corta distancia
Alta velocidad	Monomodo	1° Monomodo 2° Multimodo
Baja velocidad	1° Multimodo 2° Monomodo	Multimodo

De donde se deduce que para el caso que nos ocupa la elección mas apropiada corresponde a una fibra Multimodo.



El paso 11 no ha concluido todavía, pues falta elegir la fibra óptica a usar. Para este fin se deben consultar los manuales y hojas de datos proporcionados por los fabricantes. Un listado de fibras ópticas con un resumen de las especificaciones principales puede encontrarse al final de los pasos a seguir. Se deberá prestar atención a la columnas correspondientes a la ventana de trabajo, que debe ser 850 nm en razón que el detector seleccionado opera en esta longitud de onda, y por otro lado el valor del producto ancho de banda por distancia, que en el caso planteado en este problema es de 56 Mbps.km, buscando el tipo de fibra cuyos valores sean los mas próximos. En este caso la mejor elección cae sobre la fibra tipo F10.

En el caso de haber correspondido la elección de una fibra monomodo, la selección debería hacerse tomando en cuenta la ecuación b-3:

$$\text{Disp} = .187/(\text{AB} \cdot \text{Long-fibra} \cdot \text{Ancho espectral}) \quad (\text{b-3})$$

Donde AB es el ancho de banda (para sistemas analógicos), o el promedio de bits (para sistemas digitales).

Con este valor se busca de manera análoga entre las fibras monomodo, teniendo en cuenta la longitud de onda del detector seleccionado.

Paso 12:

La atenuación de la fibra viene especificada en la hoja de datos. Allí puede verse que las perdidas dependen de la ventana elegida. En este caso, dado que la longitud de onda de trabajo es 850nm, el valor de atenuación es: 3,5 dB/km

Paso 13:

Determinamos ahora la atenuación total de la fibra multiplicando el valor de atenuación por km., por la longitud total del enlace.

$$3,5 \text{ dB/km} \cdot 14 \text{ km} = 49 \text{ dB.}$$

Paso 14:

En este paso se determina que tipo de empalmes usaremos. Normalmente el criterio a seguir es optar por empalmes mecánicos para fibras multimodo, y empalmes por fusión para fibras monomodo. En una primera instancia nos ajustaremos a este criterio y proseguiremos en el calculo suponiendo que se utilizan empalmes mecánicos, luego si es necesario se revisara el paso.

Paso 15:

El número de empalmes a usar quedará determinado por la relación entre la longitud total del enlace y el largo de la cantidad de fibra óptica de cada rollo. En la carta de selección de fibras, puede verse que el fabricante suministra la misma en carretes de 500 m, por lo tanto:

$$\text{Nro. de empalmes} = (14 \text{ km} / 0,5 \text{ km}) - 1 = 27 \text{ empalmes}$$

Debe ahora tomarse una decisión sobre como terminar los extremos de la fibra que corresponden al detector y al emisor. El fotodiodo seleccionado puede conseguirse para montaje con conector, o con "pigtail". Un buen criterio seria optar por este ultimo tipo de encapsulado y suponer que lo propio sucederá con el emisor a elegir. Por lo tanto agregamos dos (2) empalmes más al total calculado, y así el numero total será:

$$27 + 2 = 29 \text{ empalmes.}$$



**Paso 16:**

Los empalmes agregan atenuación al sistema. Los valores típicos son (con un criterio conservador): Empalmes mecánicos = 0,5 dB . Empalmes por fusión = 0,1 dB.

**Paso 17:**

Se calcula el monto de atenuación total agregado al sistema a causa de los empalmes:

$$\text{Perdida total de los empalmes} = 29 \text{ empalmes} \cdot 0,5 \text{ dB} = 14,5 \text{ dB}$$

**Paso 18:**

El número de conectores se decide en función de la terminación que se le dará a los extremos de la fibra. En este caso hemos optado por usar empalmes con encapsulados en pigtail, por lo tanto no hay conectores.

**Paso 19:**

El tipo de conector que se debe elegir queda determinado principalmente por el tipo de fibra elegido. Se debe optar por un conector que sea compatible con la fibra elegida, y para ello se consultan las hojas de datos de los fabricantes. Un resumen de especificaciones se encuentra al final de los pasos a seguir, y en el mismo puede buscarse el tipo de conector más conveniente para un propósito determinado. En este caso el conector apropiado es el C10, y aunque no se usará, es conveniente tomar nota del mismo por si mas adelante se decide cambiar de opinión.

**Paso 20:**

La pérdida por inserción del conector elegido es 1,2 dB.

**Paso 21:**

El valor total de pérdidas por conectores se calcula multiplicando el numero de conectores por las perdidas de cada uno. En nuestro caso el valor es 0, ya que por el momento no se usan conectores.

**Paso 22:**

En esta etapa del trabajo, corresponde elegir el tipo de emisor apropiado. La tabla que sigue es una buena guía de referencia para este propósito:

Velocidad/Longitud del enlace	Fuente
Alta velocidad/larga distancia	Laser
Alta velocidad/Corta distancia	Laser o LED
Baja velocidad/Larga distancia	Laser
Baja velocidad/Corta distancia	LED

En el paso 11 se determinó que el sistema es de larga distancia y de baja velocidad. En base a esto y en función de la tabla anterior se opta por usar un emisor tipo Láser. Corresponde a continuación elegir el tipo de emisor que más se adecua para el propósito que se persigue. Se utiliza para ello el resumen de características y especificaciones que podemos encontrar al final de los pasos a seguir y con un criterio similar al aplicado para la elección de la fibra (longitud de onda de operación) se determina el componente apropiado. La elección justa cae sobre el diodo Láser modelo SO7.



**Paso 23:**

La potencia óptica promedio de salida de la fuente se obtiene de la columna correspondiente del resumen, que en este caso es  $P = 4000\mu\text{W}$ . Como resulta más útil trabajar con potencias en dBm, se usa para ello la misma expresión que se utilizó en la parte 2 del paso 9:

$$\text{dBm} = 10 \log (P / 1 \text{ mW}) = 10 \log (4000\mu\text{W}/1 \text{ mW}) = 6,02 \text{ dBm}$$

**Paso 24:**

El factor de degradación de tiempo es un valor constante que debe agregarse siempre en el diseño de un sistema para tener en cuenta la disminución de potencia de salida del emisor, y el aumento de atenuación por envejecimiento de los componentes. Un valor apropiado es 3dB.

**Paso 25:**

El factor de degradación ambiental es también un valor constante que debe agregarse. Un criterio conservador puede ser usar un valor de 5 dB para aplicaciones en espacios externos y 2 dB para ambientes interiores. En este caso por tratarse de un sistema para exteriores, se adopta un valor de 5dB.

**Paso 26:**

La atenuación total del sistema se calcula sumando todos los factores de pérdidas del mismo:

Pérdida total de la fibra	49 dB	(paso 13)
Perdida total de empalmes	14,5 dB	(paso 17)
Perdida total de conectores	0 dB	(paso 21)
Degradación temporal	3 dB	(paso 24)
Degradación ambiental	5 dB	(paso 25)
<u>Atenuación total</u>	<u>71,5 dB</u>	

**Paso 27:**

El margen de pérdidas (Pérdida marginal total) se calcula haciendo la diferencia entre la potencia promedio de salida del emisor, calculada en el paso 23, menos la sensibilidad del detector obtenida en el paso 9:

$$\text{Margen de pérdidas} = 6,02 \text{ dBm} - (-64,6 \text{ dBm}) = 70,64 \text{ dB}$$

**Paso 28:**

Para el cálculo de la potencia en exceso del sistema, se debe restar el margen de pérdidas del valor de atenuación total del sistema:

Margen de pérdidas	70,64 dB	(paso 27)
<u>Atenuación total</u>	<u>71,5 dB</u>	(paso 26)
Potencia en exceso	-1,21 dB	

Este resultado sugiere que en este punto se ha llegado a un valor que indica que la atenuación total del sistema es excesiva, ya que la potencia disponible sobre el detector no alcanza para asegurar la correcta recepción (si bien por una diferencia muy pequeña). Se hace necesario usar algún criterio para seguir adelante. Uno de ellos sería tolerar la diferencia aceptando un valor menor de factor de degradación ambiental y/o temporal. La otra opción es tratar de reducir las pérdidas del sistema.



En este ejemplo se optará por la segunda opción, para lo cual será necesario revisar los pasos 16, 17, y 26. Si se decide cambiar y usar empalmes por fusión en lugar de empalmes mecánicos, el valor de pérdidas por empalmes de reduce a:

$$\text{Pérdida total de los empalmes} = 29 \text{ empalmes} \cdot 0,1 \text{ dB} = 2,9 \text{ dB} \text{ (revisión paso 17)}$$

Y la atenuación total será:

$$\text{Atenuación total} = 49 \text{ dB} + 2,9 \text{ dB} + 3 \text{ dB} + 5 \text{ dB} = 59,9 \text{ dB} \text{ (revisión paso 26)}$$

El valor recalculado de Potencia en exceso será:

$$\text{Pot. ex.} = 70,64 \text{ dB} - 59,9 \text{ dB} = 10,7 \text{ dB}$$

Que ahora si se puede considerar un valor aceptable.

Paso 29:

El valor de potencia en el receptor se obtiene sumando el exceso de potencia al valor de sensibilidad del mismo:

Sensibilidad del receptor	- 64,6 dBm	(paso 9)
Exceso de potencia	10,7 dB	(paso 28)
Potencia en el receptor	-53,8 dBm	

Desde el punto de vista del diseño del sistema partiendo del balance de las potencias se ha llegado al final. Corresponde ahora verificar si los componentes elegidos cumplen con las especificaciones teniendo en cuenta los tiempos de crecimiento.

Análisis de los tiempos de crecimiento.

Paso 30:

El tiempo de crecimiento requerido para el sistema depende de una suma de factores, entre los cuales está el tipo de código de modulación elegido. La tabla siguiente sirve como referencia:

Código de modulación	Tiempo de crecimiento requerido
Miller / NRZ / NRZI	$T_c = \frac{.70}{\text{Bit rate}}$
Manchester / Bifase-MRZ	$T_c = \frac{.35}{\text{Bit rate}}$



En el paso 8 se optó inicialmente por el uso del código de modulación sin retorno a cero, de manera que el valor de tiempo de crecimiento requerido para el sistema propuesto es:

$$T. \text{ crec. requerido del sist.} = .7 / 4 \text{ Mbps} = 175 \text{ ns}$$

Paso 31:

Para el cálculo del ancho de banda de la fibra, se divide el valor del producto ancho de banda por distancia obtenido de las especificaciones por la longitud total del sistema:

$$AB \text{ de la fibra} = 100 \text{ Mhz.km} / 14 \text{ km} = 7,14 \text{ Mhz.}$$

Hay que recordar que si la fibra elegida hubiera sido monomodo, el cálculo se debería efectuar sobre la base de la ecuación b-3 :

$$\text{Disp} = .187 / (\text{AB. Long-fibra} \cdot \text{Ancho espectral}) \quad (\text{b-3})$$

$$\therefore \text{AB} = .187 / (\text{Disp. Long-fibra} \cdot \text{Ancho espectral})$$

Paso 32:

El Tiempo de crecimiento de la fibra se calcula en base al ancho de banda:

$$T. \text{ Crec. fibra} = .35 / \text{AB de la fibra} = .35 / 7,14 \text{ Mhz.} = 49 \text{ ns}$$

Paso 33:

El tiempo de crecimiento del emisor, se obtiene directamente de la hoja de datos del mismo, siendo en este caso:

$$T. \text{ Crec. emisor} = 1,5 \text{ ns}$$

Paso 34:

De igual manera que con el emisor, el tiempo de crecimiento del detector se obtiene de la hoja de datos del mismo:

$$T. \text{ Crec. detector} = 1 \text{ ns}$$

Los pasos finales (35 a 38), permiten calcular las especificaciones finales del sistema. Luego de averiguados estos valores sabremos finalmente si el procedimiento ha llevado a una conclusión acertada, de lo contrario deberá retrotraerse al inicio de la rutina y rehacer los pasos necesarios.

Paso 35:

Para el cálculo del tiempo de crecimiento total del sistema se usa la ecuación b-4:

$$T. \text{ Crec. Sist} = 1,1 \sqrt{(T. \text{ Crec. fibra})^2 + (T. \text{ Crec. emisor})^2 + (\text{Crec. detector})^2} \quad (\text{b-4})$$

$$T. \text{ Crec. Sist} = 1,1 \sqrt{49^2 + 1,5^2 + 1^2} = 53,93 \text{ ns}$$



Paso 36:

Este paso se aplica en caso de que el sistema a diseñar sea analógico, (que no es el caso de este ejemplo). El ancho de banda se calcula de la siguiente manera:

$$AB = .35 / T. \text{Crec. Sist}$$

Paso 37:

Se calcula en este punto el valor final definitivo de Promedio de bits (Bit rate) del sistema, teniendo en cuenta que se optó inicialmente por codificación NRZ.

$$\text{Promedio de bits (Bit rate)} = .70 / 53,93 \text{ ns} = 12,97 \text{ Mbps.}$$

Este valor esta por encima del especificado inicialmente, de manera que cumple el requerimiento inicial.

Paso 38:

Para el calculo del valor final de relación señal ruido del sistema se usa la ecuación b-5:

$$\text{Rel. SR} = 10 \log ( P. R / 2.e.F.\Delta f) \quad (\text{b-5})$$

Los valores de  $R = 0,5 \text{ A/W}$  y  $F = 1$  fueron fijados en el paso 9 ( e es la carga elemental de un electrón:  $1,6 \cdot 10^{-19} \text{ Coul.}$ ),  $\Delta f$  es el valor de promedio de bits definitivo calculado en el paso 37.

P es la potencia óptica disponible sobre el detector en W y se calcula a partir del valor de dBm obtenido en el paso 29 (-54,3 dBm).

$$P = 1 \text{ mW} \cdot \log^{-1} (-53,8 \text{ dBm}/10) = 4,1 \text{ nW}$$

Estos valores aplicados a la ecuación b-5 permiten obtener:

$$\text{Relac. SR} = 10 \log \left( \frac{4,1 \text{ nW} \cdot 0,5 \text{ A/W}}{2 \cdot 1,6 \cdot 10^{-19} \text{ C} \cdot 1 \cdot 12,97 \text{ Mbps}} \right) = 26,5 \text{ dB}$$

Como se trata de un sistema digital, el valor obtenido debe convertirse a su equivalente de BER, para lo cual podría utilizarse el mismo ábaco del paso 9. Por desgracia el valor calculado cae fuera del mismo, sin embargo puede verse claramente, haciendo una extrapolación por la tendencia de la curva que el valor equivalente de BER estará muy por debajo del requerimiento inicial y será:

$$\text{BER} \ll 10^{-12}$$

Llegado a este punto podemos asumir que el diseño se ha completado exitosamente.

**FIBRAS OPTICAS**

LONGITUD DE ONDA DE OPERACION	Tamaño de la fibra (μm)	AN	Producto distancia. ancho de banda (MHz km)	Dispersión (ps/km.nm)	Atenuación dB/km	Longitud del rollo	Modelo #
1300 - 1550	3,2/125	0,11	----	425	2,0 - 0,3	500	FO1
780	4/125	0,10	----	285	3,0	500	FO2
850	5/125	0,10	----	120	3,0	500	FO3
1550	6,7/125	0,11	----	3,0	2,0	500	FO4
1300	8/125	0,10	----	2,0	4,0	500	FO5
850 - 1300	50/125	0,20	1000	----	8,4 - 7,1	500	FO6
850 - 1300	50/125	0,20	400	----	5,0 - 4,0	500	FO7
850 - 1300	62,5/125	0,27	250	----	3,0 - 1,0	500	FO8
850 - 1300	62,5/125	0,27	200	----	4,5 - 3,0	500	FO9
850 - 1300	100/125	0,30	100	----	3,5 - 2,0	500	FO10

g-1

**EMISORES**

Tipo	LONGITUD DE ONDA DE OPERACION	Potencia de salida μW	Ancho espectral (nm)	Tiempo de crecimiento (ns)	Corriente de excitación (mA)	Encapsulado	Modelo #
LED	1300	300	40	8,0	2	PIGTAIL	SO1
LED	850	500	35	10,0	10	SMA/PIGTAIL	SO2
LED	780	300	25	4,9	15	SMA/PIGTAIL	SO3
LED	850	30	30	0,25	20	SMA/PIGTAIL	SO4
LED	1300	1000	40	4,1	20	PIGTAIL	SO5
LASER	1300	1000	4	2,1	40	PIGTAIL	SO6
LASER	850	4000	4	1,5	60	SMA/PIGTAIL	SO7
LASER	1300	10000	4	3,0	50	SMA/PIGTAIL	SO8
LASER	1550	5000	2	2,0	50	SMA/PIGTAIL	SO9
LASER	1300	1000	4	2,1	80	SMA/PIGTAIL	SO10

g-2

**DETECTORES**

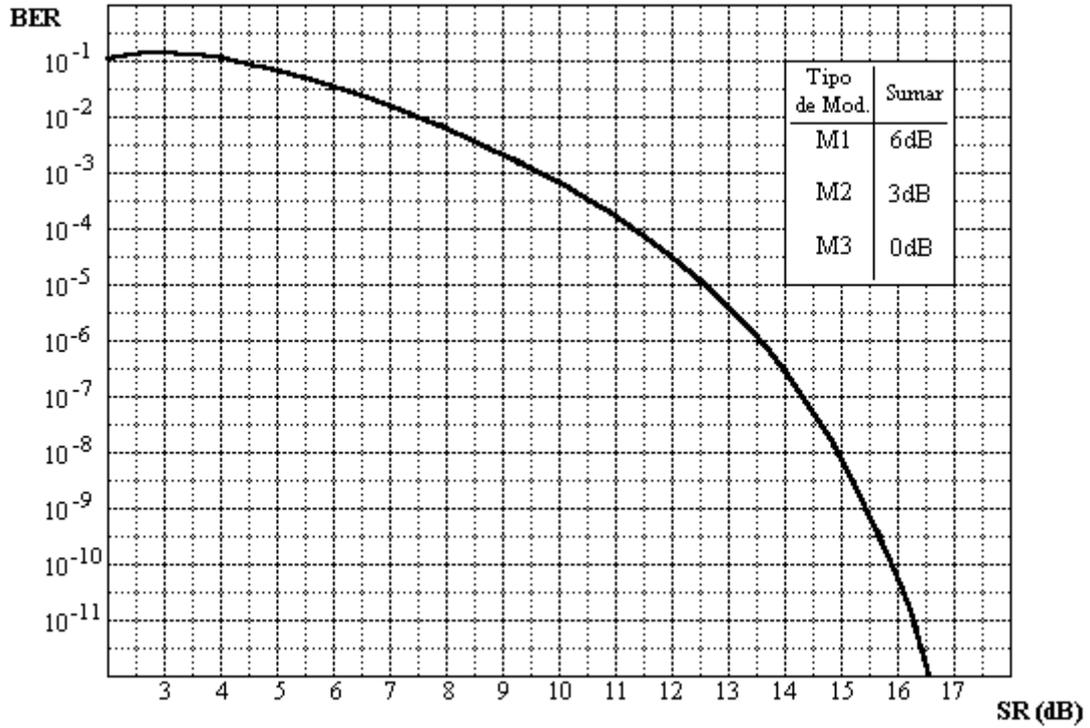
Estructura del dispositivo	Material activo	Tiempo de crecimiento	Longitud de onda de sensibilidad	Nivel de potencia mínima aceptable (dBm)	Responsividad (A/W)	Encapsulado	Modelo #
PIN	Si	1,0	850	-80	0,5	SMA/PIGTAIL	DO1
APD	Ge	2,0	850	-51	1,0	SMA/PIGTAIL	DO2
PIN	Si	0,1	850	-47	0,5	SMA/PIGTAIL	DO3
PIN	Ge	1,0	1300	-41	0,5	SMA/PIGTAIL	DO4
APD	InGaAs	1,2	1300	-36	1,5	SMA/PIGTAIL	DO5
APD	Ge	1,1	1300	-58	2,1	SMA/PIGTAIL	DO6
PIN	Si	0,9	1550	-28	0,5	SMA/PIGTAIL	DO7
PIN	Ge	1,0	1300	-51	0,5	SMA/PIGTAIL	DO8
APD	Si	0,8	1550	-36	1,9	SMA/PIGTAIL	DO9
APD	Ge	1,0	1550	-43	5,0	SMA/PIGTAIL	DO10

g-3

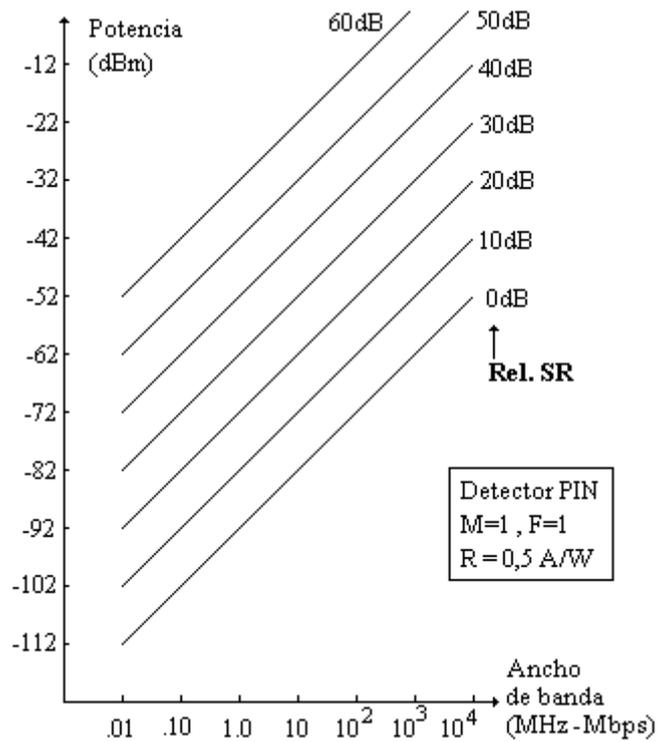
**CONECTORES**

Tipo de fibra	Tamaño de la fibra ( $\mu\text{m}$ )	Atenuación (dB)	Compatible con	Modelo #
Mono modo	3,2/125	0,25	SMA/PIGTAIL	CO1
Mono modo	4/125	0,70	SMA/PIGTAIL	CO2
Mono modo	5/125	1,0	SMA/PIGTAIL	CO3
Mono modo	6,7/125	0,35	SMA/PIGTAIL	CO4
Mono modo	8/125	0,35	SMA/PIGTAIL	CO5
Multi modo	50/125	3,00	SMA	CO6
Multi modo	50/125	1,00	PIGTAIL	CO7
Multi modo	62,5/125	1,25	SMA/PIGTAIL	CO8
Multi modo	62,5/125	1,35	SMA/PIGTAIL	CO9
Multi modo	100/125	1,20	SMA/PIGTAIL	CO10

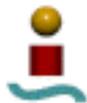
g-4



(g-5) Curva de BER vs. Relación SR.



(g-6) Potencia óptica requerida vs. AB / Bit rate



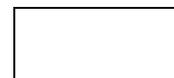
## DESARROLLO DE LA SOLUCIÓN. DIAGRAMAS DE FLUJO

Como ayuda en el desarrollo de la solución de un problema, podemos utilizar los diagramas de flujo. Un diagrama de flujo es una representación gráfica del flujo lógico de datos que se utilizará en la formulación, generalmente de una determinada parte del programa.

Los símbolos más utilizados son los siguientes:

1. Proceso

Proceso interno del ordenador; esto es, cualquier serie de transferencia de datos u operaciones aritméticas.



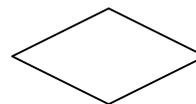
2. Entrada/Salida – E/S

Operación de entrada o salida, como lectura o escritura.



3. Decisión

Verificar si el resultado de una expresión es cierto o falso.



4. Conector

Punto de referencia que indica donde debe continuar el diagrama de flujo. Se utiliza para indicar un cambio en el flujo normal de datos (transferencia o bifurcación).



5. Terminal

Indica el comienzo o punto final de un módulo.



6. Entrada desde teclado

Indica que hay que realizar una entrada de datos desde el teclado.



7. Sentido del flujo de datos

Indica cuál es la siguiente operación a realizar a partir del símbolo actual en el que nos encontramos.



8. Procedimiento

Este símbolo sustituye a todo un subprograma cuyo desarrollo posponemos para el final.



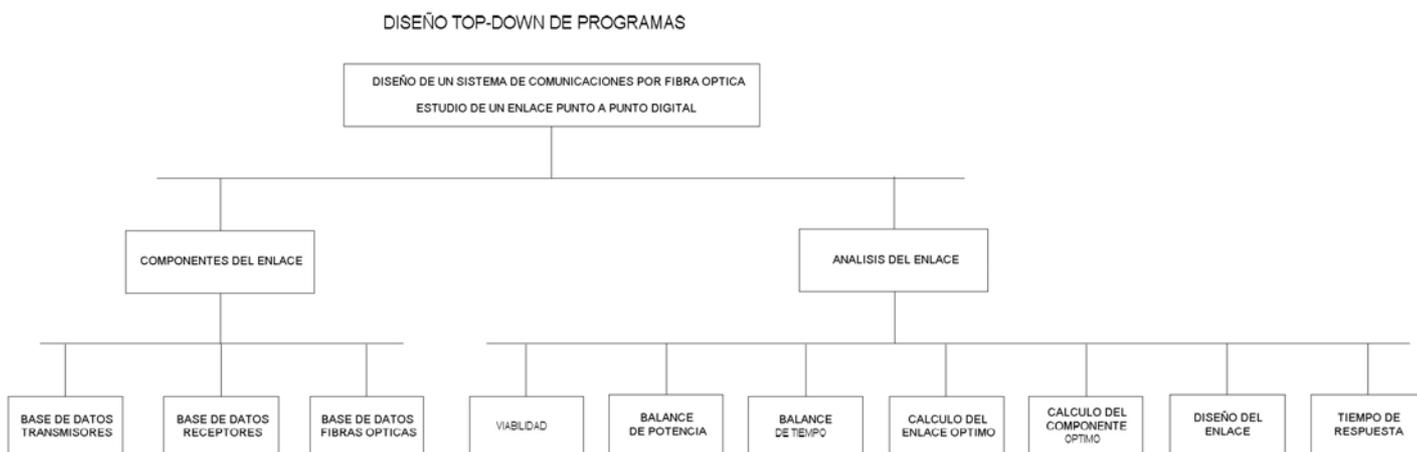


## DISEÑO TOP-DOWN DE PROGRAMAS

La solución de cualquier problema puede darse en varias formas o niveles de abstracción. Según Niklaus Wirth: “Nuestra herramienta mental más importante para competir con la complejidad es la abstracción. Por tanto, un problema complejo no deberá considerarse inmediatamente en términos de instrucciones de un lenguaje, sino de elementos naturales del problema mismo, abstraídos de alguna manera”.

El diseño top-down de un problema consiste en encontrar la solución de un problema mediante la aplicación sistemática de descomposición en subproblemas cada vez más simples.

En la figura siguiente se muestra la descomposición en subproblemas que da lugar a la estructura para el diseño del programa.



A partir de los rectángulos inferiores se desarrollarán los diagramas de flujo de las distintas funciones que constituyen el programa.

Como podemos observar en el gráfico, el análisis del problema nos conduce a una primera subdivisión del problema en dos partes. Por una parte tendremos que recopilar toda la información necesaria para el diseño del enlace de fibra, para lo cual se crearán tres bases de datos: una para los transmisores, otra para los receptores y por último una para las fibras ópticas. En estas bases de datos se tendrán almacenados componentes de todos los fabricantes con los que trabajemos, y serán de gran utilidad a la hora de querer calcular la solución óptima.

Por otra parte tendremos que considerar todas las funciones involucradas en el análisis del enlace, como son el análisis de compatibilidad de los componentes, los balances de potencia y de tiempo, así como los distintos tipos de cálculos que podremos realizar, como son el cálculo del enlace óptimo, el de los componentes óptimos o el diseño global del enlace. Por último se añade una funcionalidad adicional que nos permite calcular el tiempo de respuesta de la fibra en el caso de que este no sea especificado por el fabricante.



## **BASE DE DATOS**



## **COMPATIBILIDAD**



## **BALANCE DE POTENCIA**



## **BALANCE DE TIEMPO**

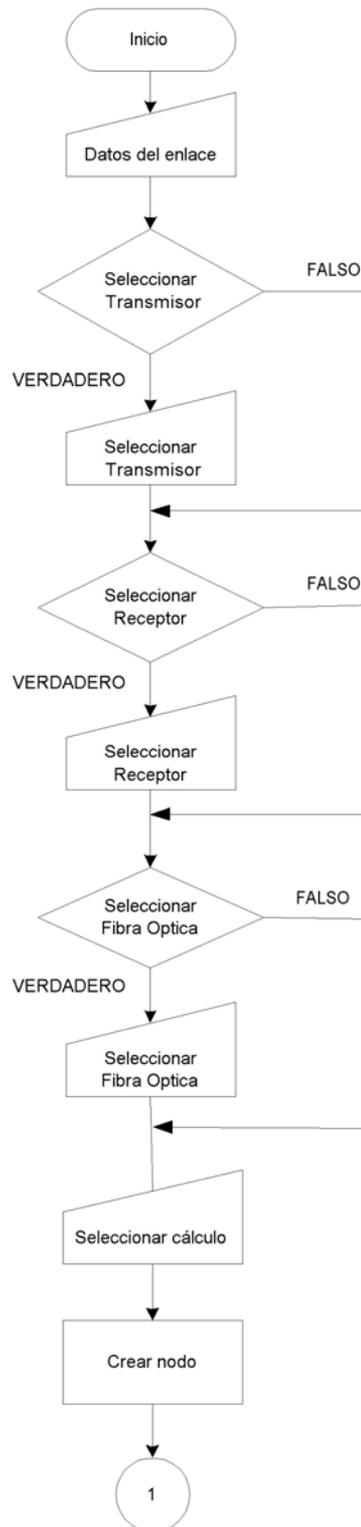


## **CALCULO DEL ENLACE ÓPTIMO**



## CALCULO DEL COMPONENTE OPTIMO

### COMPONENTE OPTIMO







## **DISEÑO DEL ENLACE**





## **TIEMPO DE SUBIDA**



## **MODULO BALANCE POTENCIA**

## **MODULO PRESUPUESTO**



## **MODULO BALANCE TIEMPO**

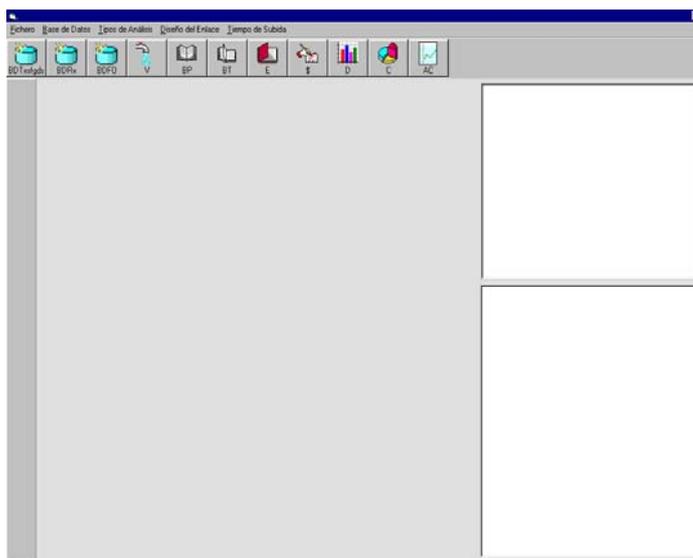


## DESARROLLO DE LA SOLUCIÓN MEDIANTE VISUAL BASIC 6.0

### PÁGINA PRINCIPAL

#### Descripción del formulario

Este formulario es el primero que aparece tras ejecutar la aplicación y es desde el que ejecutaremos las distintas funciones que tiene el programa y que se describen a continuación, para lo cual podremos hacerlo a través del interfaz de menús desplegables que aparecen en la parte superior de la pantalla o bien ejecutando cualquiera de los botones de pulsación que nos aparecen justo debajo.



**FIGURA 5.1.** Página principal desde donde podremos ejecutar las distintas funciones del programa.

La pantalla principal se completa con otras dos zonas más que constituyen el marco donde irán apareciendo los formularios de las aplicaciones a las cuales se vayan accediendo y los resultados de los cálculos realizados. La parte más a la derecha de la pantalla se reserva para el interfaz gráfico de resultados. La parte superior del mismo se ha diseñado siguiendo una estructura de árbol donde aparecerán los distintos elementos resultado de los cálculos realizados. Pinchando sobre cualquiera de estos elementos los resultados adjuntos aparecerán en la parte inferior de la interfaz gráfica de resultados.



## BASE DE DATOS EMISORES

### Descripción del formulario

Este formulario se utiliza para navegar a través de la Base de Datos creada para almacenar los Emisores Ópticos, en la cual podremos acceder a una gran variedad de Emisores de distintos fabricantes, mostrando las características de los diferentes componentes.

Tendremos la opción de modificar la Base de Datos, añadiendo o eliminando nuevos elementos. Los distintos elementos aparecerán ordenados por orden alfabético, tomando como referencia para almacenarlos el campo "Modelo", donde veremos el fabricante y el modelo según su propio catálogo.

Característica	Valor	Unidad
Modelo	laser-prueba	
Tipo de conector	SC	
Longitud de Onda Mínima	850	( nm )
Longitud de Onda Máxima	850	( nm )
Velocidad Máxima	40	( Mbps )
Potencia	0	( dBm )
Tiempo de Respuesta	0,25	( ns )
Incremento de Longitud de Onda	2	( nm )
Precio	120	( € )
Precio Circuitería Adicional	30	( € )

**FIGURA 5.2.** Formulario asociado a la Base de Datos donde están almacenados los Emisores Ópticos.

Una vez creado un nuevo elemento deberemos almacenarlo en la Base de Datos utilizando para ello el botón "Grabar" que vemos en el formulario (inhabilitado mientras no se introduzca un nuevo elemento). Este elemento será almacenado al final de la tabla donde están almacenados los distintos elementos. Para que se incluya en su posición correspondiente, según el orden alfabético tendremos que presionar el botón "Refrescar" del formulario.



Con el botón "Editar" podremos cambiar las características del elemento seleccionado (se habilita el botón "Grabar" para poder salvar los cambios realizados).

Para movernos a través de la Base de Datos, podremos hacerlo de dos maneras distintas:

- **Secuencial:** los datos serán accedidos siguiendo el orden alfabético, para lo cual se usarán los botones del formulario "Anterior" y "Siguiente" para desplazarnos en la Base de Datos. Tendremos la opción de posicionarnos en el primero de los elementos o en el último haciendo uso de los botones "Inicio" y "Final".
- **Aleatoria:** Podremos seleccionar cualquier elemento que queramos utilizando el botón "Buscar" del formulario.

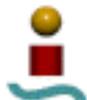
### Controles utilizados en el formulario

El título asignado al formulario en su Propiedad **Nombre** es BDEmisores. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	CARACTERISTICAS DEL EMISOR OPTICO Label21
Etiqueta	Caption Name	Modelo : Label1
Caja de Texto	Name Text DataSource DataField	Text1 (nada) Adodc1 Modelo
Etiqueta	Caption Name	Tipo de conector : Label2
Caja de Texto	Name Text DataSource DataField	Text2 (nada) Adodc1 Tipo_Conector
Etiqueta	Caption Name	Longitud de Onda Mínima : Label3
Caja de Texto	Name Text DataSource DataField	Text3 (nada) Adodc1 Longitud_Onda_Min
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Longitud de Onda Máxima : Label4
Caja de Texto	Name Text DataSource DataField	Text4 (nada) Adodc1 Longitud_Onda_Max
Etiqueta	Caption Name	( nm ) Label13
Etiqueta	Caption Name	Velocidad Máxima : Label5



Caja de Texto	Name Text DataSource DataField	Text5 (nada) Adodc1 Velocidad_Max
Etiqueta	Caption Name	( Mbps ) Label14
Etiqueta	Caption Name	Potencia : Label6
Caja de Texto	Name Text DataSource DataField	Text6 (nada) Adodc1 Potencia
Etiqueta	Caption Name	( dBm ) Label15
Etiqueta	Caption Name	Tiempo de Respuesta : Label7
Caja de Texto	Name Text DataSource DataField	Text7 (nada) Adodc1 Tiempo_Respuesta
Etiqueta	Caption Name	( ns ) Label16
Etiqueta	Caption Name	Precio : Label10
Caja de Texto	Name Text DataSource DataField	Text10 (nada) Adodc1 Precio
Etiqueta	Caption Name	( € ) Label19
Etiqueta	Caption Name	Precio Circuitería Adicional : Label11
Caja de Texto	Caption Text DataSource DataField	Text11 (nada) Adodc1 Precio_CtoAdicional
Etiqueta	Caption Name	( € ) Label20
Botón de Pulsación	Caption Name	Nuevo Nuevo
Botón de Pulsación	Caption Name	Editar Editar
Botón de Pulsación	Caption Name	Grabar Grabar
Botón de Pulsación	Caption Name	Borrar Borrar
Botón de Pulsación	Caption Name	Cancelar Cancelar
Botón de Pulsación	Caption Name	Refrescar Refrescar
Botón de Pulsación	Caption Name	Inicio Inicio

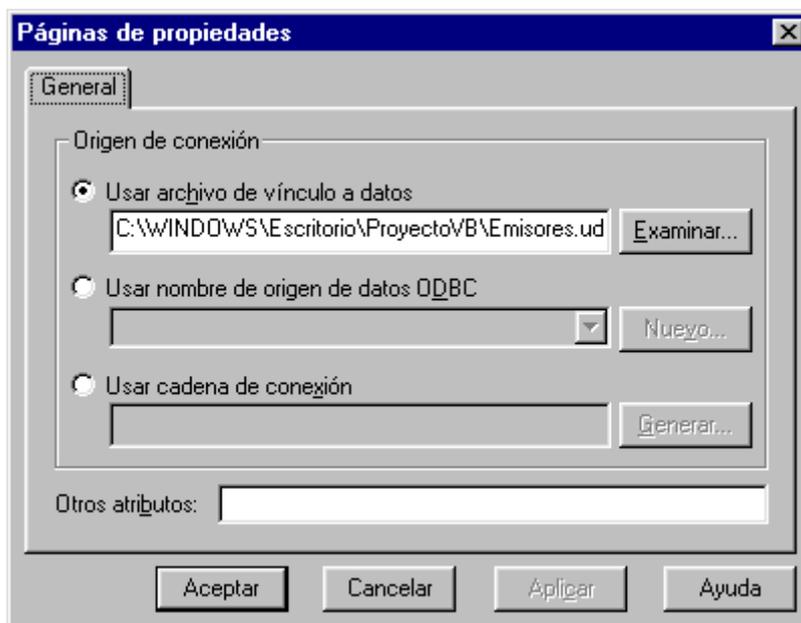


Botón de Pulsación	Caption Name	Anterior Anterior
Botón de Pulsación	Caption Name	Siguiente Siguiente
Botón de Pulsación	Caption Name	Final Final
Botón de Pulsación	Caption Name	Buscar Buscar
Control de Datos	Caption Name ConnectionString RecordSource Visible	Adodc1 Adodc1 C:\WINDOWS\Escritorio\ProyectoVB\Emisores.udl select * from Emisores order by Modelo False

**TABLA 5.1.** Tabla de controles y propiedades asociadas al formulario BDEmisores.

El control de datos (Adodc1) deberá ser vinculado con la Base de Datos correspondiente (Emisores.mdb), y enlazarlo con los controles que van a visualizar los datos de cada registro. Para ello se han utilizado las propiedades ConnectionString y RecordSource.

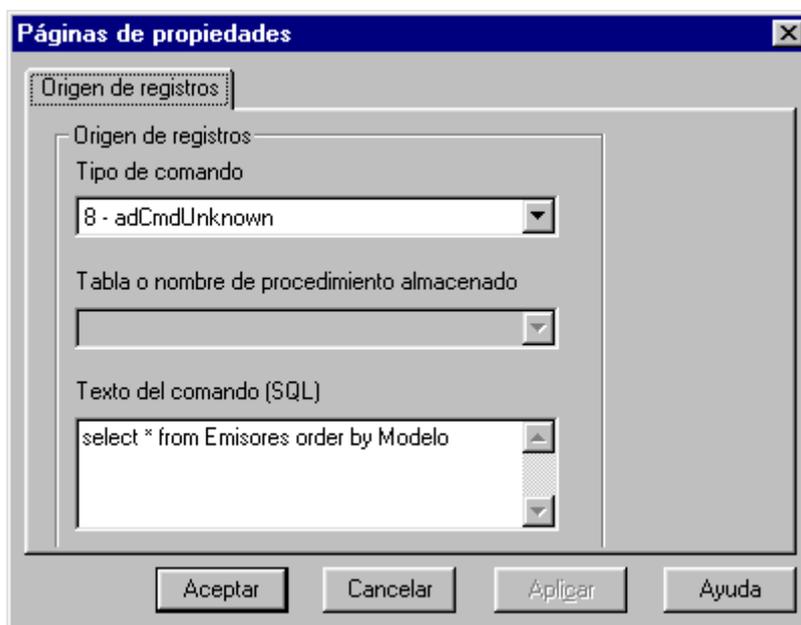
Seleccionando la propiedad **ConnectionString**, podemos especificar el fichero de vínculo a datos deseado (en nuestro caso sería Emisores.udl). Para ello deberemos pulsar el botón que hay a la derecha del control [...], y elegir en la caja de diálogo que se visualiza, *Usar archivo de vínculo a datos*. Haciendo clic en el botón *Examinar* seleccionamos el fichero .udl deseado.



**FIGURA 5.3.** Caja de Diálogo mostrada al pulsar [...] vinculado a la Propiedad ConnectionString.



Para especificar el origen de los registros de la Base de Datos a la que queremos acceder se hace mediante la propiedad **RecordSource**. Pulsando el botón de la derecha [...], deberemos escribir en la caja de texto SQL del diálogo que se visualiza, la sentencia SQL requerida para obtener el conjunto de registros deseado. Nosotros lo que haremos será seleccionarlos todos ordenados por el campo "Modelo".



**FIGURA 5.4.** Caja de Diálogo mostrada al pulsar [...] vinculado a la Propiedad RecordSource.

Por último tendremos que enlazar cada una de las cajas de texto del formulario con el control *Adodc1* para que visualice el dato correspondiente. Para ello usaremos la propiedad *DataSource*, para vincular la caja de texto y el control de datos, y la propiedad *DataField* para seleccionar el campo deseado en el registro de la Base de Datos.

El control de datos *Adodc1* permanecerá oculto al ejecutarse el formulario, y nos desplazaremos por la Base de Datos con los controles creados para tal fin.

### Descripción del código fuente

En este punto se explica el código desarrollado en la Aplicación ( que podemos ver en su totalidad en el **Apéndice I** ) para la parte del formulario BDEmisores.

Con la sentencia *Option Explicit* indicamos a Visual Basic que genere un mensaje de error siempre que encuentre una variable no declarada explícitamente. Esto se debe hacer para evitar posibles fuentes de errores, ya que Visual Basic permite el uso de variables sin ser declaradas previamente. Por lo tanto si usamos variables sin declarar, si nos equivocamos al escribir una variable, Visual no lo interpretará como un error sino como una nueva variable, lo que originará claramente que el código no funcione, al tener declaradas de forma implícita dos variables, mientras que nosotros pensamos que tenemos una sola.



Antes de explicar el código asociado a cada uno de los botones que nos permiten navegar por la Base de Datos, hay que saber que un control de datos tiene una propiedad Recordset que se corresponde con un objeto para acceso a datos denominado también Recordset.

Para posicionarnos en el primer registro de la Base de Datos usaremos el botón *Inicio* del formulario. El código asociado a ese botón se muestra a continuación:

```
Private Sub Inicio_Click( )  
    Adodc1.Recordset.MoveFirst  
End Sub
```

El evento que desarrolla el procedimiento desencadenado por el botón *Inicio* es el evento Click (al pulsar el botón se desarrolla la acción asociada). El método MoveFirst asociado al objeto Adodc1.Recordset me permite posicionarme en el primer registro.

Siguiendo un razonamiento análogo para cada uno de los otros tres botones ( *Anterior*, *Siguiente* y *Final* ), el código asociado a cada uno de ellos sería:

```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```

```
Private Sub Siguiente_Click( )  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click( )  
    Adodc1.Recordset.MoveLast  
End Sub
```

El método MovePrevious nos permite posicionarnos en el registro anterior. El método MoveNext nos permite posicionarnos en el registro siguiente. El método MoveLast nos permite posicionarnos en el último registro.

Si la condición Adodc1.Recordset.BOF es True, es porque estábamos en el primer registro e intentábamos movernos a uno anterior, por lo que nos quedamos en el primero. Si la condición Adodc1.Recordset.EOF es True, es porque estábamos en el último registro e intentábamos movernos a uno posterior, por lo que nos quedamos en el último. De no hacerlo así obtendríamos un error.

Con los botones que se describen a continuación podremos añadir y borrar registros de nuestra Base de Datos. Para hacer un uso inteligente de estos controles se deberá desarrollar previamente dos funciones que me permitan habilitar y deshabilitar los botones cuando la acción que se pretenda realizar no sea muy coherente, como por ejemplo cambiar de registro en una operación de altas, o pulsar reiteradamente el botón *Nuevo*.

Para habilitar o inhabilitar un control en general, actuaremos sobre su propiedad Enabled, asignando un valor True (habilitado) o False (inhabilitado).



Cuando el usuario referencia este formulario se encontrará el botón *Grabar* inhabilitado, ya que el registro que se visualiza inicialmente ya está grabado o está vacío. Igualmente no podrá escribir sobre las cajas de texto mientras no pulse el botón *Nuevo* o *Editar*. Todo esto queda plasmado añadiendo en el procedimiento *Form\_Load* el siguiente código:

```
Private Sub Form_Load( )
    Grabar.Enabled = False
    InhabilitarCajas
    ChDir App.Path
End Sub

Private Sub InhabilitarCajas( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```

La función *Inhabilitar\_Cajas* me permite inhabilitar todos aquellos controles que sean cajas de texto. El control *TextBox* (Caja de Texto), es un control de una matriz de controles asociada al formulario denominada *Controls*. Con la condición *TypeOf Controls(n) Is TextBox* estoy diciendo que para todo control del formulario que sea del tipo *TextBox* se realice la acción de inhabilitar dicho control: *Controls(n).Enabled = False*. La colección *Controls* tiene una propiedad, *Count*, que especifica el número de elementos de la matriz de controles (el primer elemento de la matriz es el cero).

Un razonamiento análogo nos conduce al procedimiento que permite habilitar todos los controles que sean cajas de texto:

```
Private Sub HabilitarCajas( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = True
        End If
    Next n
End Sub
```

El mismo razonamiento se seguiría para inhabilitar o habilitar botones de pulsación.

```
Private Sub InhabilitarBotones( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is CommandButton Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```



```
Private Sub HabilitarBotones()  
Dim n As Integer  
For n = 0 To Controls.Count - 1  
If TypeOf Controls(n) Is CommandButton Then  
Controls(n).Enabled = True  
End If  
Next n  
End Sub
```

Cuando el usuario pulse el botón de *Nuevo*, se deberán habilitar todas las cajas para poder escribir e inhabilitar todos los botones excepto el de *Grabar* y el de *Cancelar*. Así se evitará que el usuario se vaya a otro registro durante una operación de alta de un registro.

```
Private Sub Nuevo_Click()  
HabilitarCajas  
InhabilitarBotones  
Grabar.Enabled = True  
Cancelar.Enabled = True  
Adodc1.Recordset.AddNew  
Text1.SetFocus  
End Sub
```

El método `AddNew` crea un nuevo registro vacío que puede editar y agregar al objeto `Recordset`.

El caso del botón *Editar* sería similar al del botón *Nuevo*, utilizando el método `Edit` para editar el registro.

```
Private Sub Editar_Click()  
HabilitarCajas  
InhabilitarBotones  
Grabar.Enabled = True  
Cancelar.Enabled = True  
Text1.SetFocus  
End Sub
```

Una vez editado el registro nuevo o efectuados los cambios en los campos del registro actual, hay que utilizar el método `Update` para guardar.

```
Private Sub Grabar_Click()  
Adodc1.Recordset.Update  
HabilitarBotones  
Grabar.Enabled = False  
InhabilitarCajas  
End Sub
```

Con el botón *Cancelar* podemos corregir el alta o la actualización de un registro restaurando su contenido anterior. Para ello usaremos el control `UpdateControls`. También pondremos los controles en su estado inicial.

```
Private Sub Cancelar_Click()  
Adodc1.Recordset.CancelUpdate  
HabilitarBotones  
Grabar.Enabled = False  
InhabilitarCajas  
End Sub
```



Con el botón *Borrar* el usuario podrá eliminar el registro que está visualizando, para lo cual se requerirá una confirmación previa. Para ello utilizaremos el método Delete.

```
Private Sub Borrar_Click()  
    Dim r As Integer  
    On Error GoTo RutinaDeError  
    r = MsgBox("¿Desea borrar el registro?", vbYesNo, "Atención")  
    If r <> vbYes Then Exit Sub  
    Adodc1.Recordset.Delete  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
    Exit Sub  
RutinaDeError:  
    r = MsgBox(Error, vbOKOnly, "Se ha producido un error:")  
    Adodc1.Recordset.CancelUpdate  
End Sub
```

Con el botón *Refrescar* podemos ver el contenido actualizado de la base de datos. Por ejemplo, si la base de datos está ordenada por el nombre ( como es nuestro caso ) y añadimos un nuevo registro, con este botón podremos verlo en su sitio y no colocado al final de la base de datos. Para ello el botón *Refrescar* invoca al método Refresh.

```
Private Sub Refrescar_Click()  
    Adodc1.Recordset.Requery  
    HabilitarBotones  
    Grabar.Enabled = False  
End Sub
```

Finalmente el botón *Buscar* permitirá al usuario buscar un registro determinado, utilizando el método FindFirst y código SQL. El lenguaje SQL es un lenguaje avanzado para la consulta y modificación de bases de datos.

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo del Emisor que quiere  
        buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like '*' & Buscado & '*'"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
    ' Buscar desde el principio  
    Adodc1.Recordset.Find Criterio  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
        MsgBox ("No encuentro ese modelo")  
    End If  
End If  
End Sub
```



## BASE DE DATOS RECEPTORES

### Descripción del formulario

Este formulario se utiliza para navegar a través de la Base de Datos creada para almacenar los Receptores Ópticos, en la cual podremos acceder a una gran variedad de Receptores de distintos fabricantes, mostrando las características de los diferentes componentes.

Tendremos la opción de modificar la Base de Datos, añadiendo o eliminando nuevos elementos. Los distintos elementos aparecerán ordenados por orden alfabético, tomando como referencia para almacenarlos el campo "Modelo", donde veremos el fabricante y el modelo según su propio catálogo.

Característica	Valor	Unidad
Modelo	pin2	
Tipo de conector	SC	
Longitud de Onda Mínima	850	( nm )
Longitud de Onda Máxima	850	( nm )
Velocidad Máxima	40	( Mbps )
Sensibilidad	-41	( dBm )
Tiempo de Respuesta	5	( ns )
Precio	12	( € )
Precio Circuitería Amplificadora	9	( € )

**FIGURA 5.5.** Formulario asociado a la Base de Datos donde están almacenados los Receptores Ópticos.

Una vez creado un nuevo elemento deberemos almacenarlo en la Base de Datos utilizando para ello el botón "Grabar" que vemos en el formulario (inhabilitado mientras no se introduzca un nuevo elemento). Este elemento será almacenado al final de la tabla donde están almacenados los distintos elementos. Para que se incluya en su posición correspondiente, según el orden alfabético tendremos que presionar el botón "Refrescar" del formulario.

Con el botón "Editar" podremos cambiar las características del elemento seleccionado (se habilita el botón "Grabar" para poder salvar los cambios realizados).



Para movernos a través de la Base de Datos, podremos hacerlo de dos maneras distintas:

- **Secuencial:** los datos serán accedidos siguiendo el orden alfabético, para lo cual se usarán los botones del formulario “Anterior” y “Siguiete” para desplazarnos en la Base de Datos. Tendremos la opción de posicionarnos en el primero de los elementos o en el último haciendo uso de los botones “Inicio” y “Final”.
- **Aleatoria:** Podremos seleccionar cualquier elemento que queramos utilizando el botón “Buscar” del formulario.

### Controles utilizados en el formulario

El título asignado al formulario en su Propiedad **Nombre** es BDRreceptores. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	CARACTERISTICAS DEL RECEPTOR OPTICO Label21
Etiqueta	Caption Name	Modelo : Label1
Caja de Texto	Name Text DataSource DataField	Text1 (nada) Adodc1 Modelo
Etiqueta	Caption Name	Tipo de conector : Label2
Caja de Texto	Name Text DataSource DataField	Text2 (nada) Adodc1 Tipo_Conector
Etiqueta	Caption Name	Longitud de Onda Mínima : Label3
Caja de Texto	Name Text DataSource DataField	Text3 (nada) Adodc1 Longitud_Onda_Min
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Longitud de Onda Máxima : Label4
Caja de Texto	Name Text DataSource DataField	Text4 (nada) Adodc1 Longitud_Onda_Max
Etiqueta	Caption Name	( nm ) Label13
Etiqueta	Caption Name	Velocidad Máxima : Label5



Caja de Texto	Name Text DataSource DataField	Text5 (nada) Adodc1 Velocidad_Max
Etiqueta	Caption Name	( Mbps ) Label14
Etiqueta	Caption Name	Potencia : Label6
Caja de Texto	Name Text DataSource DataField	Text6 (nada) Adodc1 Potencia
Etiqueta	Caption Name	( dBm ) Label15
Etiqueta	Caption Name	Tiempo de Respuesta : Label7
Caja de Texto	Name Text DataSource DataField	Text7 (nada) Adodc1 Tiempo_Respuesta
Etiqueta	Caption Name	( ns ) Label16
Etiqueta	Caption Name	Precio : Label10
Caja de Texto	Name Text DataSource DataField	Text10 (nada) Adodc1 Precio
Etiqueta	Caption Name	( € ) Label19
Etiqueta	Caption Name	Precio Circuitería Amplificadora : Label11
Caja de Texto	Caption Text DataSource DataField	Text11 (nada) Adodc1 Precio_CtoAmplificador
Etiqueta	Caption Name	( € ) Label20
Botón de Pulsación	Caption Name	Nuevo Nuevo
Botón de Pulsación	Caption Name	Editar Editar
Botón de Pulsación	Caption Name	Grabar Grabar
Botón de Pulsación	Caption Name	Borrar Borrar
Botón de Pulsación	Caption Name	Cancelar Cancelar
Botón de Pulsación	Caption Name	Refrescar Refrescar
Botón de Pulsación	Caption Name	Inicio Inicio

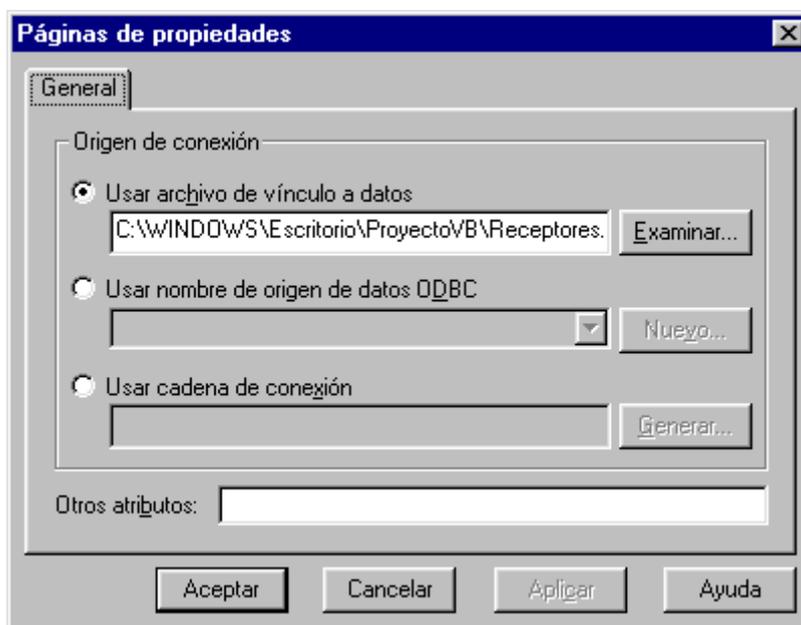


Botón de Pulsación	Caption Name	Anterior Anterior
Botón de Pulsación	Caption Name	Siguiente Siguiete
Botón de Pulsación	Caption Name	Final Final
Botón de Pulsación	Caption Name	Buscar Buscar
Control de Datos	Caption Name ConnectionString RecordSource Visible	Adodc1 Adodc1 C:\WINDOWS\Escritorio\ProyectoVB\Receptores.udl select * from es order by Modelo False

**TABLA 5.2.** Tabla de controles y propiedades asociadas al formulario BDRceptores.

El control de datos (Adodc1) deberá ser vinculado con la Base de Datos correspondiente (Receptores.mdb), y enlazarlo con los controles que van a visualizar los datos de cada registro. Para ello se han utilizado las propiedades ConnectionString y RecordSource.

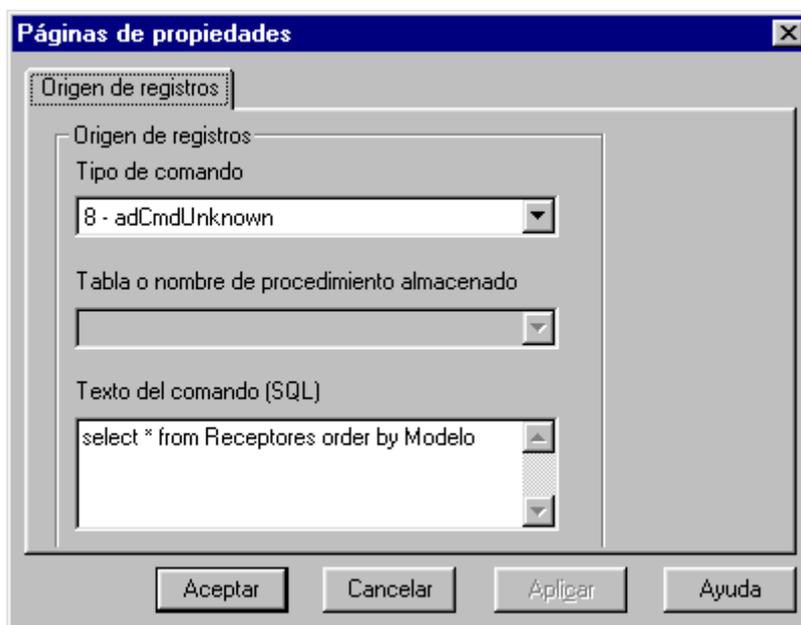
Seleccionando la propiedad **ConnectionString**, podemos especificar el fichero de vínculo a datos deseado (en nuestro caso sería Receptores.udl). Para ello deberemos pulsar el botón que hay a la derecha del control [...], y elegir en la caja de diálogo que se visualiza, *Usar archivo de vínculo a datos*. Haciendo clic en el botón *Examinar* seleccionamos el fichero .udl deseado.



**FIGURA 5.6.** Caja de Diálogo mostrada al pulsar [...] vinculado a la Propiedad ConnectionString.



Para especificar el origen de los registros de la Base de Datos a la que queremos acceder se hace mediante la propiedad **RecordSource**. Pulsando el botón de la derecha [...], deberemos escribir en la caja de texto SQL del diálogo que se visualiza, la sentencia SQL requerida para obtener el conjunto de registros deseado. Nosotros lo que haremos será seleccionarlos todos ordenados por el campo "Modelo".



**FIGURA 5.7.** Caja de Diálogo mostrada al pulsar [...] vinculado a la Propiedad RecordSource.

Por último tendremos que enlazar cada una de las cajas de texto del formulario con el control *Adodc1* para que visualice el dato correspondiente. Para ello usaremos la propiedad *DataSource*, para vincular la caja de texto y el control de datos, y la propiedad *DataField* para seleccionar el campo deseado en el registro de la Base de Datos.

El control de datos *Adodc1* permanecerá oculto al ejecutarse el formulario, y nos desplazaremos por la Base de Datos con los controles creados para tal fin.

### Descripción del código fuente

En este punto se explica el código desarrollado en la Aplicación (que podemos ver en su totalidad en el **Apéndice I**) para la parte del formulario BDRceptores.

Con la sentencia *Option Explicit* indicamos a Visual Basic que genere un mensaje de error siempre que encuentre una variable no declarada explícitamente. Esto se debe hacer para evitar posibles fuentes de errores, ya que Visual Basic permite el uso de variables sin ser declaradas previamente. Por lo tanto si usamos variables sin declarar, si nos equivocamos al escribir una variable, Visual no lo interpretará como un error sino como una nueva variable, lo que originará claramente que el código no funcione, al tener declaradas de forma implícita dos variables, mientras que nosotros pensamos que tenemos una sola.



Antes de explicar el código asociado a cada uno de los botones que nos permiten navegar por la Base de Datos, hay que saber que un control de datos tiene una propiedad Recordset que se corresponde con un objeto para acceso a datos denominado también Recordset.

Para posicionarnos en el primer registro de la Base de Datos usaremos el botón *Inicio* del formulario. El código asociado a ese botón se muestra a continuación:

```
Private Sub Inicio_Click( )
    Adodc1.Recordset.MoveFirst
End Sub
```

El evento que desarrolla el procedimiento desencadenado por el botón *Inicio* es el evento Click (al pulsar el botón se desarrolla la acción asociada). El método MoveFirst asociado al objeto Adodc1.Recordset me permite posicionarme en el primer registro.

Siguiendo un razonamiento análogo para cada uno de los otros tres botones ( *Anterior*, *Siguiente* y *Final* ), el código asociado a cada uno de ellos sería:

```
Private Sub Anterior_Click()
    Adodc1.Recordset.MovePrevious
    If Adodc1.Recordset.BOF Then
        Adodc1.Recordset.MoveFirst
    End If
End Sub
```

```
Private Sub Siguiente_Click( )
    Adodc1.Recordset.MoveNext
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
End Sub
```

```
Private Sub Final_Click( )
    Adodc1.Recordset.MoveLast
End Sub
```

El método MovePrevious nos permite posicionarnos en el registro anterior. El método MoveNext nos permite posicionarnos en el registro siguiente. El método MoveLast nos permite posicionarnos en el último registro.

Si la condición Adodc1.Recordset.BOF es True, es porque estábamos en el primer registro e intentábamos movernos a uno anterior, por lo que nos quedamos en el primero. Si la condición Adodc1.Recordset.EOF es True, es porque estábamos en el último registro e intentábamos movernos a uno posterior, por lo que nos quedamos en el último. De no hacerlo así obtendríamos un error.

Con los botones que se describen a continuación podremos añadir y borrar registros de nuestra Base de Datos. Para hacer un uso inteligente de estos controles se deberá desarrollar previamente dos funciones que me permitan habilitar y deshabilitar los botones cuando la acción que se pretenda realizar no sea muy coherente, como por ejemplo cambiar de registro en una operación de altas, o pulsar reiteradamente el botón *Nuevo*.

Para habilitar o inhabilitar un control en general, actuaremos sobre su propiedad Enabled, asignando un valor True (habilitado) o False (inhabilitado).



Cuando el usuario referencia este formulario se encontrará el botón *Grabar* inhabilitado, ya que el registro que se visualiza inicialmente ya está grabado o está vacío. Igualmente no podrá escribir sobre las cajas de texto mientras no pulse el botón *Nuevo* o *Editar*. Todo esto queda plasmado añadiendo en el procedimiento *Form\_Load* el siguiente código:

```
Private Sub Form_Load( )
    Grabar.Enabled = False
    InhabilitarCajas
    ChDir App.Path
End Sub

Private Sub InhabilitarCajas( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```

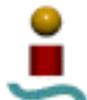
La función *Inhabilitar\_Cajas* me permite inhabilitar todos aquellos controles que sean cajas de texto. El control *TextBox* (Caja de Texto), es un control de una matriz de controles asociada al formulario denominada *Controls*. Con la condición *TypeOf Controls(n) Is TextBox* estoy diciendo que para todo control del formulario que sea del tipo *TextBox* se realice la acción de inhabilitar dicho control: *Controls(n).Enabled = False*. La colección *Controls* tiene una propiedad, *Count*, que especifica el número de elementos de la matriz de controles (el primer elemento de la matriz es el cero).

Un razonamiento análogo nos conduce al procedimiento que permite habilitar todos los controles que sean cajas de texto:

```
Private Sub HabilitarCajas( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = True
        End If
    Next n
End Sub
```

El mismo razonamiento se seguiría para inhabilitar o habilitar botones de pulsación.

```
Private Sub InhabilitarBotones( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is CommandButton Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```



```
Private Sub HabilitarBotones()  
Dim n As Integer  
For n = 0 To Controls.Count - 1  
If TypeOf Controls(n) Is CommandButton Then  
Controls(n).Enabled = True  
End If  
Next n  
End Sub
```

Cuando el usuario pulse el botón de *Nuevo*, se deberán habilitar todas las cajas para poder escribir e inhabilitar todos los botones excepto el de *Grabar* y el de *Cancelar*. Así se evitará que el usuario se vaya a otro registro durante una operación de alta de un registro.

```
Private Sub Nuevo_Click()  
HabilitarCajas  
InhabilitarBotones  
Grabar.Enabled = True  
Cancelar.Enabled = True  
Adodc1.Recordset.AddNew  
Text1.SetFocus  
End Sub
```

El método `AddNew` crea un nuevo registro vacío que puede editar y agregar al objeto `Recordset`.

El caso del botón *Editar* sería similar al del botón *Nuevo*, utilizando el método `Edit` para editar el registro.

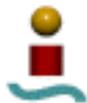
```
Private Sub Editar_Click()  
HabilitarCajas  
InhabilitarBotones  
Grabar.Enabled = True  
Cancelar.Enabled = True  
Text1.SetFocus  
End Sub
```

Una vez editado el registro nuevo o efectuados los cambios en los campos del registro actual, hay que utilizar el método `Update` para guardar.

```
Private Sub Grabar_Click()  
Adodc1.Recordset.Update  
HabilitarBotones  
Grabar.Enabled = False  
InhabilitarCajas  
End Sub
```

Con el botón *Cancelar* podemos corregir el alta o la actualización de un registro restaurando su contenido anterior. Para ello usaremos el control `UpdateControls`. También pondremos los controles en su estado inicial.

```
Private Sub Cancelar_Click()  
Adodc1.Recordset.CancelUpdate  
HabilitarBotones  
Grabar.Enabled = False  
InhabilitarCajas  
End Sub
```



Con el botón *Borrar* el usuario podrá eliminar el registro que está visualizando, para lo cual se requerirá una confirmación previa. Para ello utilizaremos el método Delete.

```
Private Sub Borrar_Click()  
    Dim r As Integer  
    On Error GoTo RutinaDeError  
    r = MsgBox("¿Desea borrar el registro?", vbYesNo, "Atención")  
    If r <> vbYes Then Exit Sub  
    Adodc1.Recordset.Delete  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
    Exit Sub  
RutinaDeError:  
    r = MsgBox(Error, vbOKOnly, "Se ha producido un error:")  
    Adodc1.Recordset.CancelUpdate  
End Sub
```

Con el botón *Refrescar* podemos ver el contenido actualizado de la base de datos. Por ejemplo, si la base de datos está ordenada por el nombre ( como es nuestro caso ) y añadimos un nuevo registro, con este botón podremos verlo en su sitio y no colocado al final de la base de datos. Para ello el botón *Refrescar* invoca al método Refresh.

```
Private Sub Refrescar_Click()  
    Adodc1.Recordset.Requery  
    HabilitarBotones  
    Grabar.Enabled = False  
End Sub
```

Finalmente el botón *Buscar* permitirá al usuario buscar un registro determinado, utilizando el método FindFirst y código SQL. El lenguaje SQL es un lenguaje avanzado para la consulta y modificación de bases de datos.

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo del Emisor que quiere  
        buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like "*" & Buscado & "*"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
    ' Buscar desde el principio  
    Adodc1.Recordset.Find Criterio  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
        MsgBox ("No encuentro ese modelo")  
    End If  
End If  
End Sub
```



## BASE DE DATOS FIBRAS OPTICAS

### Descripción del formulario

Este formulario se utiliza para navegar a través de la Base de Datos creada para almacenar las Fibras Ópticas.

Tendremos la opción de modificar la Base de Datos, añadiendo o eliminando nuevos elementos. Los distintos elementos aparecerán ordenados por orden alfabético, tomando como referencia para almacenarlos el campo "Modelo", donde veremos el fabricante y el modelo según su propio catálogo.

CARACTERISTICAS DE LA FIBRA OPTICA	
Modelo :	fibra-prueba
Tipo de Conector :	SC
Longitud de Onda Mínima :	850 ( nm )
Longitud de Onda Máxima :	850 ( nm )
Atenuación :	3,5 ( dB / Km )
Dispersión :	0,1 ( ns / nm * Km )
Precio / metro :	0 ( € )

**FIGURA 5.8.** Formulario asociado a la Base de Datos donde están almacenados las Fibras Ópticas.

Una vez creado un nuevo elemento deberemos almacenarlo en la Base de Datos utilizando para ello el botón "Grabar" que vemos en el formulario (inhabilitado mientras no se introduzca un nuevo elemento). Este elemento será almacenado al final de la tabla donde están almacenados los distintos elementos. Para que se incluya en su posición correspondiente, según el orden alfabético tendremos que presionar el botón "Refrescar" del formulario.

Con el botón "Editar" podremos cambiar las características del elemento seleccionado (se habilita el botón "Grabar" para poder salvar los cambios realizados).



Para movernos a través de la Base de Datos, podremos hacerlo de dos maneras distintas:

- **Secuencial:** los datos serán accedidos siguiendo el orden alfabético, para lo cual se usarán los botones del formulario “Anterior” y “Siguiete” para desplazarnos en la Base de Datos. Tendremos la opción de posicionarnos en el primero de los elementos o en el último haciendo uso de los botones “Inicio” y “Final”.
- **Aleatoria:** Podremos seleccionar cualquier elemento que queramos utilizando el botón “Buscar” del formulario.

### Controles utilizados en el formulario

El título asignado al formulario en su Propiedad **Nombre** es BDFibras. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	CARACTERISTICAS DE LA FIBRA OPTICA Label21
Etiqueta	Caption Name	Modelo : Label1
Caja de Texto	Name Text DataSource DataField	Modelo (nada) Adodc1 Modelo
Etiqueta	Caption Name	Tipo de conector : Label2
Caja de Texto	Name Text DataSource DataField	TipoConector (nada) Adodc1 Tipo_Conector
Etiqueta	Caption Name	Longitud de Onda Mínima : Label3
Caja de Texto	Name Text DataSource DataField	LOndaMin (nada) Adodc1 Longitud_Onda_Min
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Longitud de Onda Máxima : Label4
Caja de Texto	Name Text DataSource DataField	LOndaMax (nada) Adodc1 Longitud_Onda_Max
Etiqueta	Caption Name	( nm ) Label13
Etiqueta	Caption Name	Atenuación : Label6



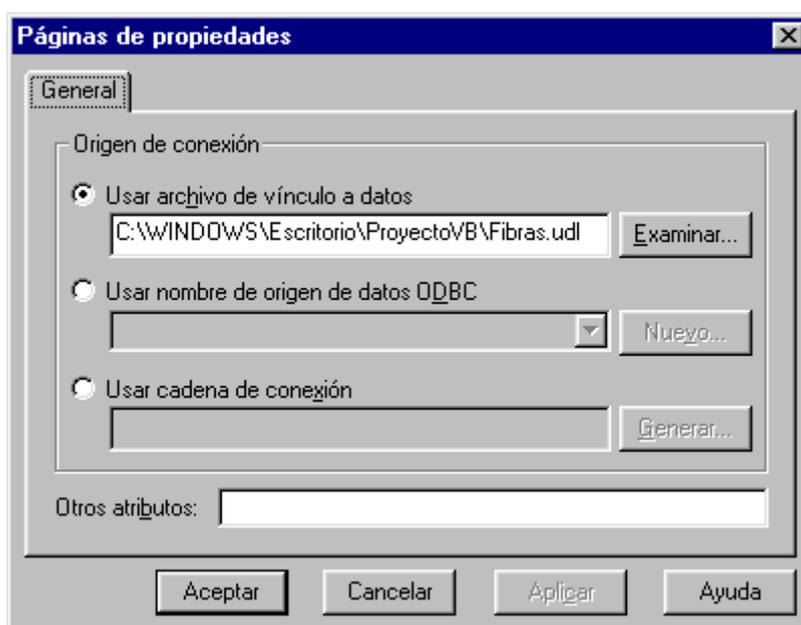
Caja de Texto	Name Text DataSource DataField	Atenuacion (nada) Adodc1 Atenuacion
Etiqueta	Caption Name	( dB / Km ) Label15
Etiqueta	Caption Name	Dispersión : Label7
Caja de Texto	Name Text DataSource DataField	Dispersion (nada) Adodc1 Dispersion
Etiqueta	Caption Name	( ns / nm * Km ) Label16
Etiqueta	Caption Name	Precio / metro : Label10
Caja de Texto	Name Text DataSource DataField	Precio (nada) Adodc1 Precio/metro
Etiqueta	Caption Name	( € ) Label19
Botón de Pulsación	Caption Name	Nuevo Nuevo
Botón de Pulsación	Caption Name	Editar Editar
Botón de Pulsación	Caption Name	Grabar Grabar
Botón de Pulsación	Caption Name	Borrar Borrar
Botón de Pulsación	Caption Name	Cancelar Cancelar
Botón de Pulsación	Caption Name	Refrescar Refrescar
Botón de Pulsación	Caption Name	Inicio Inicio
Botón de Pulsación	Caption Name	Anterior Anterior
Botón de Pulsación	Caption Name	Siguiente Siguiente
Botón de Pulsación	Caption Name	Final Final
Botón de Pulsación	Caption Name	Buscar Buscar
Control de Datos	Caption Name ConnectionString RecordSource Visible	Adodc1 Adodc1 C:\WINDOWS\Escritorio\ProyectoVB\Fibras.udl select * from es order by Modelo False

**TABLA 5.3.** Tabla de controles y propiedades asociadas al formulario BDFibras.



El control de datos (Adodc1) deberá ser vinculado con la Base de Datos correspondiente (Fibras.mdb), y enlazarlo con los controles que van a visualizar los datos de cada registro. Para ello se han utilizado las propiedades ConnectionString y RecordSource.

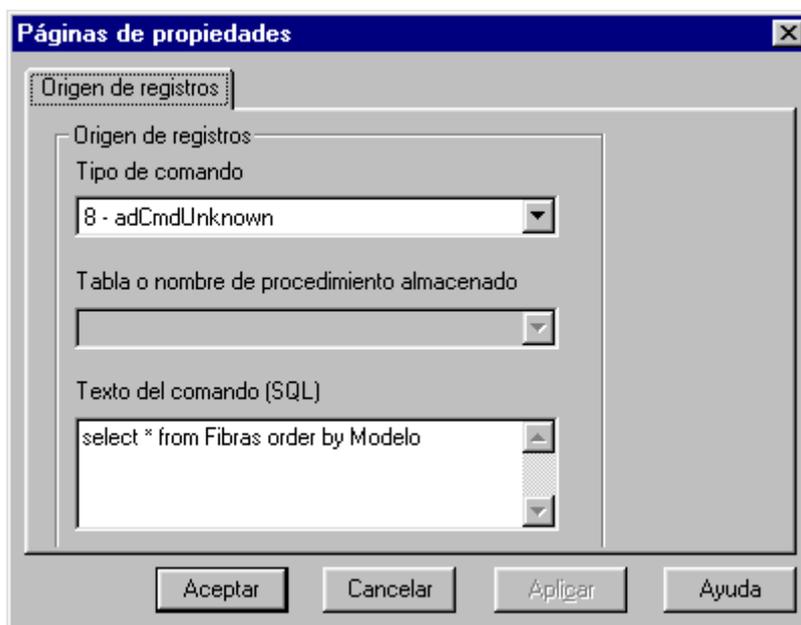
Seleccionando la propiedad **ConnectionString**, podemos especificar el fichero de vínculo a datos deseado (en nuestro caso sería Fibras.udl). Para ello deberemos pulsar el botón que hay a la derecha del control [...], y elegir en la caja de diálogo que se visualiza, *Usar archivo de vínculo a datos*. Haciendo clic en el botón *Examinar* seleccionamos el fichero .udl deseado.



**FIGURA 5.9.** Caja de Diálogo mostrada al pulsar [...] vinculado a la Propiedad ConnectionString.



Para especificar el origen de los registros de la Base de Datos a la que queremos acceder se hace mediante la propiedad **RecordSource**. Pulsando el botón de la derecha [...], deberemos escribir en la caja de texto SQL del diálogo que se visualiza, la sentencia SQL requerida para obtener el conjunto de registros deseado. Nosotros lo que haremos será seleccionarlos todos ordenados por el campo "Modelo".



**FIGURA 5.10.** Caja de Diálogo mostrada al pulsar [...] vinculado a la Propiedad RecordSource.

Por último tendremos que enlazar cada una de las cajas de texto del formulario con el control *Adodc1* para que visualice el dato correspondiente. Para ello usaremos la propiedad *DataSource*, para vincular la caja de texto y el control de datos, y la propiedad *DataField* para seleccionar el campo deseado en el registro de la Base de Datos.

El control de datos *Adodc1* permanecerá oculto al ejecutarse el formulario, y nos desplazaremos por la Base de Datos con los controles creados para tal fin.

### Descripción del código fuente

En este punto se explica el código desarrollado en la Aplicación (que podemos ver en su totalidad en el **Apéndice I**) para la parte del formulario BDFibras.

Con la sentencia `Option Explicit` indicamos a Visual Basic que genere un mensaje de error siempre que encuentre una variable no declarada explícitamente. Esto se debe hacer para evitar posibles fuentes de errores, ya que Visual Basic permite el uso de variables sin ser declaradas previamente. Por lo tanto si usamos variables sin declarar, si nos equivocamos al escribir una variable, Visual no lo interpretará como un error sino como una nueva variable, lo que originará claramente que el código no funcione, al tener declaradas de forma implícita dos variables, mientras que nosotros pensamos que tenemos una sola.



Antes de explicar el código asociado a cada uno de los botones que nos permiten navegar por la Base de Datos, hay que saber que un control de datos tiene una propiedad Recordset que se corresponde con un objeto para acceso a datos denominado también Recordset.

Para posicionarnos en el primer registro de la Base de Datos usaremos el botón *Inicio* del formulario. El código asociado a ese botón se muestra a continuación:

```
Private Sub Inicio_Click( )
    Adodc1.Recordset.MoveFirst
End Sub
```

El evento que desarrolla el procedimiento desencadenado por el botón *Inicio* es el evento Click (al pulsar el botón se desarrolla la acción asociada). El método MoveFirst asociado al objeto Adodc1.Recordset me permite posicionarme en el primer registro.

Siguiendo un razonamiento análogo para cada uno de los otros tres botones ( *Anterior*, *Siguiente* y *Final* ), el código asociado a cada uno de ellos sería:

```
Private Sub Anterior_Click()
    Adodc1.Recordset.MovePrevious
    If Adodc1.Recordset.BOF Then
        Adodc1.Recordset.MoveFirst
    End If
End Sub
```

```
Private Sub Siguiente_Click( )
    Adodc1.Recordset.MoveNext
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
End Sub
```

```
Private Sub Final_Click( )
    Adodc1.Recordset.MoveLast
End Sub
```

El método MovePrevious nos permite posicionarnos en el registro anterior. El método MoveNext nos permite posicionarnos en el registro siguiente. El método MoveLast nos permite posicionarnos en el último registro.

Si la condición Adodc1.Recordset.BOF es True, es porque estábamos en el primer registro e intentábamos movernos a uno anterior, por lo que nos quedamos en el primero. Si la condición Adodc1.Recordset.EOF es True, es porque estábamos en el último registro e intentábamos movernos a uno posterior, por lo que nos quedamos en el último. De no hacerlo así obtendríamos un error.

Con los botones que se describen a continuación podremos añadir y borrar registros de nuestra Base de Datos. Para hacer un uso inteligente de estos controles se deberá desarrollar previamente dos funciones que me permitan habilitar y deshabilitar los botones cuando la acción que se pretenda realizar no sea muy coherente, como por ejemplo cambiar de registro en una operación de altas, o pulsar reiteradamente el botón *Nuevo*.

Para habilitar o inhabilitar un control en general, actuaremos sobre su propiedad Enabled, asignando un valor True (habilitado) o False (inhabilitado).



Cuando el usuario referencia este formulario se encontrará el botón *Grabar* inhabilitado, ya que el registro que se visualiza inicialmente ya está grabado o está vacío. Igualmente no podrá escribir sobre las cajas de texto mientras no pulse el botón *Nuevo* o *Editar*. Todo esto queda plasmado añadiendo en el procedimiento *Form\_Load* el siguiente código:

```
Private Sub Form_Load( )
    Grabar.Enabled = False
    InhabilitarCajas
    ChDir App.Path
End Sub

Private Sub InhabilitarCajas( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```

La función *Inhabilitar\_Cajas* me permite inhabilitar todos aquellos controles que sean cajas de texto. El control *TextBox* (Caja de Texto), es un control de una matriz de controles asociada al formulario denominada *Controls*. Con la condición *TypeOf Controls(n) Is TextBox* estoy diciendo que para todo control del formulario que sea del tipo *TextBox* se realice la acción de inhabilitar dicho control: *Controls(n).Enabled = False*. La colección *Controls* tiene una propiedad, *Count*, que especifica el número de elementos de la matriz de controles (el primer elemento de la matriz es el cero).

Un razonamiento análogo nos conduce al procedimiento que permite habilitar todos los controles que sean cajas de texto:

```
Private Sub HabilitarCajas( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = True
        End If
    Next n
End Sub
```

El mismo razonamiento se seguiría para inhabilitar o habilitar botones de pulsación.

```
Private Sub InhabilitarBotones( )
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is CommandButton Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```



```
Private Sub HabilitarBotones()  
Dim n As Integer  
For n = 0 To Controls.Count - 1  
If TypeOf Controls(n) Is CommandButton Then  
Controls(n).Enabled = True  
End If  
Next n  
End Sub
```

Cuando el usuario pulse el botón de *Nuevo*, se deberán habilitar todas las cajas para poder escribir e inhabilitar todos los botones excepto el de *Grabar* y el de *Cancelar*. Así se evitará que el usuario se vaya a otro registro durante una operación de alta de un registro.

```
Private Sub Nuevo_Click()  
HabilitarCajas  
InhabilitarBotones  
Grabar.Enabled = True  
Cancelar.Enabled = True  
Adodc1.Recordset.AddNew  
Text1.SetFocus  
End Sub
```

El método `AddNew` crea un nuevo registro vacío que puede editar y agregar al objeto `Recordset`.

El caso del botón *Editar* sería similar al del botón *Nuevo*, utilizando el método `Edit` para editar el registro.

```
Private Sub Editar_Click()  
HabilitarCajas  
InhabilitarBotones  
Grabar.Enabled = True  
Cancelar.Enabled = True  
Text1.SetFocus  
End Sub
```

Una vez editado el registro nuevo o efectuados los cambios en los campos del registro actual, hay que utilizar el método `Update` para guardar.

```
Private Sub Grabar_Click()  
Adodc1.Recordset.Update  
HabilitarBotones  
Grabar.Enabled = False  
InhabilitarCajas  
End Sub
```

Con el botón *Cancelar* podemos corregir el alta o la actualización de un registro restaurando su contenido anterior. Para ello usaremos el control `UpdateControls`. También pondremos los controles en su estado inicial.

```
Private Sub Cancelar_Click()  
Adodc1.Recordset.CancelUpdate  
HabilitarBotones  
Grabar.Enabled = False  
InhabilitarCajas  
End Sub
```



Con el botón *Borrar* el usuario podrá eliminar el registro que está visualizando, para lo cual se requerirá una confirmación previa. Para ello utilizaremos el método Delete.

```
Private Sub Borrar_Click()  
    Dim r As Integer  
    On Error GoTo RutinaDeError  
    r = MsgBox("¿Desea borrar el registro?", vbYesNo, "Atención")  
    If r <> vbYes Then Exit Sub  
    Adodc1.Recordset.Delete  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
    Exit Sub  
RutinaDeError:  
    r = MsgBox(Error, vbOKOnly, "Se ha producido un error:")  
    Adodc1.Recordset.CancelUpdate  
End Sub
```

Con el botón *Refrescar* podemos ver el contenido actualizado de la base de datos. Por ejemplo, si la base de datos está ordenada por el nombre ( como es nuestro caso ) y añadimos un nuevo registro, con este botón podremos verlo en su sitio y no colocado al final de la base de datos. Para ello el botón *Refrescar* invoca al método Refresh.

```
Private Sub Refrescar_Click()  
    Adodc1.Recordset.Requery  
    HabilitarBotones  
    Grabar.Enabled = False  
End Sub
```

Finalmente el botón *Buscar* permitirá al usuario buscar un registro determinado, utilizando el método FindFirst y código SQL. El lenguaje SQL es un lenguaje avanzado para la consulta y modificación de bases de datos.

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo del Emisor que quiere  
        buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like '*' & Buscado & '*'"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
    ' Buscar desde el principio  
    Adodc1.Recordset.Find Criterio  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
        MsgBox ("No encuentro ese modelo")  
    End If  
End If  
End Sub
```



## COMPATIBILIDAD

### Descripción del formulario

Este formulario nos permite determinar si los elementos seleccionados para la implementación del enlace son compatibles entre sí y con los parámetros del enlace. Nos basamos para ello en dos parámetros: la longitud de onda del enlace de fibra y el tipo de conectores empleados.

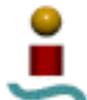
The screenshot shows a software window titled "CARACTERISTICAS DEL ENLACE DE FIBRA OPTICA". It contains several input fields and buttons. At the top, there is a field for "Longitud de Onda del Enlace de Fibra" with a unit "(nm)". Below it is a field for "Tipo de conector disponible :". A blue bar contains three tabs: "TRANSMISOR", "RECEPTOR", and "FIBRA OPTICA". Below these tabs are three dropdown menus: "Lista de transmisores", "Lista de receptores", and "Lista de fibras ópticas". At the bottom, there are two buttons: "COMPATIBILIDAD" and "GUARDAR".

**FIGURA 5.11.** Formulario empleado determinar la compatibilidad en el enlace de fibra.

En el formulario tendremos que introducir en las casillas habilitadas para ello el valor correspondiente a los parámetros anteriormente mencionados. Para seleccionar los elementos que constituyen el enlace tendremos que desplegar cada una de las listas que aparecen en el formulario y seleccionar un elemento de cada una de ellas.

Para visualizar los resultados tendremos que pulsar el botón *Compatibilidad* y se cargará un nuevo formulario con los resultados obtenidos. Podremos realizar tantos cálculos como sean necesarios, con tan sólo seleccionar nuevos elementos o modificar los parámetros y volver a pulsar el botón *Compatibilidad*. Si queremos almacenar los cálculos en un fichero, lo podremos hacer pulsando el botón *Guardar*, generándose de este modo un fichero con extensión *.vbl* que podremos recuperar con la opción *Abrir* del formulario.

Para mostrar los resultados obtenidos se ha creado un nuevo formulario que define una interfaz gráfica que se divide en dos zonas claramente diferenciadas. En la parte superior del formulario se ha colocado un Objeto **TreeView** que utilizaremos para seleccionar, mediante un sistema de nodos, los distintos cálculos que realicemos con el formulario.



En cada nodo se refleja el Transmisor-FibraOptica -Receptor empleados para la realización del cálculo, constituyendo esta la base para diferenciar cada una de las operaciones realizadas. Así en la FIGURA 5.X vemos como se han realizado dos operaciones, la primera usando como transmisor laser-prueba, como fibra óptica fibra-prueba y como receptor pin-prueba, y la segunda usando como transmisor led-prueba, como fibra óptica fibra-prueba y como receptor pin-prueba.

The screenshot shows a window titled "COMPATIBILIDAD" with a close button in the top right corner. The window is divided into two sections. The top section contains two lines of text: "laser-prueba-fibra-prueba-pinprueba" (highlighted with a blue selection box) and "led-prueba-fibra-prueba-pinprueba". The bottom section contains the following text: "Enlace punto a punto a l=850nm", "Tipo de Conector empleado SC", and "Resultando el Enlace COMPATIBLE".

**FIGURA 5.12.** Formulario empleado para mostrar los resultados obtenidos.



En la parte inferior del formulario tenemos un Objeto **Text** ( Caja de Texto ), que será donde visualizaremos los resultados de la operación realizada, seleccionada pulsando sobre el Objeto **Node** de la estructura en forma de árbol de la parte superior del formulario. En nuestro caso estamos visualizando los resultados correspondientes al primer nodo del árbol.

Un control TreeView muestra una lista jerárquica de objetos Node, cada uno de los cuales consta de una etiqueta y un mapa de bits opcional. Los controles TreeView se utilizan habitualmente para mostrar los encabezados de un documento, las entradas de un índice, los archivos y directorios de un disco o cualquier otro tipo de información que pueda ser útil ver en una estructura jerárquica.

Una vez creado un control TreeView, puede agregar, quitar, organizar y manipular de diversas formas los objetos Node si establece propiedades e invoca métodos. Puede expandir y contraer mediante programa los objetos Node para mostrar u ocultar todos los nodos secundarios. Existen tres eventos, Collapse, Expand y NodeClick, que también ofrecen funcionalidad mediante programa.

### Controles utilizados en el formulario

El título asignado al formulario en su Propiedad **Nombre** es Viabilidad. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	Longitud de Onda del Enlace de Fibra Optica : Label3
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label7
Etiqueta	Caption Name	Tipo de conector disponible : Label1
Caja de Texto	Name Text	Tipo Conector : (nada)
Etiqueta	Caption Name	TRANSMISOR Label17
Etiqueta	Caption Name	RECEPTOR Label18
Etiqueta	Caption Name	FIBRA OPTICA Label19
Lista	Name Sorted	Combo4 True
Lista	Name Sorted	Combo5 True
Lista	Name Sorted	Combo6 True



Control de Datos	Name Caption DatabaseName RecordSource Visible	Data1 Transmisor C:\WINDOWS\Escritorio\ProyectoVB\Emisores.mdb Emisores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data2 Receptor C:\WINDOWS\Escritorio\ProyectoVB\Receptores.mdb Receptores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data3 Fibra Optica C:\WINDOWS\Escritorio\ProyectoVB\Fibras.mdb Fibras False
Botón de pulsación	Caption Name	Calcular BTiempo
Botón de pulsación	Caption Name	Guardar GuardarComo

**TABLA 5.4.** Tabla de controles y propiedades asociadas al formulario Viabilidad.

En el Objeto *Lista* significamos la propiedad **Sorted**, que cuando tenga un valor **True**, nos permitirá visualizar la lista de forma ordenada.

Los controles de datos empleados para este formulario son controles DATA, en vez de controles ADO. Conceptualmente son similares. Conectan un origen de datos con controles que poseen la propiedad **DataResource** ( controles enlazados con datos ), propiedad que les habilita para conectarse al control de datos y visualizar los datos del origen de datos.

Con la propiedad *DatabaseName* se especifica el nombre del fichero base de datos o del directorio de la base de datos que se quiere conectar. El nombre de la tabla de la base de datos a la que se quiere acceder se selecciona mediante la propiedad *RecordSource*

### Descripción del código fuente

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnodeV* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad *Index*. El valor de índice del último miembro agregado es el valor de la propiedad *Count* de la colección *Nodes*.

Dim resulnodeV As Node  
Dim m As Integer



Cuando el formulario es cargado ( `Private Sub Form_Load()` ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción `Set` con una variable de tipo `Node` ( `resulnodeV` ). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad `Text`, asignándole el nombre que queremos que tenga el nodo (`COMPATIBILIDAD`).

```
Private Sub Form_Load()  
m = 0  
Set resulnodeV = ResultadosV.TreeView1.Nodes.Add()  
resulnodeV.Text = "COMPATIBILIDAD"  
End Sub
```

A la hora de introducir el valor solicitado en el formulario para la longitud de onda podríamos cometer un error, y no introducir un valor numérico. Para evitar este error se han implementado la siguiente función que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOnda_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOnda) Then  
Cancel = True  
MsgBox "La longitud de onda debe ser un valor numérico"  
End If  
End Sub
```

Las función que se muestra a continuación se utiliza para actualizar la lista de transmisores cuando esta es enfocada ( `Evento GotFocus` ). Para ello lo que se hace es cargar los elementos de la base de datos de Transmisores en la lista, para lo cual nos situamos en el primer elemento de la base de datos, y mientras no se llegue al final copiamos el campo `Modelo` en la lista de transmisores. Las dos funciones siguientes hacen lo propio para la actualización de las listas de receptores y fibras ópticas.

```
Private Sub Combo1_GotFocus()  
Data1.Recordset.MoveFirst  
While Data1.Recordset.EOF = False  
Combo1.AddItem Data1.Recordset.Fields("Modelo")  
Data1.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo2_GotFocus()  
Data2.Recordset.MoveFirst  
While Data2.Recordset.EOF = False  
Combo2.AddItem Data2.Recordset.Fields("Modelo")  
Data2.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo3_GotFocus()  
Data3.Recordset.MoveFirst  
While Data3.Recordset.EOF = False  
Combo3.AddItem Data3.Recordset.Fields("Modelo")  
Data3.Recordset.MoveNext  
Wend  
End Sub
```



La función `Viabilidad_Click()` se ejecuta cuando el usuario hace "click" sobre el botón *Compatibilidad*. Dada la extensión de esta función se irán introduciendo comentarios intercalándolos con el código fuente. Las primeras líneas de la función se corresponden con la definición de variables locales, que serán utilizadas en el entorno de la función.

```
Private Sub Viabilidad_Click()  
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String  
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer  
Dim lomintx As Integer, lominx As Integer, lominfo As Integer  
Dim LOenl As Integer  
Dim vb As Integer, causa As String
```

Para identificar el elemento seleccionado para realizar los cálculos entre los elementos de cada una de las bases de datos ( Transmisores-Receptores-Fibras Ópticas ), dispondremos de un objeto Combo, enlazado con un objeto Data, para acceder a la base de datos.

Al iniciar la función, situamos cada uno de los objetos Data al principio de la base de datos correspondiente y comprobamos que existe algún elemento en la base de datos. En este caso, y hasta que se llegue al último elemento de la base de datos comparamos cada uno de los elementos de la base de datos con el elemento que ha sido seleccionado para realizar los cálculos ( de él hasta ahora tan sólo conocemos el campo "Modelo" que nos da información acerca del nombre del elemento, pero no disponemos de información del resto de sus características hasta que lo ubiquemos en la base de datos ). Esta operación se realiza en cada una de las tres bases de datos que tenemos.

```
Data1.Recordset.MoveFirst  
Data2.Recordset.MoveFirst  
Data3.Recordset.MoveFirst  
If Data1.Recordset.RecordCount > 0 Then  
Do Until Data1.Recordset.EOF  
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then  
If Data2.Recordset.RecordCount > 0 Then  
Do Until Data2.Recordset.EOF  
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo2.Text)) Then  
If Data3.Recordset.RecordCount > 0 Then  
Do Until Data3.Recordset.EOF  
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo3.Text)) Then
```

Una vez identificados el transmisor, el receptor y la fibra óptica, asociamos el valor correspondiente a cada una de las variables definidas.

```
conectortx = Data1.Recordset.Fields("Tipo_Conector")  
conectorrx = Data2.Recordset.Fields("Tipo_Conector")  
conectorfo = Data3.Recordset.Fields("Tipo_Conector")  
conectortipo = TipoConector.Text  
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")  
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")  
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")  
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")  
lominx = Data2.Recordset.Fields("Longitud_Onda_Min")  
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")  
LOenl = LOnda.Text
```



Estas variables son pasadas como parámetro a la función *Compatibilidad* para analizar si son compatibles los elementos seleccionados con las características del enlace, basándonos para ello en dos parámetros: la longitud de onda del enlace y los tipo de conectores empleados.

```
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _  
          lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```

La función *viab* devolverá un entero que nos dirá si el enlace es o no compatible, y para el caso de no serlo la causa que hace que el enlace no sea compatible. En función del valor devuelto por la función *viab*, lo que hacemos es escribir el mensaje que se visualizará al mostrar los resultados.

```
Select Case vb  
Case 0  
    causa = "COMPATIBLE" & vbCrLf  
Case 1  
    causa = "NO COMPATIBLE" & vbCrLf & "debido a que el tipo de conector que  
admite el equipo" & vbCrLf _  
& "transmisor no es compatible" & vbCrLf  
Case 2  
    causa = "NO COMPATIBLE" & vbCrLf & "debido a que el tipo de conector que admite  
el equipo" & vbCrLf _  
& "receptor no es compatible" & vbCrLf  
Case 3  
    causa = "NO COMPATIBLE" & vbCrLf & "debido a que el tipo de conector que admite  
la" & vbCrLf _  
& "fibra óptica no es compatible" & vbCrLf  
Case 4  
    causa = "NO COMPATIBLE" & vbCrLf & "debido a que la longitud de onda del equipo"  
& vbCrLf _  
& "transmisor no es compatible" & vbCrLf  
Case 5  
    causa = "NO COMPATIBLE" & vbCrLf & "debido a que la longitud de onda del equipo"  
& vbCrLf _  
& "receptor no es compatible" & vbCrLf  
Case 6  
    causa = "NO COMPATIBLE" & vbCrLf & "debido a que la longitud de onda de la" &  
vbCrLf _  
& "fibra óptica no es compatible" & vbCrLf  
End Select
```

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

```
m = m + 1
```

Por último, tendré que almacenar el resultado obtenido en una variable tipo Cadena de Caracteres, de forma que cuando seleccione el nodo correspondiente en la estructura de árbol, se muestre en el cuadro de texto habilitado para ello el resultado correspondiente. Con este fin se ha definido una variable de carácter Público ( para que sea reconocida en todo el módulo ), como una lista de Cadena de Caracteres, de forma que cada elemento de la lista contendrá el resultado correspondiente al nodo referenciado.



```
resulV(m) = vbCrLf & _  
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _  
vbCrLf & _  
"Tipo de Conector empleado " & conectortipo & vbCrLf & _  
vbCrLf & _  
"Resultando el Enlace " & causa  
Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)  
resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
```

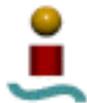
Cada nodo se identifica mediante los elementos utilizados en los cálculos ( transmisor-receptor-fibra óptica ). Cada mensaje no muestra las características del enlace y de los elementos involucrados, la fórmula sobre la que se apoya la función para realizar los cálculos, y una valoración del enlace, sugiriendo en cuánto se podría aún aumentar la distancia del enlace.

Se terminan los bucles y los condicionales abiertos al principio de la función.

```
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")  
End If  
Data3.Recordset.MoveNext  
Loop  
End If  
End If  
Data2.Recordset.MoveNext  
Loop  
End If  
End If  
Data1.Recordset.MoveNext  
Loop  
End If  
ResultadosV.Show  
End Sub
```

La función *GuardarComo* se utiliza para generar un fichero con la extensión *.vbl* que será donde almacenemos los datos empleados para la realización de un determinado cálculo.

```
Private Sub GuardarComo_Click()  
Dim path As String  
Dim puntero  
GuardarVBL.TipoSeleccion = Index  
GuardarVBL.Show vbModal  
If Trim(GuardarVBL.Retorna) <> "" Then  
path = GuardarVBL.Retorna  
End If  
puntero = FreeFile  
If path <> "" Then  
Open path For Output As #puntero  
Write #puntero, LOnda.Text, TipoConector.Text, Combo1.Text, Combo2.Text,  
Combo3.Text  
End If  
Close #puntero  
End Sub
```



El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario de Balance de Potencia.

```
Private Sub Form_Unload(Cancel As Integer)
Unload ResultadosV
End Sub
```

A continuación se describe el módulo *viab* al que llama la función *Viabilidad* al pulsar el botón *Compatibilidad*. Esta función devuelve un valor del tipo entero, en función del cual podremos determinar si la función es compatible ( *viab* = 0 ), o en el caso de no serlo, nos informará del tipo de incompatibilidad existente. La función se declara como pública ( *Public* ) para que pueda ser accedida desde cualquier formulario, ya que para todos los cálculos realizados nos interesará comprobar en primer lugar si nos conducen a un resultado implementable o no.

Se definen las variables locales a la función *viab* :

```
Public Function viab(Ctipo, Ctx, Crx, Cfo, _
LOmatx, LOmarx, LOmafo, LOmitx, LOmrx, LOmifo, LOenl) As Integer
Dim patron1 As Boolean, patron2 As Boolean, patron3 As Boolean
Dim fo1 As Boolean, fo2 As Boolean, fo3 As Boolean, fo As Boolean
Dim txfo1 As Boolean, txfo2 As Boolean, txfo3 As Boolean, txfo As Boolean
Dim rxfo1 As Boolean, rxfo2 As Boolean, rxfo3 As Boolean, rxfo As Boolean
Dim ltx As Boolean, lrx As Boolean, lfo As Boolean
Dim vbl As Integer
```

En primer lugar se analiza la compatibilidad del enlace desde el punto de vista del tipo de conectores disponibles. Para establecer la comparación establezco como patrón el tipo de conector disponible:

```
If Ctipo Like "**ST*" = True Then
patron1 = True
Else
patron1 = False
End If
If Ctipo Like "**SC*" = True Then
patron2 = True
Else
patron2 = False
End If
If Ctipo Like "**FC*" = True Then
patron3 = True
Else
patron3 = False
End If
```

Se compara con el patrón cada uno de los conectores tipo disponibles para las fibras, transmisores y receptores. El resultado de la comparación es almacenado en una variable del tipo lógico. Primero se compara el conector disponible con el tipo de conectores que pueden ser utilizados con la fibra óptica adquirida.

```
If patron1 = True Then
fo1 = Cfo Like "**ST*"
Else
fo1 = False
End If
```



```
If patron2 = True Then
  fo2 = Cfo Like "**SC*"
Else
  fo2 = False
End If
If patron3 = True Then
  fo3 = Cfo Like "**FC*"
Else
  fo3 = False
End If
```

A continuación se compara el conector disponible con el tipo de conectores que pueden ser utilizados con el transmisor óptico.

```
If patron1 = True Then
  txfo1 = Ctx Like "**ST*"
Else
  txfo1 = False
End If
If patron2 = True Then
  txfo2 = Ctx Like "**SC*"
Else
  txfo2 = False
End If
If patron3 = True Then
  txfo3 = Ctx Like "**FC*"
Else
  txfo3 = False
End If
```

Y por último se compara el conector disponible con el tipo de conectores que pueden ser utilizados con el receptor óptico.

```
If patron1 = True Then
  rxfo1 = Crx Like "**ST*"
Else
  rxfo1 = False
End If
If patron2 = True Then
  rxfo2 = Crx Like "**SC*"
Else
  rxfo2 = False
End If
If patron3 = True Then
  rxfo3 = Crx Like "**FC*"
Else
  rxfo3 = False
End If
```

Si el conector disponible coincide con el patrón la variable lógica tomará el valor verdadero.

```
If fo1 = True Or fo2 = True Or fo3 = True Then
  fo = True
```

```
Else
```

---



```
fo = False
End If
If txfo1 = True Or txfo2 = True Or txfo3 = True Then
txfo = True
Else
txfo = False
End If
If rxfo1 = True Or rxfo2 = True Or rxfo3 = True Then
rxfo = True
Else
rxfo = False
End If
```

Si el conector disponible no coincide con el patrón la variable lógica tomará el valor de falso y el enlace no será compatible desde el punto de vista del tipo de conector utilizado.

```
If fo = False Then
vbl = 1
End If
If txfo = False Then
vbl = 2
End If
If rxfo = False Then
vbl = 3
End If
```

A continuación se analiza la compatibilidad del enlace desde el punto de vista de la longitud de onda, comparándola con la longitud de onda de la fibra, del transmisor y del receptor.

```
If LOmatx = LOenl Or LOmitx = LOenl Then
Itx = True
Else
Itx = False
vbl = 4
End If
If LOMarx = LOenl Or LOMirx = LOenl Then
Irx = True
Else
Irx = False
vbl = 5
End If
If LOMafo = LOenl Or LOmifo = LOenl Then
Ifo = True
Else
Ifo = False
vbl = 6
End If
If fo = True And txfo = True And rxfo = True And Itx = True And Irx = True And
Ifo = True Then
vbl = 0
End If
viab = vbl
End Function
```



## BALANCE DE POTENCIA

### Descripción del formulario

El formulario Balance de Potencia nos informa acerca de si el enlace de fibra es implementable o no, en base a la tabulación de todas las pérdidas ( o ganancias ) de dicho enlace.

**BALANCE DE POTENCIA**

**CARACTERISTICAS DEL ENLACE DE FIBRA**

Longitud de Onda :  ( nm )      Número de Regeneradores :

Distancia :  ( Km )      Margen de Seguridad:  ( dB )

**CARACTERISTICAS DEL CONECTOR**

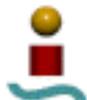
Tipo de conector :       Perdidas :  ( dB )

**TRANSMISOR      RECEPTOR      FIBRA OPTICA**

Lista de transmisores ▼       Lista de receptores ▼       Lista de fibras ópticas ▼

**FIGURA 5.13.** Formulario empleado para determeinar el balance de potencia en el enlace.



Para mostrar los resultados obtenidos se ha creado un nuevo formulario que define una interfaz gráfica que se divide en dos zonas claramente diferenciadas. En la parte superior del formulario se ha colocado un Objeto **TreeView** que utilizaremos para seleccionar, mediante un sistema de nodos, los distintos cálculos que realicemos con el formulario. En cada nodo se refleja el Transmisor-FibraOptica -Receptor empleados para la realización del cálculo, constituyendo esta la base para diferenciar cada una de las operaciones realizadas. Así en la FIGURA 5.X vemos como se han realizado dos operaciones, la primera usando como transmisor laser-prueba, como fibra óptica fibra2 y como receptor pin2, y la segunda usando como transmisor laser-prueba, como fibra óptica fibra-prueba y como receptor pin-prueba.

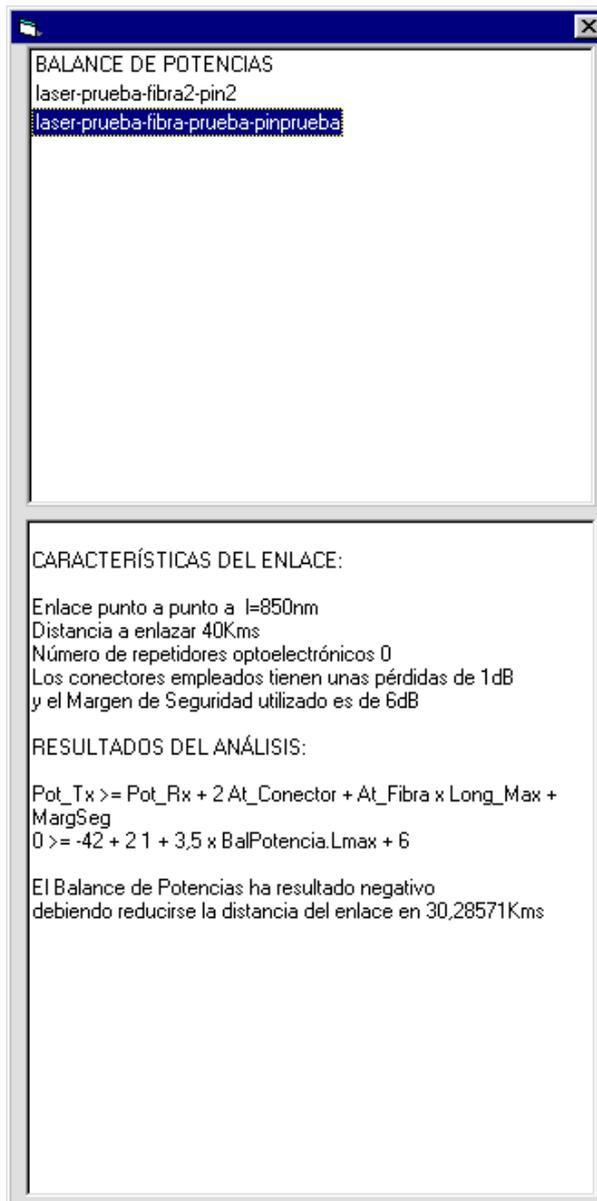


FIGURA 5.14. Formulario empleado para mostrar los resultados obtenidos.



En la parte inferior del formulario tenemos un Objeto **Text** ( Caja de Texto ), que será donde visualizaremos los resultados de la operación realizada, seleccionada pulsando sobre el Objeto **Node** de la estructura en forma de árbol de la parte superior del formulario. En nuestro caso estamos visualizando los resultados correspondientes al segundo nodo del árbol.

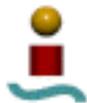
Un control TreeView muestra una lista jerárquica de objetos Node, cada uno de los cuales consta de una etiqueta y un mapa de bits opcional. Los controles TreeView se utilizan habitualmente para mostrar los encabezados de un documento, las entradas de un índice, los archivos y directorios de un disco o cualquier otro tipo de información que pueda ser útil ver en una estructura jerárquica.

Una vez creado un control TreeView, puede agregar, quitar, organizar y manipular de diversas formas los objetos Node si establece propiedades e invoca métodos. Puede expandir y contraer mediante programa los objetos Node para mostrar u ocultar todos los nodos secundarios. Existen tres eventos, Collapse, Expand y NodeClick, que también ofrecen funcionalidad mediante programa.

### Controles utilizados en el formulario

El título asignado al formulario en su Propiedad **Nombre** es BalancePotencia. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	CARACTERISTICAS DEL ENLACE DE FIBRA Label2
Etiqueta	Caption Name	Longitud de Onda : Label3
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label7
Etiqueta	Caption Name	Distancia : Label5
Caja de Texto	Name Text	Distancia (nada)
Etiqueta	Caption Name	( Km ) Label8
Etiqueta	Caption Name	Número de Regeneradores : Label6
Caja de Texto	Name Text	NRg (nada)
Etiqueta	Caption Name	Margen de Seguridad : Label10
Caja de Texto	Name Text	MargS (nada)
Etiqueta	Caption Name	( dB ) Label11
Etiqueta	Caption Name	CONECTOR Label1
Etiqueta	Caption Name	Tipo de Conector : Label4
Caja de Texto	Name Text	TipoConector (nada)
Etiqueta	Caption Name	Perdidas : Label13
Caja de Texto	Name	Perdidas



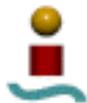
	Text	(nada)
Etiqueta	Caption Name	( dB ) Label15
Etiqueta	Caption Name	TRANSMISOR Label17
Etiqueta	Caption Name	RECEPTOR Label18
Etiqueta	Caption Name	FIBRA OPTICA Label19
Lista	Name Sorted	Combo4 True
Lista	Name Sorted	Combo5 True
Lista	Name Sorted	Combo6 True
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data1 Transmisor C:\WINDOWS\Escritorio\ProyectoVB\Emisores.mdb Emisores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data2 Receptor C:\WINDOWS\Escritorio\ProyectoVB\Receptores.mdb Receptores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data3 Fibra Optica C:\WINDOWS\Escritorio\ProyectoVB\Fibras.mdb Fibras False
Botón de pulsación	Caption Name	Calcular BTiempo
Botón de pulsación	Caption Name	Guardar GuardarComo

**TABLA 5.5.** Tabla de controles y propiedades asociadas al formulario BalancePotencia.

En el Objeto *Lista* significamos la propiedad **Sorted**, que cuando tenga un valor **True**, nos permitirá visualizar la lista de forma ordenada.

Los controles de datos empleados para este formulario son controles DATA, en vez de controles ADO. Conceptualmente son similares. Conectan un origen de datos con controles que poseen la propiedad **DataResource** ( controles enlazados con datos ), propiedad que les habilita para conectarse al control de datos y visualizar los datos del origen de datos.

Con la propiedad **DatabaseName** se especifica el nombre del fichero base de datos o del directorio de la base de datos que se quiere conectar. El nombre de la tabla de la base de datos a la que se quiere acceder se selecciona mediante la propiedad **RecordSource**.



## Descripción del código fuente

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnodeBP* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad Index. El valor de índice del último miembro agregado es el valor de la propiedad Count de la colección Nodes.

```
Dim resulnodeBP As Node
Dim m As Integer
```

Cuando el formulario es cargado ( `Private Sub Form_Load()` ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción Set con una variable de tipo Node ( *resulnodeBP* ). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad Text, asignándole el nombre que queremos que tenga el nodo ( BALANCE DE POTENCIAS ).

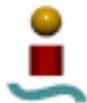
```
Private Sub Form_Load()
    m = 0
    Set resulnodeBP = ResultadosBP.TreeView1.Nodes.Add()
    resulnodeBP.Text = "BALANCE DE POTENCIAS"
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOnda_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(LOnda) Then
        Cancel = True
        MsgBox "La longitud de onda debe ser un valor numérico"
    End If
End Sub
```

```
Private Sub MargS_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(MargS) Then
        Cancel = True
        MsgBox "El margen de seguridad debe ser un valor numérico"
    End If
End Sub
```

```
Private Sub NRg_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(NRg) Then
        Cancel = True
        MsgBox "El valor numérico de regeneradores debe ser un valor numérico"
    End If
End Sub
```



```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

Las función que se muestra a continuación se utiliza para actualizar la lista de transmisores cuando esta es enfocada ( Evento GotFocus ). Para ello lo que se hace es cargar los elementos de la base de datos de Transmisores en la lista, para lo cual nos situamos en el primer elemento de la base de datos, y mientras no se llegue al final copiamos el campo Modelo en la lista de transmisores.

```
Private Sub Combo4_GotFocus()
Data1.Recordset.MoveFirst
While Data1.Recordset.EOF = False
    Combo4.AddItem Data1.Recordset.Fields("Modelo")
    Data1.Recordset.MoveNext
Wend
End Sub
```

Las dos funciones siguientes hacen lo propio para la actualización de las listas de receptores y fibras ópticas.

```
Private Sub Combo5_GotFocus()
Data2.Recordset.MoveFirst
While Data2.Recordset.EOF = False
    Combo5.AddItem Data2.Recordset.Fields("Modelo")
    Data2.Recordset.MoveNext
Wend
End Sub
```

```
Private Sub Combo6_GotFocus()
Data3.Recordset.MoveFirst
While Data3.Recordset.EOF = False
    Combo6.AddItem Data3.Recordset.Fields("Modelo")
    Data3.Recordset.MoveNext
Wend
End Sub
```



La función `BPotencia_Click()` se ejecuta cuando el usuario hace "click" sobre el botón Calcular. Dada la extensión de esta función se irán introduciendo comentarios intercalándolos con el código fuente. Las primeras líneas de la función se corresponden con la definición de variables locales, que serán utilizadas en el entorno de la función.

```
Private Sub BPotencia_Click()  
Dim balp As Boolean, ptx As Single, prx As Single, afib As Single, acon As Single  
Dim MSeg As Single, lenl As Single, lmax As Single, NRegt As Single  
Dim conectortx As String, conectorr As String, conectorfo As String  
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer  
Dim lomintx As Integer, lominx As Integer, lominfo As Integer  
Dim LOenl As Integer, conectortipo As String  
Dim vb As Integer, vbl As Boolean, causa As String
```

Para identificar el elemento seleccionado para realizar los cálculos entre los elementos de cada una de las bases de datos ( Transmisores-Receptores-Fibras Ópticas ), dispondremos de un objeto Combo, enlazado con un objeto Data, para acceder a la base de datos.

Al iniciar la función, situamos cada uno de los objetos Data al principio de la base de datos correspondiente y comprobamos que existe algún elemento en la base de datos. En este caso, y hasta que se llegue al último elemento de la base de datos comparamos cada uno de los elementos de la base de datos con el elemento que ha sido seleccionado para realizar los cálculos ( de él hasta ahora tan sólo conocemos el campo "Modelo" que nos da información acerca del nombre del elemento, pero no disponemos de información del resto de sus características hasta que lo ubiquemos en la base de datos ). Esta operación se realiza en cada una de las tres bases de datos que tenemos.

```
Data1.Recordset.MoveFirst  
Data2.Recordset.MoveFirst  
Data3.Recordset.MoveFirst  
If Data1.Recordset.RecordCount > 0 Then  
Do Until Data1.Recordset.EOF  
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo4.Text)) Then  
If Data2.Recordset.RecordCount > 0 Then  
Do Until Data2.Recordset.EOF  
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo5.Text)) Then  
If Data3.Recordset.RecordCount > 0 Then  
Do Until Data3.Recordset.EOF  
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo6.Text)) Then
```

Una vez identificados el transmisor, el receptor y la fibra óptica, asociamos el valor correspondiente a cada una de las variables definidas. El primer grupo de variables son pasadas como parámetro a la función *Compatibilidad* para analizar si son compatibles los elementos seleccionados con las características del enlace, basándonos para ello en dos parámetros: la longitud de onda del enlace y los tipo de conectores empleados.

```
conectortx = Data1.Recordset.Fields("Tipo_Conector")  
conectorr = Data2.Recordset.Fields("Tipo_Conector")  
conectorfo = Data3.Recordset.Fields("Tipo_Conector")  
conectortipo = TipoConector.Text  
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")  
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")  
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")  
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
```



```
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorr, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```

Si resultan compatibles los elementos seleccionados con las características del enlace, inicializamos cada una de las variables definidas, requeridas para los cálculos realizados por la función *BalancePotencia*.

```
If vb = 0 Then
vbl = True
Else
vbl = False
End If
If vbl = True Then
```

Necesitamos conocer la potencia del transmisor, la sensibilidad del receptor, la atenuación de la fibra óptica, distancia del enlace, el margen de seguridad, el número de nodos intermedios utilizados como regeneradores y la atenuación de los conectores empleados.

```
ptx = Data1.Recordset.Fields("Potencia")
prx = Data2.Recordset.Fields("Potencia")
afib = Data3.Recordset.Fields("Atenuacion")
lenl = Distancia.Text
MSeg = MargS.Text
NRegt = NRg.Text
acon = Perdidas.Text
```

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

```
m = m + 1
```

Calculo la distancia máxima que podrá tener el enlace con las características anteriores, y llamo a la función *BalPot* que me devolverá una variable lógica, que tomará el valor verdadero siempre y cuando la distancia del enlace sea menor o igual que la distancia máxima.

```
lmax = (ptx - (prx + 2 * acon + MSeg)) / afib
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
```

Por último, tendré que almacenar el resultado obtenido en una variable tipo Cadena de Caracteres, de forma que cuando seleccione el nodo correspondiente en la estructura de árbol, se muestre en el cuadro de texto habilitado para ello el resultado correspondiente. Con este fin se ha definido una variable de carácter Público ( para que sea reconocida en todo el módulo ), como una lista de Cadena de Caracteres, de forma que cada elemento de la lista contendrá el resultado correspondiente al nodo referenciado.

```
If balp = False Then
resulBP(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
```



```
        "Los conectores empleados tienen unas pérdidas de " & acon & "dB"
    & vbCrLf & _
        "y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf &
    _
        vbCrLf & _
        "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
        vbCrLf & _
        "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max +
MargSeg " & vbCrLf & _
        ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " &
"BalPotencia.Lmax" & " + " & MSeg & vbCrLf & _
        vbCrLf & _
        "El Balance de Potencias ha resultado negativo" & vbCrLf & _
        "debiendo reducirse la distancia del enlace en " & Abs(lmax - lenl) &
"Kms"

    Else
    resulBP(m) = vbCrLf & _
        "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
        vbCrLf & _
        "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
        "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
        "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
        "Los conectores empleados tienen unas pérdidas de " & acon & "dB"
    & vbCrLf & _
        "y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf &
    _
        vbCrLf & _
        "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
        vbCrLf & _
        "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max +
MargSeg " & vbCrLf & _
        ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " &
"BalPotencia.Lmax" & " + " & MSeg & vbCrLf & _
        vbCrLf & _
        "El Balance de Potencias ha resultado positivo" & vbCrLf & _
        "pudiendo aumentar la distancia del enlace en " & Abs(lmax - lenl) &
"Kms"
    End If
```

Cada nodo se identifica mediante los elementos utilizados en los cálculos ( transmisor-receptor-fibra óptica ). Cada mensaje no muestra las características del enlace y de los elementos involucrados, la fórmula sobre la que se apoya la función para realizar los cálculos, y una valoración del enlace, sugiriendo en cuánto se podría aún aumentar la distancia del enlace.



En el caso de no ser compatible el enlace con alguno de los elementos que lo constituyen, se mostraría un mensaje de error identificando la causa y el elemento que lo ocasiona.

```
If vbl = False Then
Select Case vb
Case 1
causa = "del tipo de conector usado para la fibra óptica"
Case 2
causa = "del tipo de conector usado para el transmisor"
Case 3
causa = "del tipo de conector usado para el receptor"
Case 4
causa = "de la longitud de onda del transmisor"
Case 5
causa = "de la longitud de onda del receptor"
Case 6
causa = "de la longitud de onda de la fibra óptica"
End Select
m = m + 1
resulBP(m) = vbCrLf & _
vbCrLf & _
"El enlace de fibra óptica no es viable debido a una incompatibilidad" &
causa
'MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad"
& causa
End If
Set resulnodeBP = ResultadosBP.TreeView1.Nodes.Add(m)
resulnodeBP.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
```

Se terminan los bucles y los condicionales abiertos al principio de la función.

```
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
```

Ejecutamos la función ResultadosBP que nos muestra los resultados obtenidos.

```
ResultadosBP.Show
End Sub
```



La función *GuardarComo* se utiliza para generar un fichero con la extensión *.bpt* que será donde almacenemos los datos empleados para la realización de un determinado cálculo.

```
Private Sub GuardarComo_Click()  
Dim path As String  
Dim puntero  
GuardarBPT.TipoSeleccion = Index  
GuardarBPT.Show vbModal  
If Trim(GuardarBPT.Retorna) <> "" Then  
    path = GuardarBPT.Retorna  
End If  
puntero = FreeFile  
Open path For Output As #puntero  
Write #puntero, LOnda.Text, Distancia.Text, NRg.Text, _  
MargS.Text, TipoConector.Text, _  
Perdidas.Text, Combo4.Text, Combo5.Text, Combo6.Text  
Close #puntero  
End Sub
```

El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario de Balance de Potencia.

```
Private Sub Form_Unload(Cancel As Integer)  
Unload ResultadosBP  
End Sub
```

Por último falta por describir el código del módulo *balpot* al que recurre la función *BalancePotencias* pasándole como argumentos la potencia del transmisor, la sensibilidad del receptor, la atenuación de la fibra y de los conectores, el margen de seguridad, la distancia del enlace y el número de regeneradores intermedios y que devuelve un valor lógico que será verdadero si se verifica el balance de potencias según la ecuación:

$$Pot_{Tx} \geq Pot_{Rx} + 2 * \alpha_c + \alpha_{fo} * L_{enl} + MS$$

```
Public Function balpot(ptx As Single, prx As Single, afib As Single, _  
    acon As Single, MS As Single, lenl As Single, NReg As Single) As Boolean  
Dim lmax As Single  
lmax = (ptx - (prx + 2 * acon + MS)) / afib  
If lmax > (lenl / (1 + NReg)) Then  
    balpot = True  
Else  
    balpot = False  
End If  
End Function
```



## BALANCE DE TIEMPO

### Descripción del formulario

El formulario Balance de Tiempo nos informa acerca de si el enlace de fibra es implementable o no, en base a la respuesta temporal de los elementos que componen de dicho enlace.

**BALANCE DE TIEMPO**

**CARACTERISTICAS DEL ENLACE DE FIBRA**

Longitud de Onda :  ( nm )      Distancia :  ( Kms )

Velocidad :  ( Mbps )      Número de Regeneradores :

Tipo de conector disponible :       Código  NRZ  RZ

**TRANSMISOR**      **RECEPTOR**      **FIBRA OPTICA**

**CALCULAR**      **GUARDAR**

**FIGURA 5.15.** Formulario empleado para determinar el balance de tiempo en el enlace.

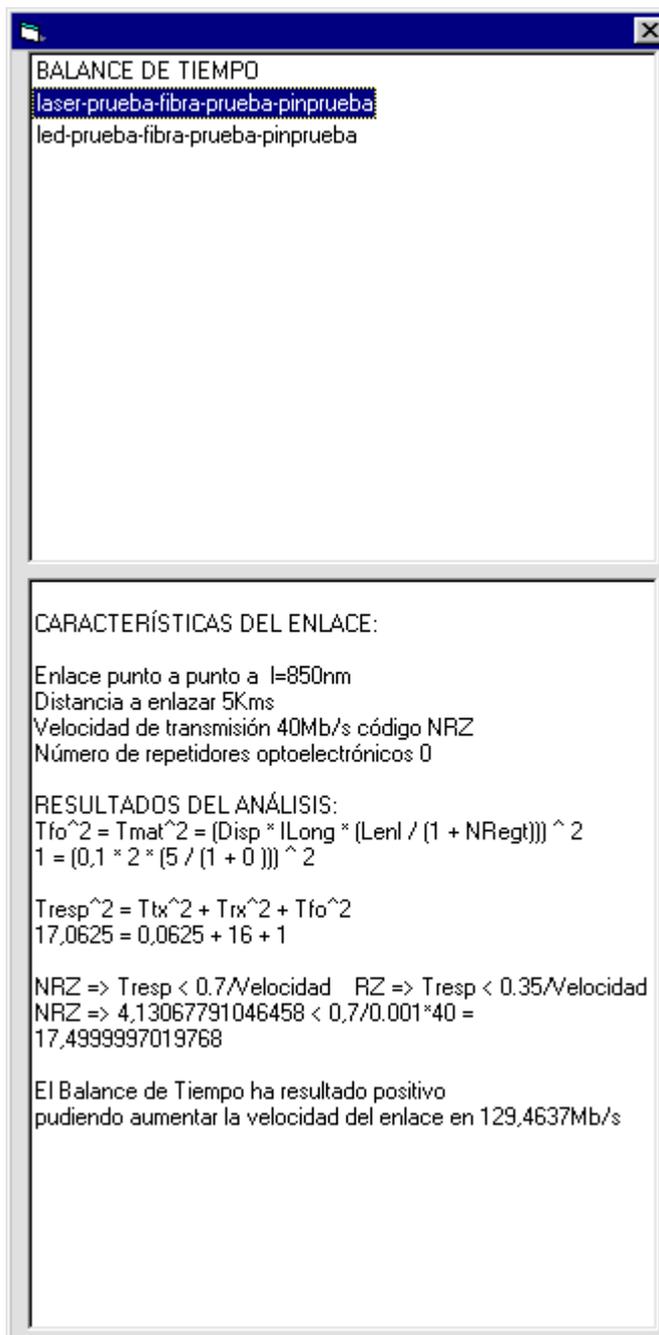


FIGURA 5.16. Formulario empleado para mostrar los resultados obtenidos.



### Controles utilizados en el formulario

El título asignado al formulario en su Propiedad **Nombre** es BalanceTiempo. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	CARACTERISTICAS DEL ENLACE DE FIBRA Label2
Etiqueta	Caption Name	Longitud de Onda : Label3
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label7
Etiqueta	Caption Name	Velocidad : Label4
Caja de Texto	Name Text	Velocidad (nada)
Etiqueta	Caption Name	( Mbps ) Label9
Etiqueta	Caption Name	Distancia : Label1
Caja de Texto	Name Text	Distancia (nada)
Etiqueta	Caption Name	( Kms ) Label5
Etiqueta	Caption Name	Número de Regeneradores : Label6
Caja de Texto	Name Text	NRg (nada)
Etiqueta	Caption Name	Tipo de Conector disponible : Label8
Caja de Texto	Name Text	TipoConector (nada)
Etiqueta	Caption Name	Código : Label12
	Name Text	NRZ (nada)
	Caption Name	RZ Label13
Etiqueta	Caption Name	TRANSMISOR Label17
Etiqueta	Caption Name	RECEPTOR Label18
Etiqueta	Caption Name	FIBRA OPTICA Label19



Lista	Name Sorted	Combo4 True
Lista	Name Sorted	Combo5 True
Lista	Name Sorted	Combo6 True
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data1 Transmisor C:\WINDOWS\Escritorio\ProyectoVB\Emisores.mdb Emisores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data2 Receptor C:\WINDOWS\Escritorio\ProyectoVB\Receptores.mdb Receptores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data3 Fibra Optica C:\WINDOWS\Escritorio\ProyectoVB\Fibras.mdb Fibras False
Botón de pulsación	Caption Name	Calcular BTiempo
Botón de pulsación	Caption Name	Guardar GuardarComo

**TABLA 5.6.** Tabla de controles y propiedades asociadas al formulario BalanceTiempo.

En el Objeto *Lista* significamos la propiedad **Sorted**, que cuando tenga un valor **True**, nos permitirá visualizar la lista de forma ordenada.

Los controles de datos empleados para este formulario son controles DATA, en vez de controles ADO. Conceptualmente son similares. Conectan un origen de datos con controles que poseen la propiedad **DataResource** ( controles enlazados con datos ), propiedad que les habilita para conectarse al control de datos y visualizar los datos del origen de datos.

Con la propiedad **DatabaseName** se especifica el nombre del fichero base de datos o del directorio de la base de datos que se quiere conectar. El nombre de la tabla de la base de datos a la que se quiere acceder se selecciona mediante la propiedad **RecordSource**.

### Descripción del código fuente

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnodeBT* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad Index. El valor de índice del último miembro agregado es el valor de la propiedad Count de la colección Nodes.

Dim resulnodeBT As Node  
Dim m As Integer



Cuando el formulario es cargado ( Private Sub Form\_Load() ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción Set con una variable de tipo Node ( *resulnodeBT* ). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad Text, asignándole el nombre que queremos que tenga el nodo ( BALANCE DE TIEMPO ).

```
Private Sub Form_Load()  
m = 0  
Set resulnodeBT = ResultadosBT.TreeView1.Nodes.Add()  
resulnodeBT.Text = "BALANCE DE TIEMPO"  
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOnda_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOnda) Then  
Cancel = True  
MsgBox "La longitud de onda debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub NRg_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(NRg) Then  
Cancel = True  
MsgBox "El valor numérico de regeneradores debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Distancia_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Distancia) Then  
Cancel = True  
MsgBox "La distancia del enlace debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Velocidad) Then  
Cancel = True  
MsgBox "La velocidad debe ser un valor numérico"  
End If  
End Sub
```

Las función que se muestra a continuación se utiliza para actualizar la lista de transmisores cuando esta es enfocada ( Evento GotFocus ).



Para ello lo que se hace es cargar los elementos de la base de datos de Transmisores en la lista, para lo cual nos situamos en el primer elemento de la base de datos, y mientras no se llegue al final copiamos el campo Modelo en la lista de transmisores. Las dos funciones siguientes hacen lo propio para la actualización de las listas de receptores y fibras ópticas.

```
Private Sub Combo4_GotFocus()  
Data1.Recordset.MoveFirst  
While Data1.Recordset.EOF = False  
    Combo4.AddItem Data1.Recordset.Fields("Modelo")  
    Data1.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo2_GotFocus()  
Data2.Recordset.MoveFirst  
While Data2.Recordset.EOF = False  
    Combo2.AddItem Data2.Recordset.Fields("Modelo")  
    Data2.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo3_GotFocus()  
Data3.Recordset.MoveFirst  
While Data3.Recordset.EOF = False  
    Combo3.AddItem Data3.Recordset.Fields("Modelo")  
    Data3.Recordset.MoveNext  
Wend  
End Sub
```

La función BTiempo\_Click() se ejecuta cuando el usuario hace "click" sobre el botón Calcular. Dada la extensión de esta función se irán introduciendo comentarios intercalándolos con el código fuente. Las primeras líneas de la función se corresponden con la definición de variables locales, que serán utilizadas en el entorno de la función.

```
Private Sub BTiempo_Click()  
Dim balt As Boolean  
Dim lenl As Single, NRegt As Single, vel As Single, restvel As Single  
Dim cod As Boolean, cdg As Single, cogigo As String  
Dim disp As Single, ilong As Single, ttx As Single, trx As Single  
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String  
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer  
Dim lomintx As Integer, lominrx As Integer, lominfo As Integer  
Dim LOenl As Integer  
Dim vb As Integer, vbl As Boolean, causa As String  
Dim TmatCuad As Single, TresCuad As Single, Tres As Single
```

Para identificar el elemento seleccionado para realizar los cálculos entre los elementos de cada una de las bases de datos ( Transmisores-Receptores-Fibras Ópticas ), dispondremos de un objeto Combo, enlazado con un objeto Data, para acceder a la base de datos.



Al iniciar la función, situamos cada uno de los objetos Data al principio de la base de datos correspondiente y comprobamos que existe algún elemento en la base de datos. En este caso, y hasta que se llegue al último elemento de la base de datos comparamos cada uno de los elementos de la base de datos con el elemento que ha sido seleccionado para realizar los cálculos ( de él hasta ahora tan sólo conocemos el campo "Modelo" que nos da información acerca del nombre del elemento, pero no disponemos de información del resto de sus características hasta que lo ubiquemos en la base de datos ). Esta operación se realiza en cada una de las tres bases de datos que tenemos.

```
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo4.Text)) Then
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo5.Text)) Then
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo6.Text)) Then
```

Una vez identificados el transmisor, el receptor y la fibra óptica, asociamos el valor correspondiente a cada una de las variables definidas. El primer grupo de variables son pasadas como parámetro a la función *Compatibilidad* para analizar si son compatibles los elementos seleccionados con las características del enlace, basándonos para ello en dos parámetros: la longitud de onda del enlace y los tipo de conectores empleados.

```
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```

Si resultan compatibles los elementos seleccionados con las características del enlace, inicializamos cada una de las variables definidas, requeridas para los cálculos realizados por la función *BalanceTiempos*.

Necesitamos conocer el tiempo de subida del transmisor, el tiempo de subida del receptor, el incremento de longitud de onda del transmisor, la dispersión de la fibra óptica, la velocidad de transmisión, el número de nodos intermedios utilizados como regeneradores y el tipo de código utilizado.

```
If vb = 0 Then
tx = Data1.Recordset.Fields("Tiempo_Respuesta")
trx = Data2.Recordset.Fields("Tiempo_Respuesta")
```



```
ilong = Data1.Recordset.Fields("Incremento_LOnda")
disp = Data3.Recordset.Fields("Dispersion")
vel = Velocidad.Text
lenl = Distancia.Text
NRegt = NRg.Text
If NRZ.Value = True Then
    cod = False
Else
    cod = True
End If
```

Para evaluar el balance de tiempos se aplicará la ecuación de evaluación del tiempo de subida, que nos dice que el tiempo de subida del sistema al cuadrado, expresado en nanosegundos, lo podemos calcular como la suma de los cuadrados de los tiempos de subida del generador, del receptor y de la fibra.

En función del tipo de código empleado, el tiempo de subida multiplicado por la velocidad de transmisión deberá ser menor que un determinado coeficiente:

```
RZ => Tr x B < 0.35    NRZ => Tr x B < 0.7

TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
Tres = Sqr(TresCuad)
If cod = False Then
    cdg = 0.7
    codigo = "NRZ"
Else
    cdg = 0.35
    codigo = "RZ"
End If
restvel = Abs(vel - (cdg / (0.001 * Tres)))
```

En el parámetro *restvel* se refleja el incremento de la velocidad de transmisión posible para que el enlace siga siendo implementable desde el punto de vista del balance de tiempo.

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

```
m = m + 1
```

Llamo a la función *BalanceTiempo* que me devolverá una variable lógica, que tomará el valor verdadero siempre y cuando se cumpla la condición expresada anteriormente .

```
balt = baltime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)
```

Por último, tendré que almacenar el resultado obtenido en una variable tipo Cadena de Caracteres, de forma que cuando seleccione el nodo correspondiente en la estructura de árbol, se muestre en el cuadro de texto habilitado para ello el resultado correspondiente. Con este fin se ha definido una variable de carácter Público ( para que sea reconocida en todo el módulo ), como una lista de Cadena de Caracteres, de forma que cada elemento de la lista contendrá el resultado correspondiente al nodo referenciado.

```
If balt = False Then
    resulBT(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
```



```
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
"Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
TmatCuad & " = (" & disp & "*" & ilong & "*" & lenl & " / (1 + " & NRegt & " )) ^ 2" &
vbCrLf & _
vbCrLf & _
"Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
TresCuad & " = " & (tx ^ 2) & " + " & (rx ^ 2) & " + " & TmatCuad & vbCrLf & _
vbCrLf & _
"NRZ => Tresp < 0.7/Velocidad  RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
codigo & " => " & Sqr(TresCuad) & " < " & cdg & "/0.001*" & vel & " = " &
cdg / (vel * 0.001) & vbCrLf & _
vbCrLf & _
"El Balance de Tiempo ha resultado negativo" & vbCrLf & _
"debiendo reducirse la velocidad del enlace en " & restvel & "Mb/s"
Else
resulBT(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
"Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
TmatCuad & " = (" & disp & "*" & ilong & "*" & lenl & " / (1 + " & NRegt & " )) ^ 2" &
& vbCrLf & _
vbCrLf & _
"Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
TresCuad & " = " & (tx ^ 2) & " + " & (rx ^ 2) & " + " & TmatCuad & vbCrLf & _
vbCrLf & _
"NRZ => Tresp < 0.7/Velocidad  RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
codigo & " => " & Sqr(TresCuad) & " < " & cdg & "/0.001*" & vel & " = " &
& cdg / (vel * 0.001) & vbCrLf & _
vbCrLf & _
"El Balance de Tiempo ha resultado positivo" & vbCrLf & _
"pudiendo aumentar la velocidad del enlace en " & restvel & "Mb/s"
End If
```

Cada nodo se identifica mediante los elementos utilizados en los cálculos ( transmisor-receptor-fibra óptica ). Cada mensaje no muestra las características del enlace y de los elementos involucrados, la fórmula sobre la que se apoya la función para realizar los cálculos, y una valoración del enlace, sugiriendo en cuánto se podría aún aumentar la velocidad de transmisión.



En el caso de no ser compatible el enlace con alguno de los elementos que lo constituyen, se mostraría un mensaje de error identificando la causa y el elemento que lo ocasiona.

```
If vbl = False Then
Select Case vb
Case 1
causa = "del tipo de conector usado para la fibra óptica"
Case 2
causa = "del tipo de conector usado para el transmisor"
Case 3
causa = "del tipo de conector usado para el receptor"
Case 4
causa = "de la longitud de onda del transmisor"
Case 5
causa = "de la longitud de onda del receptor"
Case 6
causa = "de la longitud de onda de la fibra óptica"
End Select
m = m + 1
resulBT(m) = vbCrLf & _
vbCrLf & _
"El enlace de fibra óptica no es viable debido a una incompatibilidad" &
causa
'MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad"
& causa
End If
Set resulnodeBT = ResultadosBT.TreeView1.Nodes.Add(m)
resulnodeBT.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
```

Se terminan los bucles y los condicionales abiertos al principio de la función.

```
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
```

Ejecutamos la función ResultadosBT que nos muestra los resultados obtenidos.

```
ResultadosBT.Show
End Sub
```



La función *GuardarComo* se utiliza para generar un fichero con la extensión *.btp* que será donde almacenemos los datos empleados para la realización de un determinado cálculo.

```
Private Sub GuardarComo_Click()  
Dim path As String  
Dim puntero  
GuardarBTP.TipoSeleccion = Index  
GuardarBTP.Show vbModal  
If Trim(GuardarBTP.Retorna) <> "" Then  
    path = GuardarBTP.Retorna  
End If  
puntero = FreeFile  
If path <> "" Then  
    Open path For Output As #puntero  
    Write #puntero, LOnda.Text, Velocidad.Text, Distancia.Text, NRg.Text,  
    TipoConector.Text, _  
    NRZ.Value, Combo4.Text, Combo2.Text, Combo3.Text  
End If  
Close #puntero  
End Sub
```

El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario de Balance de Potencia.

```
Private Sub Form_Unload(Cancel As Integer)  
    Unload ResultadosBT  
End Sub
```

Por último falta por describir el código del módulo *baltime* al que recurre la función *BalanceTiempos* pasándole como argumentos el tiempo de subida del transmisor, el tiempo de subida del receptor, la velocidad de transmisión, la distancia del enlace, el número de regeneradores intermedios, la dispersión de la fibra, el incremento de longitud de onda del transmisor y el tipo de codificación empleada y que devuelve un valor lógico que será verdadero si se verifica el balance de tiempos.

```
Public Function baltime(disp As Single, lOnd As Single, lenl As Single, _  
    ttx As Single, trx As Single, vel As Single, cod As Boolean, _  
    NReg As Single) As Boolean  
    Dim Tres As Single, TresCuad As Double, TmatCuad As Double, c As Single  
    TmatCuad = (disp * lOnd * (lenl / (1 + NReg))) ^ 2  
    TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)  
    Tres = Sqr(TresCuad)  
    If cod = False Then  
        c = 0.7  
    Else  
        c = 0.35  
    End If  
    If (Tres * vel * 0.001) < c Then  
        baltime = True  
    Else  
        baltime = False  
    End If  
End Function
```



## ANÁLISIS DEL ENLACE DE FIBRA ÓPTICA

### Descripción del formulario

Este formulario nos permitirá, conocidas todas las características del enlace de fibra, obtener una información completa del mismo. De este modo, podremos conocer si el enlace es o no implementable, si podemos ajustar algún parámetro consiguiendo optimizar su comportamiento y tener una idea de que coste asociado tiene su realización.

**CARACTERÍSTICAS DEL ENLACE DE FIBRA OPTICA**

**CARACTERÍSTICAS DEL ENLACE**

Longitud de Onda :  ( nm    Velocidad :  ( Mbps )

Distancia :  ( Km )    Margen de Seguridad:  ( dB )

Número de Regeneradores :

**CARACTERÍSTICAS DEL CONECTOR**

Tipo :  Perdidas :  ( dB ) Precio :  )

**Código**     NRZ     RZ

**TRANSMISOR    RECEPTOR    FIBRA OPTICA**

Lista de transmisores     Lista de receptores     Lista de fibras ópticas

**CALCULAR    GUARDAR**

**FIGURA 5.17.** Formulario empleado para el análisis del enlace de fibra óptica.

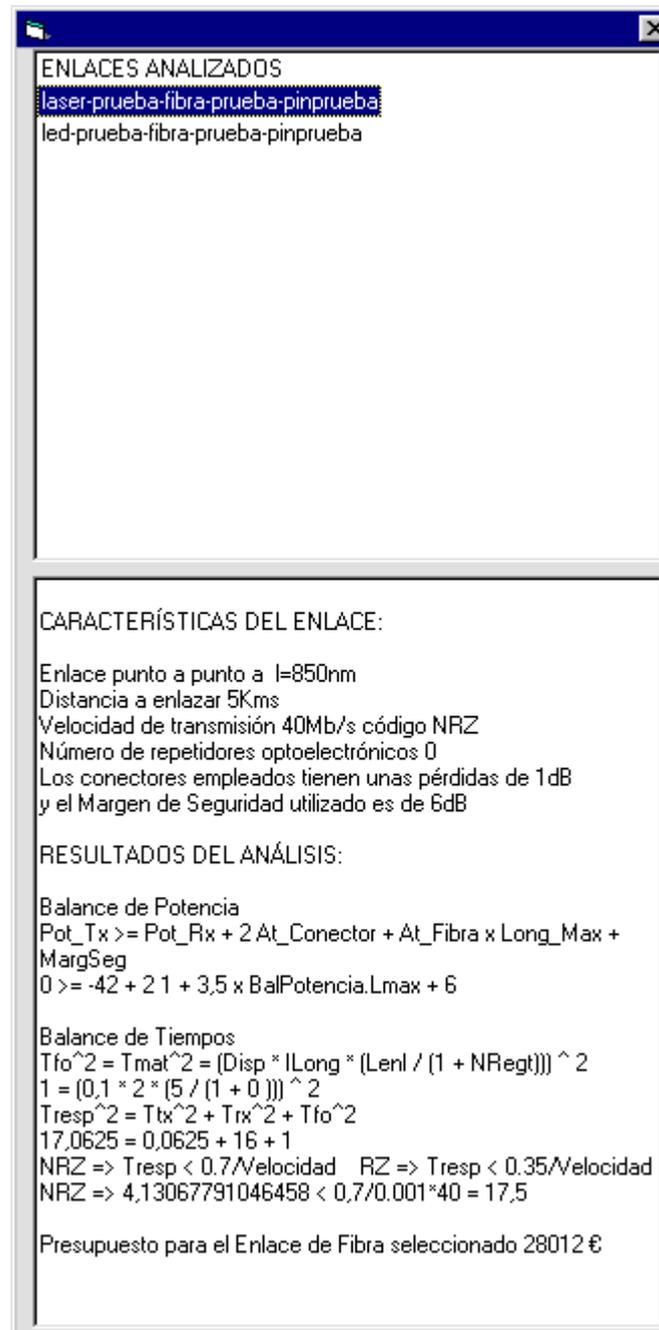


FIGURA 5.18. Formulario empleado para mostrar los resultados obtenidos.



### Controles utilizados en el formulario Análisis del Enlace de Fibra Óptica

El título asignado al formulario en su Propiedad **Nombre** es Enlace. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	ENLACE Label2
Etiqueta	Caption Name	Longitud de Onda : Label3
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label7
Etiqueta	Caption Name	Velocidad : Label4
Caja de Texto	Name Text	Velocidad (nada)
Etiqueta	Caption Name	( Mbps ) Label9
Etiqueta	Caption Name	Distancia : Label5
Caja de Texto	Name Text	Distancia (nada)
Etiqueta	Caption Name	( Kms ) Label8
Etiqueta	Caption Name	Número de Regeneradores : Label6
Caja de Texto	Name Text	NRg (nada)
Etiqueta	Caption Name	Margen de Seguridad : Label10
Caja de Texto	Name Text	MargS (nada)
Etiqueta	Caption Name	( dB ) Label11
Etiqueta	Caption Name	CONECTOR Label1
Etiqueta	Caption Name	Tipo : Label20
Caja de Texto	Name Text	TipoConector (nada)
Etiqueta	Caption Name	Pérdidas : Label13
Caja de Texto	Name Text	Perdidas (nada)
Etiqueta	Caption Name	( dB ) Label15
Etiqueta	Caption Name	Precio : Label14
Caja de Texto	Name Text	Precio (nada)



Etiqueta	Caption Name	( € ) Label16
Etiqueta	Caption Name	Código : Label12
	Name Text	NRZ (nada)
	Name Text	RZ Label13
Etiqueta	Caption Name	TRANSMISOR Label17
Etiqueta	Caption Name	RECEPTOR Label18
Etiqueta	Caption Name	FIBRA OPTICA Label19
Lista	Name Sorted	Combo1 True
Lista	Name Sorted	Combo2 True
Lista	Name Sorted	Combo3 True
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data1 Transmisor C:\WINDOWS\Escritorio\ProyectoVB\Emisores.mdb Emisores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data2 Receptor C:\WINDOWS\Escritorio\ProyectoVB\Receptores.mdb Receptores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data3 Fibra Optica C:\WINDOWS\Escritorio\ProyectoVB\Fibras.mdb Fibras False
Botón de pulsación	Caption Name	CALCULAR Command1
Botón de pulsación	Caption Name	GUARDAR GuardarComo

**TABLA 5.7.** Tabla de controles y propiedades asociadas al formulario Enlace.

En el Objeto *Lista* significamos la propiedad **Sorted**, que cuando tenga un valor **True**, nos permitirá visualizar la lista de forma ordenada.

Los controles de datos empleados para este formulario son controles DATA, en vez de controles ADO. Conceptualmente son similares. Conectan un origen de datos con controles que poseen la propiedad **DataResource** ( controles enlazados con datos ), propiedad que les habilita para conectarse al control de datos y visualizar los datos del origen de datos.

Con la propiedad **DatabaseName** se especifica el nombre del fichero base de datos o del directorio de la base de datos que se quiere conectar. El nombre de la tabla de la base de datos a la que se quiere acceder se selecciona mediante la propiedad **RecordSource**.



## Descripción del código fuente

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnodeEnl* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad Index. El valor de índice del último miembro agregado es el valor de la propiedad Count de la colección Nodes.

```
Dim resulnodeEnl As Node
Dim m As Integer
```

Cuando el formulario es cargado ( Private Sub Form\_Load() ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción Set con una variable de tipo Node ( *resulnodeEnl* ). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad Text, asignándole el nombre que queremos que tenga el nodo (ENLACES ANALIZADOS).

```
Private Sub Form_Load()
m = 0
Set resulnodeEnl = ResultadosEnl.TreeView1.Nodes.Add()
resulnodeEnl.Text = "ENLACES ANALIZADOS"
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
Cancel = True
MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```

```
Private Sub MargS_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(MargS) Then
Cancel = True
MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub NRg_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(NRg) Then
Cancel = True
MsgBox "El valor numérico de regeneradores debe ser un valor numérico"
End If
End Sub
```



```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub

Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub

Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub

Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```

Las función que se muestra a continuación se utiliza para actualizar la lista de transmisores cuando esta es enfocada ( Evento GotFocus ). Para ello lo que se hace es cargar los elementos de la base de datos de Transmisores en la lista, para lo cual nos situamos en el primer elemento de la base de datos, y mientras no se llegue al final copiamos el campo Modelo en la lista de transmisores. Las dos funciones siguientes hacen lo propio para la actualización de las listas de receptores y fibras ópticas.

```
Private Sub Combo1_GotFocus()
Data1.Recordset.MoveFirst
While Data1.Recordset.EOF = False
    'If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then
    Combo1.AddItem Data1.Recordset.Fields("Modelo")
    Data1.Recordset.MoveNext
Wend
End Sub

Private Sub Combo2_GotFocus()
Data2.Recordset.MoveFirst
While Data2.Recordset.EOF = False
    Combo2.AddItem Data2.Recordset.Fields("Modelo")
    Data2.Recordset.MoveNext
Wend
End Sub
```



```
Private Sub Combo3_GotFocus()  
Data3.Recordset.MoveFirst  
While Data3.Recordset.EOF = False  
    Combo3.AddItem Data3.Recordset.Fields("Modelo")  
    Data3.Recordset.MoveNext  
Wend  
End Sub
```

La función `Command1_Click()` se ejecuta cuando el usuario hace "click" sobre el botón Calcular. Dada la extensión de esta función se irán introduciendo comentarios intercalándolos con el código fuente. Las primeras líneas de la función se corresponden con la definición de variables locales, que serán utilizadas en el entorno de la función.

```
Private Sub Command1_Click()  
Dim balp As Boolean, balt As Boolean, cod As Boolean, codigo As String  
Dim ptx As Single, prx As Single, afib As Single, acon As Single  
Dim MSeg As Single, lenl As Single, NRegt As Single, vel As Single, restvel As Single  
Dim disp As Single, ilong As Single, ttx As Single, trx As Single  
Dim Ctx As Single, cadc As Single, Crx As Single, campf As Single, ccon As Single  
Dim cfib As Single, presup As Single, presupf As Single  
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String  
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer  
Dim lomintx As Integer, lominx As Integer, lominfo As Integer  
Dim LOenl As Integer  
Dim vb As Integer, vbl As Boolean, causa As String  
Dim TmatCuad As Single, TresCuad As Single, Tres As Single
```

Para evitar iniciar los cálculos sin haber seleccionado algún elemento de la base de datos de transmisores, introducimos la siguiente línea de código.

```
If Combo1.Text = "Lista de transmisores" Then  
    MsgBox "Seleccione un elemento de la lista de transmisores"  
End If
```

Lo mismo ocurre con los elementos de las bases de datos de receptores y fibras ópticas.

```
If Combo2.Text = "Lista de receptores" Then  
    MsgBox "Seleccione un elemento de la lista de receptores"  
End If  
If Combo3.Text = "Lista de fibras ópticas" Then  
    MsgBox "Seleccione un elemento de la lista de fibras ópticas"  
End If
```

Para identificar el elemento seleccionado para realizar los cálculos entre los elementos de cada una de las bases de datos ( Transmisores-Receptores-Fibras Ópticas ), dispondremos de un objeto Combo, enlazado con un objeto Data, para acceder a la base de datos.

Al iniciar la función, situamos cada uno de los objetos Data al principio de la base de datos correspondiente y comprobamos que existe algún elemento en la base de datos. En este caso, y hasta que se llegue al último elemento de la base de datos comparamos cada uno de los elementos de la base de datos con el elemento que ha sido seleccionado para realizar los cálculos ( de él hasta ahora tan sólo conocemos el campo "Modelo" que nos da información acerca del nombre del elemento, pero no disponemos de información del resto de sus características hasta que lo ubiquemos en la base de datos ). Esta operación se realiza en cada una de las tres bases de datos que tenemos.

---



```
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo2.Text)) Then
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo3.Text)) Then
```

Una vez identificados el transmisor, el receptor y la fibra óptica, asociamos el valor correspondiente a cada una de las variables definidas. El primer grupo de variables son pasadas como parámetro a la función *Compatibilidad* para analizar si son compatibles los elementos seleccionados con las características del enlace, basándonos para ello en dos parámetros: la longitud de onda del enlace y los tipo de conectores empleados.

```
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```

Si resultan compatibles los elementos seleccionados con las características del enlace, inicializamos cada una de las variables definidas, requeridas para los cálculos realizados por la función.

Se asigna el valor de cada una de las variables definidas en la función y que describen las características del enlace, del transmisor, del receptor, de la fibra óptica y de los conectores utilizados.

```
If vb = 0 Then
vbl = True
Else
vbl = False
End If
If vbl = True Then
ptx = Data1.Recordset.Fields("Potencia")
prx = Data2.Recordset.Fields("Potencia")
ttx = Data1.Recordset.Fields("Tiempo_Respuesta")
trx = Data2.Recordset.Fields("Tiempo_Respuesta")
Ctx = Data1.Recordset.Fields("Precio")
Crx = Data2.Recordset.Fields("Precio")
cadc = Data1.Recordset.Fields("Precio_CtoAdicional")
```



```
campf = Data2.Recordset.Fields("Precio_CtoAmplificador")
ccon = Data3.Recordset.Fields("Precio/metro")
ilong = Data1.Recordset.Fields("Incremento_LOnda")
afib = Data3.Recordset.Fields("Atenuacion")
disp = Data3.Recordset.Fields("Dispersion")
vel = Velocidad.Text
lenl = Distancia.Text
MSeg = MargS.Text
NRegt = NRg.Text
ccon = Precio.Text
acon = Perdidas.Text
If NRZ.Value = True Then
  cod = False
Else
  cod = True
End If
```

Para evaluar el balance de tiempos se aplicará la ecuación de evaluación del tiempo de subida, que nos dice que el tiempo de subida del sistema al cuadrado, expresado en nanosegundos, lo podemos calcular como la suma de los cuadrados de los tiempos de subida del generador, del receptor y de la fibra.

En función del tipo de código empleado, el tiempo de subida multiplicado por la velocidad de transmisión deberá ser menor que un determinado coeficiente:

```
RZ => Tr x B < 0.35   NRZ => Tr x B < 0.7

TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
TresCuad = ((tx ^ 2) + (trx ^ 2) + TmatCuad)
Tres = Sqr(TresCuad)
If cod = False Then
  cdg = 0.7
  codigo = "NRZ"
Else
  cdg = 0.35
  codigo = "RZ"
End If
```

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

$m = m + 1$

Llamo a la función *BalancePotencia* que me devolverá una variable lógica, en función de la cuál decidiré si paso al siguiente paso en la evaluación del enlace, o si por el contrario no se cumple el balance de potencia, terminando aquí el análisis del enlace e imprimiendo un mensaje que describe la causa.

```
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
If balp = False Then
  lmax = (ptx - (prx + 2 * acon + MSeg)) / afib
```



```
resulEnl(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    vbCrLf & _
    "Los conectores empleados tienen unas pérdidas de " & acon & "dB"
    & vbCrLf & _
    "y el Margen de Seguridad utilizado es de " & MSeg & "dB" &
    vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    vbCrLf & _
    "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " &
    vbCrLf & _
    vbCrLf & _
    ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax"
    & " + " & MSeg & vbCrLf & _
    "El Balance de Potencias ha resultado negativo" & vbCrLf & _
    "debiendo reducirse la distancia del enlace en " & Abs(lmax - lenl) & "Kms"

GoTo fin
End If
```

A continuación llamo a la función *BalanceTiempo* que me devolverá una variable lógica, en función de la cuál decidiré si paso al siguiente paso en la evaluación del enlace, o si por el contrario no se cumple el balance de tiempo, terminando aquí el análisis del enlace e imprimiendo un mensaje que describe la causa.

```
balt = baltime(displ, ilong, lenl, ttx, trx, vel, cod, NRegt)
If balt = False Then
    restvel = Abs(vel - (cdg / (0.001 * Tres)))
    resulEnl(m) = vbCrLf & _
        "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
        vbCrLf & _
        "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
        "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
        "Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
        "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
        vbCrLf & _
        "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
        vbCrLf & _
        "Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
        TmatCuad & " = (" & displ & " * " & ilong & " * (" & lenl & " / (1 + " & NRegt &
        " )) ^ 2" & vbCrLf & _
        "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
        TresCuad & " = " & (ttx ^ 2) & " + " & (trx ^ 2) & " + " & TmatCuad & vbCrLf
    & _
    & _
    "NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf
    & _
    codigo & " => " & Sqr(TresCuad) & " < " & cdg & "/0.001*" & vel & " = " &
    cdg / (vel * 0.001) & vbCrLf & _
    vbCrLf & _
    "El Balance de Tiempo ha resultado negativo" & vbCrLf & _
    "debiendo reducirse la velocidad del enlace en " & restvel & "Mb/s"

GoTo fin
End If
```



Finalmente, una vez que se ha comprobado que los elementos que constituyen el enlace son compatibles, y que se verifica el balance de potencia y de tiempo, tan sólo resta por realizar la evaluación económica de su implementación, expresada en Euros.

```
presup = coste(Ctx, cadc, Crx, campf, ccon, lenl, cfib, NRegt)
presupf = 100000000
If presup < presupf Then
presupf = presup
resulEnl(0) = vbCrLf & _
vbCrLf & _
"EL enlace óptimo es" & vbCrLf & _
Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo") & _
vbCrLf & _
"con un coste de " & presupf & " €" & vbCrLf & _
"requiriendo la solución " & NRegt & " nodos intermedios"
End If
```

En la variable presupf se irá almacenando el presupuesto más económico de los presupuestos que resulten implementables, y que podremos consultar pinchando sobre el nodo raíz del árbol de resultados.

Por último, tendré que almacenar el resultado obtenido en una variable tipo Cadena de Caracteres, de forma que cuando seleccione el nodo correspondiente en la estructura de árbol, se muestre en el cuadro de texto habilitado para ello el resultado correspondiente. Con este fin se ha definido una variable de carácter Público ( para que sea reconocida en todo el módulo ), como una lista de Cadena de Caracteres, de forma que cada elemento de la lista contendrá el resultado correspondiente al nodo referenciado.

```
resulEnl(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
"Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
"y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
vbCrLf & _
"Balance de Potencia" & vbCrLf & _
"Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf & _
ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + " &
MSeg & vbCrLf & _
vbCrLf & _
"Balance de Tiempos" & vbCrLf & _
"Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
TmatCuad & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " & NRegt & " )) ^ 2"
"Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
TresCuad & " = " & (ttx ^ 2) & " + " & (trx ^ 2) & " + " & TmatCuad & vbCrLf & _
"NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
```



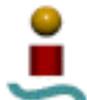
```
codigo & " => " & Sqr(TresCuad) & " < " & cdg & "/0.001*" & vel & " = " & cdg / (vel *  
0.001) & vbCrLf & _  
vbCrLf & _  
"Presupuesto para el Enlace de Fibra seleccionado " & presup & " €"  
fin:  
End If
```

En el caso de no ser compatible el enlace con alguno de los elementos que lo constituyen, se mostraría un mensaje de error identificando la causa y el elemento que lo ocasiona.

```
If vbl = False Then  
Select Case vb  
Case 1  
causa = "del tipo de conector usado para la fibra óptica"  
Case 2  
causa = "del tipo de conector usado para el transmisor"  
Case 3  
causa = "del tipo de conector usado para el receptor"  
Case 4  
causa = "de la longitud de onda del transmisor"  
Case 5  
causa = "de la longitud de onda del receptor"  
Case 6  
causa = "de la longitud de onda de la fibra óptica"  
End Select  
m = m + 1  
resulEnl(m) = vbCrLf & _  
vbCrLf & _  
"El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa  
MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa  
End If  
Set resulnodeEnl = ResultadosEnl.TreeView1.Nodes.Add(m)  
resulnodeEnl.Text = Data1.Recordset.Fields("Modelo") & "-" &  
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")  
End If
```

Se terminan los bucles y los condicionales abiertos al principio de la función.

```
End If  
Data3.Recordset.MoveNext  
Loop  
End If  
End If  
Data2.Recordset.MoveNext  
Loop  
End If  
End If  
Data1.Recordset.MoveNext  
Loop  
End If
```



Ejecutamos la función ResultadosBT que nos muestra los resultados obtenidos.

```
ResultadosEnl.Show  
End Sub
```

La función *GuardarComo* se utiliza para generar un fichero con la extensión *.enl* que será donde almacenemos los datos empleados para la realización de un determinado cálculo.

```
Private Sub GuardarComo_Click()  
Dim path As String  
Dim puntero  
GuardarENL.TipoSeleccion = Index  
GuardarENL.Show vbModal  
If Trim(GuardarENL.Retorna) <> "" Then  
    path = GuardarENL.Retorna  
End If  
puntero = FreeFile  
If path <> "" Then  
    Open path For Output As #puntero  
    Write #puntero, LOnda.Text, Velocidad.Text, Distancia.Text, NRg.Text, MargS.Text, _  
    TipoConector.Text, Perdidas.Text, Precio.Text, NRZ.Value, _  
    Combo1.Text, Combo2.Text, Combo3.Text  
End If  
Close #puntero  
End Sub
```

El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario de Balance de Potencia.

```
Private Sub Form_Unload(Cancel As Integer)  
Unload ResultadosEnl  
End Sub
```



## DISEÑO DEL ENLACE OPTIMO

### Descripción del formulario

Conocidas las características del enlace de fibra óptica y el tipo de conector del que disponemos, podremos, para un conjunto de equipos transmisores y receptores, y para un conjunto de fibras ópticas, calcular el enlace óptimo, que cumpla con las especificaciones requeridas y suponga el menor coste posible.

Diseño del Enlace Optimo

**SELECCIONAR PARÁMETROS DEL ENLACE**

DESCRIPCION DEL ENLACE

**SELECCIONAR ELEMENTOS DEL ENLACE**

TRANSMISORES

FIBRA OPTICA

RECEPTORES

DISEÑAR ENLACE

GUARDAR

**FIGURA 5.19.** Formulario principal empleado para el diseño del enlace óptimo.



Para implementar esta funcionalidad dispondremos de un formulario principal, desde donde llamaremos a otros formularios secundarios para definir las características del enlace, y seleccionar qué tipo de fibras y con qué equipos emisores y receptores calcularemos el enlace punto a punto. Una vez seleccionado, quedará reflejado en el correspondiente cuadro de texto habilitado en el formulario principal.

**CARACTERISTICAS DEL ENLACE DE FIBRA OPTICA**

**CARACTERISTICAS DEL ENLACE**

Longitud de Onda  ( nm Velocidad :  ( Mbps )

Distancia :  ( Km ) Margen de Seguridad  ( dB )

**Código**  NRZ  RZ

**CARACTERISTICAS DEL CONECTOR**

Tipo de conector disponible :

Perdidas :  ( dB ) Precio :  ( € )

VOLVER

**FIGURA 5.20.** Formulario empleado para especificar las características del enlace en el diseño del enlace óptimo.

El modo de empleo de este formulario será el siguiente:

Especificar las características del enlace que queremos diseñar, para lo cual pulsaremos el botón del formulario principal (FIGURA 5.1.) *Descripción del enlace*



En este formulario tendremos que introducir tanto las características del enlace, que quedarán especificadas introduciendo la longitud de onda del enlace, la velocidad, la distancia que queremos enlazar, y el margen de seguridad que disponemos, como el código de línea empleado y las características de los conectores disponibles.

**Selección de Emisor**

### CARACTERÍSTICAS DEL EMISOR

Modelo :

Tipo de conector :

Longitud de Onda Mínima :  (nm)

Longitud de Onda Máxima :  (nm)

Velocidad Máxima :  (Mbps)

Potencia :  (dBm)

Tiempo de Respuesta :  (ns)

Incremento Longitud Onda :  (nm)

Precio :  (€)

Precio Circuitería Adicional :  (€)

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR EMISOR

SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

VOLVER

**FIGURA 5.21.** Formulario empleado para seleccionar los emisores entre los disponibles en la Base de Datos de Emisores.

- Seleccionar los equipos emisores, receptores y fibras ópticas, para lo cual utilizaremos los botones habilitados para ello en el formulario principal: *Transmisores*, *Receptores*, *Fibras Ópticas*



Tendremos la opción de seleccionar para el diseño del enlace todos los emisores disponibles en nuestra Base de Datos, haciendo uso del botón *Seleccionar Todos*, o bien seleccionar un conjunto de ellos, utilizando el botón *Seleccionar*. Para esta última opción podremos desplazarnos por la Base de Datos mediante los botones creados para ello, y hacer una selección más precisa, en base a las características de los distintos elementos.

Cuando un elemento es seleccionado quedará grabado en el cuadro de texto situado en la parte derecha del formulario. Si queremos eliminar uno de los elementos seleccionados tendremos que marcarlo en el cuadro de texto y pulsar el botón *Eliminar*. Si pulsamos el botón *Eliminar Todos* automáticamente desaparecerán todos los elementos seleccionados.

Los formularios *Seleccionar Receptor* y *Seleccionar Fibra* son similares al descrito, y se muestran a continuación.

Selección de fibras ópticas

### CARACTERISTICAS DE LA FIBRA OPTICA

Modelo :

Tipo de Conector :

Longitud de Onda Mínima :  ( nm )

Longitud de Onda Máxima :  ( nm )

Atenuación :  ( dB / Km )

Dispersión :  ( ns / nm \* Km )

Precio / metro :  ( € )

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR FIBRAS OPTICAS

SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

VOLVER

**FIGURA 5.22.** Formulario empleado para seleccionar las fibras ópticas entre las disponibles en la Base de Datos de Fibras.

Una vez seleccionados los distintos elementos deberemos pulsar el botón *Volver* para regresar al formulario principal.



**Selección de Receptores**

### CARACTERÍSTICAS DEL RECEPTOR

Modelo : pin2

Tipo de conector : SC

Longitud de Onda Mínima : 850 (nm)

Longitud de Onda Máxima : 850 (nm)

Velocidad Máxima : 40 (Mbps)

Sensibilidad : -41 (dBm)

Tiempo de Respuesta : 5 (ns)

Precio : 12 (€)

Precio Circuitería Ampf. : 9 (€)

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR RECEPTOR

SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

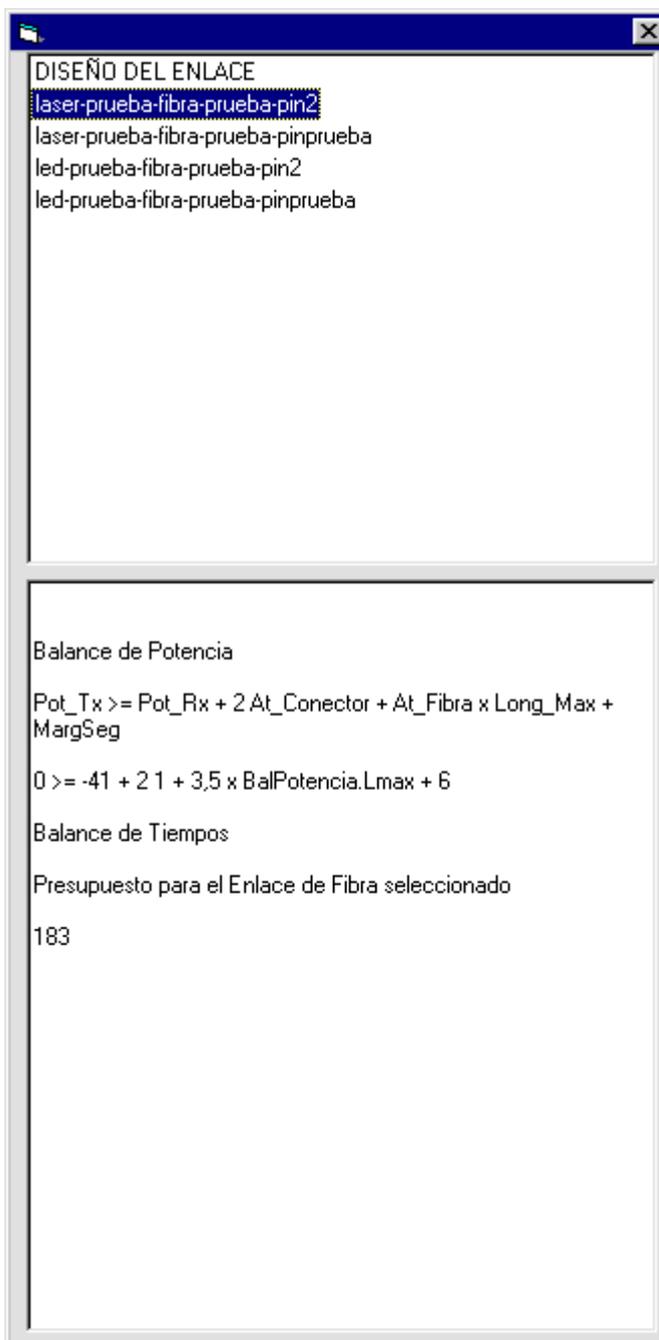
VOLVER

**FIGURA 5.23.** Formulario empleado para seleccionar los receptores entre los disponibles en la Base de Datos de Receptores.

- Calcular para cada conjunto de transmisor-receptor-fibra óptica el Balance de Potencia y de Tiempo, en caso de que en la triada seleccionada sean compatibles todos los elementos (Análisis de Viabilidad)
- Para las triadas donde el enlace es realizable (tanto el Análisis de Viabilidad, como el Balance de Potencia y Tiempo ha resultado satisfactorio), calcula el Presupuesto del enlace
- En caso contrario, nos dice porqué no es viable el enlace, y añade un nodo intermedio (conjunto de transmisor y receptor) en caso de que el fallo esté en el Balance de Potencia o de Tiempo, rehaciendo los cálculos
- Muestra los distintos resultados obtenidos para cada una de las triadas analizadas



- Para el conjunto de opciones calculadas, nos muestra cual de ellas es la mejor (la más económica, que cumple las especificaciones de diseño)



**FIGURA 5.24.** Formulario empleado para mostrar los resultados obtenidos.



### Controles utilizados en el formulario Diseño del Enlace Optimo

El título asignado al formulario en su Propiedad **Nombre** es DiseñoEnl. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	DISEÑO DEL ENLACE OPTIMO Label1
Botón de pulsación	Caption Name	DESCRIPCION DEL ENLACE Command1
Caja de Texto	Name Text	Text1 (nada)
Botón de pulsación	Caption Name	TRANSMISORES Command2
Caja de Texto	Name Text	List1 (nada)
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data1 Data1 C:\WINDOWS\Escritorio\ProyectoVB\Emisores.mdb Emisores False
Botón de pulsación	Caption Name	FIBRA OPTICA Command5
Caja de Texto	Name Text	List3 (nada)
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data3 Data3 C:\WINDOWS\Escritorio\ProyectoVB\Fibras.mdb Fibras False
Botón de pulsación	Caption Name	RECEPTORES Command3
Caja de Texto	Name Text	List2 (nada)
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data2 Data2 C:\WINDOWS\Escritorio\ProyectoVB\Receptores.md Receptores False
Botón de pulsación	Caption Name	DISEÑAR ENLACE Command4
Botón de pulsación	Caption Name	GUARDAR GuardarComo

**TABLA 5.8.** Tabla de controles y propiedades asociadas al formulario DiseñoEnl.

Los controles de datos empleados para este formulario son controles DATA, en vez de controles ADO. Conceptualmente son similares. Conectan un origen de datos con controles que poseen la propiedad **DataResource** ( controles enlazados con datos ), propiedad que les habilita para conectarse al control de datos y visualizar los datos del origen de datos.



### Controles utilizados en el formulario Seleccionar Enlace

El título asignado al formulario en su Propiedad **Nombre** es SeleccionarEnlace. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	CARACTERISTICAS DEL ENLACE Label2
Etiqueta	Caption Name	Longitud de Onda : Label3
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label7
Etiqueta	Caption Name	Velocidad : Label4
Caja de Texto	Name Text	Velocidad (nada)
Etiqueta	Caption Name	( Mbps ) Label9
Etiqueta	Caption Name	Distancia : Label5
Caja de Texto	Name Text	Distancia (nada)
Etiqueta	Caption Name	( Kms ) Label8
Etiqueta	Caption Name	Margen de Seguridad : Label10
Caja de Texto	Name Text	MargS (nada)
Etiqueta	Caption Name	( dB ) Label11
Etiqueta	Caption Name	Código : Label12
	Name Text	NRZ (nada)
	Name Text	RZ (nada)
Etiqueta	Caption Name	CARACTERISTICAS DEL CONECTOR Label1
Etiqueta	Caption Name	Tipo de conector disponible : Label6
Caja de Texto	Name Text	TipoConector (nada)
Etiqueta	Caption Name	Perdidas : Label13
Caja de Texto	Name Text	Perdidas (nada)
Etiqueta	Caption Name	( dB ) Label15



Etiqueta	Caption Name	Precio : Label14
Caja de Texto	Name Text	Precio (nada)
Etiqueta	Caption Name	( € ) Label16
Botón de pulsación	Caption Name	VOLVER Volver

**TABLA 5.9.** Tabla de controles y propiedades asociadas al formulario SeleccionarEnlace.

Los controles de datos empleados para este formulario son controles DATA, en vez de controles ADO. Conceptualmente son similares. Conectan un origen de datos con controles que poseen la propiedad **DataResource** ( controles enlazados con datos ), propiedad que les habilita para conectarse al control de datos y visualizar los datos del origen de datos.

### Controles utilizados en el formulario Seleccionar Emisor

El título asignado al formulario en su Propiedad **Nombre** es SeleccionarTx. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	SELECCIONAR EMISOR Label21
Etiqueta	Caption Name	Modelo : Label1
Caja de Texto	Name Text DataSource DataField	Text1 (nada) Adodc1 Modelo
Etiqueta	Caption Name	Tipo de conector : Label2
Caja de Texto	Name Text DataSource DataField	Text2 (nada) Adodc1 Tipo_Conector
Etiqueta	Caption Name	Longitud de Onda Mínima : Label3
Caja de Texto	Name Text DataSource DataField	Text3 (nada) Adodc1 Longitud_Onda_Min
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Longitud de Onda Máxima : Label4
Caja de Texto	Name Text DataSource DataField	Text4 (nada) Adodc1 Longitud_Onda_Max
Etiqueta	Caption Name	( nm ) Label13
Etiqueta	Caption	Velocidad Máxima :



	Name	Label5
Caja de Texto	Name Text DataSource DataField	Text5 (nada) Adodc1 Velocidad_Max
Etiqueta	Caption Name	( Mbps ) Label14
Etiqueta	Caption Name	Potencia : Label6
Caja de Texto	Name Text DataSource DataField	Text6 (nada) Adodc1 Potencia
Etiqueta	Caption Name	( dBm ) Label15
Etiqueta	Caption Name	Tiempo de Respuesta : Label7
Caja de Texto	Name Text DataSource DataField	Text7 (nada) Adodc1 Tiempo_Respuesta
Etiqueta	Caption Name	( ns ) Label16
Etiqueta	Caption Name	Precio : Label10
Caja de Texto	Name Text DataSource DataField	Text10 (nada) Adodc1 Precio
Etiqueta	Caption Name	( € ) Label19
Etiqueta	Caption Name	Precio Circuitería Adicional : Label11
Caja de Texto	Caption Text DataSource DataField	Text11 (nada) Adodc1 Precio_CtoAdicional
Etiqueta	Caption Name	( € ) Label20
Botón de Pulsación	Caption Name	SELECCIONAR Seleccionar
Botón de Pulsación	Caption Name	SELECCIONAR TODOS SeleccionarTodo
Botón de Pulsación	Caption Name	ELIMINAR TODOS EliminarTodo
Botón de Pulsación	Caption Name	ELIMINAR Eliminar
Botón de Pulsación	Caption Name	Inicio Inicio



Botón de Pulsación	Caption Name	Anterior Anterior
Botón de Pulsación	Caption Name	Siguiente Siguiete
Botón de Pulsación	Caption Name	Final Final
Botón de Pulsación	Caption Name	Buscar Buscar
Botón de Pulsación	Caption Name	VOLVER Volver
Control de Datos	Caption Name ConnectionString RecordSource Visible	Adodc1 Adodc1 C:\WINDOWS\Escritorio\ProyectoVB\Emisores.udl select * from Emisores order by Modelo False

**TABLA 5.10.** Tabla de controles y propiedades asociadas al formulario SeleccionarTx.

### Controles utilizados en el formulario Seleccionar Receptor

El título asignado al formulario en su Propiedad **Nombre** es SeleccionrRx. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	SELECCIONAR RECEPTOR Label17
Etiqueta	Caption Name	Modelo : Label1
Caja de Texto	Name Text DataSource DataField	Text1 (nada) Adodc1 Modelo
Etiqueta	Caption Name	Tipo de conector : Label2
Caja de Texto	Name Text DataSource DataField	Text2 (nada) Adodc1 Tipo_Conector
Etiqueta	Caption Name	Longitud de Onda Mínima : Label3
Caja de Texto	Name Text DataSource DataField	Text3 (nada) Adodc1 Longitud_Onda_Min
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Longitud de Onda Máxima : Label4
Caja de Texto	Name Text DataSource DataField	Text4 (nada) Adodc1 Longitud_Onda_Max
Etiqueta	Caption	( nm )



	Name	Label13
Etiqueta	Caption Name	Velocidad Máxima : Label5
Caja de Texto	Name Text DataSource DataField	Text5 (nada) Adodc1 Velocidad_Max
Etiqueta	Caption Name	( Mbps ) Label14
Etiqueta	Caption Name	Potencia : Label6
Caja de Texto	Name Text DataSource DataField	Text6 (nada) Adodc1 Potencia
Etiqueta	Caption Name	( dBm ) Label15
Etiqueta	Caption Name	Tiempo de Respuesta : Label7
Caja de Texto	Name Text DataSource DataField	Text7 (nada) Adodc1 Tiempo_Respuesta
Etiqueta	Caption Name	( ns ) Label16
Etiqueta	Caption Name	Precio : Label10
Caja de Texto	Name Text DataSource DataField	Text10 (nada) Adodc1 Precio
Etiqueta	Caption Name	( € ) Label19
Etiqueta	Caption Name	Precio Circuitería Amplificadora : Label11
Caja de Texto	Caption Text DataSource DataField	Text11 (nada) Adodc1 Precio_CtoAmplificador
Etiqueta	Caption Name	( € ) Label20
Botón de Pulsación	Caption Name	SELECCIONAR Seleccionar
Botón de Pulsación	Caption Name	SELECCIONAR TODOS SeleccionarTodo
Botón de Pulsación	Caption Name	ELIMINAR TODOS EliminarTodo
Botón de Pulsación	Caption Name	ELIMINAR Eliminar
Botón de Pulsación	Caption Name	Inicio Inicio



Botón de Pulsación	Caption Name	Anterior Anterior
Botón de Pulsación	Caption Name	Siguiente Siguiete
Botón de Pulsación	Caption Name	Final Final
Botón de Pulsación	Caption Name	Buscar Buscar
Botón de Pulsación	Caption Name	VOLVER Volver
Control de Datos	Caption Name ConnectionString RecordSource Visible	Adodc1 Adodc1 C:\WINDOWS\Escritorio\ProyectoVB\Receptores.udl select * from es order by Modelo False

**TABLA 5.11.** Tabla de controles y propiedades asociadas al formulario SeleccionrRx.

### Controles utilizados en el formulario Seleccionar Fibra Optica

El título asignado al formulario en su Propiedad **Nombre** es SeleccionarFibras. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	SELECCIONAR FIBRA OPTICA Label21
Etiqueta	Caption Name	Modelo : Label1
Caja de Texto	Name Text DataSource DataField	Modelo (nada) Adodc1 Modelo
Etiqueta	Caption Name	Tipo de conector : Label2
Caja de Texto	Name Text DataSource DataField	TipoConector (nada) Adodc1 Tipo_Conector
Etiqueta	Caption Name	Longitud de Onda Mínima : Label3
Caja de Texto	Name Text DataSource DataField	LOndaMin (nada) Adodc1 Longitud_Onda_Min
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Longitud de Onda Máxima : Label4
Caja de Texto	Name Text DataSource DataField	LOndaMax (nada) Adodc1 Longitud_Onda_Max
Etiqueta	Caption Name	( nm ) Label13
Etiqueta	Caption	Atenuación :



	Name	Label6
Caja de Texto	Name Text DataSource DataField	Atenuacion (nada) Adodc1 Atenuacion
Etiqueta	Caption Name	( dB / Km ) Label15
Etiqueta	Caption Name	Dispersión : Label7
Caja de Texto	Name Text DataSource DataField	Dispersion (nada) Adodc1 Dispersion
Etiqueta	Caption Name	( ns / nm * Km ) Label16
Etiqueta	Caption Name	Precio / metro : Label10
Caja de Texto	Name Text DataSource DataField	Precio (nada) Adodc1 Precio/metro
Etiqueta	Caption Name	( € ) Label19
Botón de Pulsación	Caption Name	SELECCIONAR Seleccionar
Botón de Pulsación	Caption Name	SELECCIONAR TODOS SeleccionarTodo
Botón de Pulsación	Caption Name	ELIMINAR TODOS EliminarTodo
Botón de Pulsación	Caption Name	ELIMINAR Eliminar
Botón de Pulsación	Caption Name	Inicio Inicio
Botón de Pulsación	Caption Name	Anterior Anterior
Botón de Pulsación	Caption Name	Siguiente Siguiete
Botón de Pulsación	Caption Name	Final Final
Botón de Pulsación	Caption Name	Buscar Buscar
Botón de Pulsación	Caption Name	VOLVER Volver
Control de Datos	Caption Name ConnectionString RecordSource Visible	Adodc1 Adodc1 C:\WINDOWS\Escritorio\ProyectoVB\Fibras.udl select * from es order by Modelo False

**TABLA 5.12.** Tabla de controles y propiedades asociadas al formulario SeleccionarFibras.



## Descripción del código fuente

### DISEÑO DEL ENLACE

#### Option Explicit

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnode* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad Index. El valor de índice del último miembro agregado es el valor de la propiedad Count de la colección Nodes.

```
Dim resulnode As Node
Dim m As Integer
```

Cuando el formulario es cargado ( Private Sub Form\_Load() ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción Set con una variable de tipo Node ( *resulnode* ). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad Text, asignándole el nombre que queremos que tenga el nodo (DISEÑO DEL ENLACE).

```
Private Sub Form_Load()
m = 0
Set resulnode = Resultados.TreeView1.Nodes.Add()
resulnode.Text = "DISEÑO DEL ENLACE"
End Sub
```

La función Command1\_Click() se ejecuta cuando el usuario hace "click" sobre el botón *Descripción del Enlace*. Esta acción provoca la aparición de un nuevo formulario donde se seleccionarán las características referentes al enlace y a los conectores.

```
Private Sub Command1_Click()
SeleccionarEnlace.Show vbModal, Me
End Sub
```

La función Command2\_Click() se ejecuta cuando el usuario hace "click" sobre el botón *Transmisores*. Esta acción provoca la aparición de un nuevo formulario donde se seleccionarán los emisores que serán utilizados para el diseño del enlace.

```
Private Sub Command2_Click()
SeleccionTx.Show vbModal, Me
End Sub
```

La función Command3\_Click() se ejecuta cuando el usuario hace "click" sobre el botón *Receptores*. Esta acción provoca la aparición de un nuevo formulario donde se seleccionarán los receptores que serán utilizados para el diseño del enlace.

```
Private Sub Command3_Click()
SeleccionarRx.Show vbModal, Me
End Sub
```



La función Command5\_Click() se ejecuta cuando el usuario hace "click" sobre el botón *Fibra Óptica*. Esta acción provoca la aparición de un nuevo formulario donde se seleccionarán las fibras ópticas que serán utilizados para el diseño del enlace.

```
Private Sub Command5_Click()  
    SeleccionarFibras.Show vbModal, Me  
End Sub
```

La función Command4\_Click() se ejecuta cuando el usuario hace "click" sobre el botón *Diseño del Enlace*. Dada la extensión de esta función se irán introduciendo comentarios intercalándolos con el código fuente. Las primeras líneas de la función se corresponden con la definición de variables locales, que serán utilizadas en el entorno de la función.

```
Private Sub Command4_Click()  
    Dim i As Integer, j As Integer, k As Integer  
    Dim balp As Boolean, balt As Boolean, presup As Single, presupf As Single  
    Dim balpp As Boolean, balpt As Boolean, baltt As Boolean  
    Dim ptx As Single, prx As Single, afib As Single, acon As Single  
    Dim MSeg As Single, lenl As Single, NRegt As Single, vel As Single, cod As Boolean  
    Dim disp As Single, ilong As Single, ttx As Single, trx As Single  
    Dim Ctx As Single, cadc As Single, Crx As Single, campf As Single, ccon As Single  
    Dim cfib As Single  
    Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String  
    Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer  
    Dim lomintx As Integer, lominrx As Integer, lominfo As Integer  
    Dim LOenl As Integer  
    Dim vb As Integer, vbl As Boolean, causa As String
```

Fijamos un valor muy alto para el presupuesto final en la variable presupf que luego iremos comparando con los presupuestos calculados para que triada de elementos (tx-rx-fo), y se irá sustituyendo por el menor. Esto me permitirá averiguar cuál es la solución óptima desde el punto de vista económico.

```
presupf = 1000000
```

Recorremos cada una de las listas de componentes que optan a formar parte del enlace, situando cada uno de los objetos Data al principio de la base de datos correspondiente y comprobando que existe algún elemento en la base de datos. En este caso, y hasta que se llegue al último elemento de la base de datos comparamos cada uno de los elementos de la base de datos con el elemento que ha sido seleccionado para realizar los cálculos ( de él hasta ahora tan sólo conocemos el campo "Modelo" que nos da información acerca del nombre del elemento, pero no disponemos de información del resto de sus características hasta que lo ubiquemos en la base de datos ). Esta operación se realiza en cada una de las tres bases de datos que tenemos.

```
For i = 0 To List1.ListCount - 1  
    Data1.Recordset.MoveFirst  
    For j = 0 To List2.ListCount - 1  
        Data1.Recordset.MoveFirst  
        Data2.Recordset.MoveFirst  
        For k = 0 To List3.ListCount - 1  
            Data1.Recordset.MoveFirst  
            Data2.Recordset.MoveFirst  
            Data3.Recordset.MoveFirst  
            If Data1.Recordset.RecordCount > 0 Then  
                Do Until Data1.Recordset.EOF
```



```
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(List1.List(i - 1))) Then
'Busco el Receptor
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(List2.List(j - 1))) Then
'Busco la Fibra
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(List3.List(k - 1))) Then
```

Una vez identificados el transmisor, el receptor y la fibra óptica, asociamos el valor correspondiente a cada una de las variables definidas. El primer grupo de variables son pasadas como parámetro a la función *Compatibilidad* para analizar si son compatibles los elementos seleccionados con las características del enlace, basándonos para ello en dos parámetros: la longitud de onda del enlace y los tipo de conectores empleados.

```
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = SeleccionarEnlace.TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = SeleccionarEnlace.LOnda.Text
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```

Si resultan compatibles los elementos seleccionados con las características del enlace, inicializamos cada una de las variables definidas, requeridas para los cálculos realizados por la función.

Se asigna el valor de cada una de las variables definidas en la función y que describen las características del enlace, del transmisor, del receptor, de la fibra óptica y de los conectores utilizados.

```
If vb = 0 Then
vbl = True
Else
vbl = False
End If
If vbl = True Then
ptx = Data1.Recordset.Fields("Potencia")
prx = Data2.Recordset.Fields("Potencia")
ilong = Data1.Recordset.Fields("Incremento_LOnda")
ttx = Data1.Recordset.Fields("Tiempo_Respuesta")
trx = Data2.Recordset.Fields("Tiempo_Respuesta")
Ctx = Data1.Recordset.Fields("Precio")
Crx = Data2.Recordset.Fields("Precio")
cadc = Data1.Recordset.Fields("Precio_CtoAdicional")
campf = Data2.Recordset.Fields("Precio_CtoAmplificador")
ccon = Data3.Recordset.Fields("Precio/metro")
afib = Data3.Recordset.Fields("Atenuacion")
```



```
disp = Data3.Recordset.Fields("Dispersion")
vel = SeleccionarEnlace.Velocidad.Text
lenl = SeleccionarEnlace.Distancia.Text
MSeg = SeleccionarEnlace.MargS.Text
ccon = SeleccionarEnlace.Precio.Text
acon = SeleccionarEnlace.Perdidas.Text
NRegt = 0
If SeleccionarEnlace.NRZ.Value = True Then
cod = False
Else
cod = True
End If

balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
If balp = False Then
NRegt = NRegt + 1
balpp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
End If
balt = baltime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)
If balt = False Then
NRegt = NRegt + 1
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
If balp = False Then
NRegt = NRegt + 1
balpt = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
End If
baltt = baltime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)
End If
presup = coste(Ctx, cadc, Crx, campf, ccon, lenl, cfib, NRegt)
If presup < presupf Then
presupf = presup
resul(0) = vbCrLf & _
vbCrLf & _
"EL enlace óptimo es" & vbCrLf & _
Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo") & _
vbCrLf & _
"con un coste de " & presupf & "ptas." & vbCrLf & _
"requiriendo la solución " & NRegt & " nodos intermedios"
End If
```

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

```
m = m + 1
```



```
resul(m) = vbCrLf & _
vbCrLf & _
"Balance de Potencia" & vbCrLf & _
vbCrLf & _
"Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf &
-
vbCrLf & _
ptx & " >=" & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + " &
MSeg & vbCrLf & _
vbCrLf & _
"Balance de Tiempos" & vbCrLf & _
vbCrLf & _
"Presupuesto para el Enlace de Fibra seleccionado " & vbCrLf & _
vbCrLf & _
presup
End If
```

En el caso de no ser compatible el enlace con alguno de los elementos que lo constituyen, se mostraría un mensaje de error identificando la causa y el elemento que lo ocasiona.

```
If vbl = False Then
Select Case vb
Case 1
causa = "del tipo de conector usado para la fibra óptica"
Case 2
causa = "del tipo de conector usado para el transmisor"
Case 3
causa = "del tipo de conector usado para el receptor"
Case 4
causa = "de la longitud de onda del transmisor"
Case 5
causa = "de la longitud de onda del receptor"
Case 6
causa = "de la longitud de onda de la fibra óptica"
End Select
m = m + 1
resul(m) = vbCrLf & _
vbCrLf & _
"El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
End If
Set resulnode = Resultados.TreeView1.Nodes.Add(m)
resulnode.Text = Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") &
"-" & Data2.Recordset.Fields("Modelo")
```



Se terminan los bucles y los condicionales abiertos al principio de la función.

```
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
Next k
Next j
Next l
```

Ejecutamos la función Resultados que nos muestra los resultados obtenidos.

```
Resultados.Show
End Sub
```

La función *GuardarComo* se utiliza para generar un fichero con la extensión *.den* que será donde almacenemos los datos empleados para la realización de un determinado cálculo

```
Private Sub GuardarComo_Click()
Dim path As String, Index As Variant
Dim puntero
Dim i As Integer, j As Integer, k As Integer
GuardarDEN.TipoSeleccion = Index
GuardarDEN.Show vbModal
If Trim(GuardarDEN.Retorna) <> "" Then
path = GuardarDEN.Retorna
End If
puntero = FreeFile
If path <> "" Then
Open path For Output As #puntero
Write #puntero, SeleccionarEnlace.LOnda.Text, SeleccionarEnlace.Velocidad.Text,
SeleccionarEnlace.Distancia.Text, _
SeleccionarEnlace.MargS.Text, SeleccionarEnlace.NRZ.Value,
SeleccionarEnlace.TipoConector.Text, _
SeleccionarEnlace.Perdidas.Text, SeleccionarEnlace.Precio.Text, _
List1.ListCount - 2, List2.ListCount - 2, List3.ListCount - 2
For i = 0 To List1.ListCount - 2
Write #puntero, List1.List(i)
Next i
For j = 0 To List2.ListCount - 2
Write #puntero, List2.List(j)
Next j
For k = 0 To List3.ListCount - 2
Write #puntero, List3.List(k)
Next k
End If
Close #puntero
End Sub
```



El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario.

```
Private Sub Form_Unload(Cancel As Integer)
Unload Resultados
End Sub
```

## SELECCIONAR ENLACE

Se introducirán los valores correspondientes a los parámetros que caracterizan el enlace y los conectores.

```
Private Sub Volver_Click()
Dim vel As String, lenl As String, MSeg As String, lond As String
Dim ccon As String, acon As String
Dim codigo As String
vel = Velocidad.Text
lenl = Distancia.Text
MSeg = MargS.Text
lond = LOnda.Text
ccon = Precio.Text
acon = Perdidas.Text
If NRZ.Value = True Then
cod = False
codigo = "NRZ"
Else
cod = True
codigo = "RZ"
End If
DiseñoEnl.Text1 = "Enlace punto a punto a una longitud de onda de " & LOnda.Text & "nm." _
& " La distancia a enlazar es de " & Distancia.Text & "Kms, pudiéndose incluir repetidores" _
& " optoelectrónicos en tramos intermedios en caso necesario." _
& " La velocidad de transmisión es de " & Velocidad.Text & "Mbps." _
& " La codificación empleada es " & codigo _
& " Se usarán conectores con pérdidas de " & acon & "dB"
SeleccionarEnlace.Hide
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
Cancel = True
MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```



```
Private Sub MargS_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(MargS) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```



## SELECCIONAR TX

### Option Explicit

A continuación se describen las funciones que permitirán habilitar y deshabilitar los botones cuando la acción que se pretenda realizar no sea muy coherente, como por ejemplo cambiar de registro en una operación de altas, o pulsar reiteradamente el botón *Nuevo*.

Para habilitar o inhabilitar un control en general, actuaremos sobre su propiedad Enabled, asignando un valor True (habilitado) o False (inhabilitado).

La función *Inhabilitar\_Cajas* me permite inhabilitar todos aquellos controles que sean cajas de texto. El control TextBox (Caja de Texto), es un control de una matriz de controles asociada al formulario denominada Controls. Con la condición TypeOf Controls(n) Is TextBox estoy diciendo que para todo control del formulario que sea del tipo TextBox se realice la acción de inhabilitar dicho control: Controls(n).Enabled = False. La colección Controls tiene una propiedad, Count, que especifica el número de elementos de la matriz de controles (el primer elemento de la matriz es el cero).

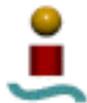
```
Private Sub InhabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

Un razonamiento análogo nos conduce al procedimiento que permite habilitar todos los controles que sean cajas de texto:

```
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

El mismo razonamiento se seguiría para inhabilitar o habilitar botones de pulsación.

```
Private Sub InhabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```



```
Private Sub HabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

Para posicionarnos en el primer registro de la Base de Datos usaremos el botón *Inicio* del formulario. El código asociado a ese botón se muestra a continuación:

```
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub
```

El evento que desarrolla el procedimiento desencadenado por el botón *Inicio* es el evento Click (al pulsar el botón se desarrolla la acción asociada). El método MoveFirst asociado al objeto Adodc1.Recordset me permite posicionarme en el primer registro.

Siguiendo un razonamiento análogo para cada uno de los otros tres botones ( *Anterior*, *Siguiente* y *Final* ), el código asociado a cada uno de ellos sería:

```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```

```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

La función Seleccionar nos permite añadir un nuevo elemento a la lista de emisores que se utilizarán para el diseño del enlace. Habrá que comprobar primero que ese elemento no se encuentra ya en la lista para no volver a añadirlo.

```
Private Sub Seleccionar_Click()  
    Dim i As Integer  
    For i = 0 To List1.ListCount  
        If (RTrim(Modelo.Text) = RTrim(List1.List(i))) Then  
            GoTo fin  
        End If  
    Next i  
    List1.AddItem Modelo.Text  
fin:  
End Sub
```

---



Con esta función se podrán seleccionar todos los elementos almacenados en la base de datos de Emisores. Para ello lo que hacemos es copiar el campo "Modelo" de cada uno de los elementos a una lista, desde donde se irán referenciando a la hora de proceder a los cálculos para el diseño del enlace.

```
Private Sub SeleccionarTodo_Click()  
    Adodc1.Recordset.MoveFirst  
    While Adodc1.Recordset.EOF = False  
        List1.AddItem Adodc1.Recordset.Fields("Modelo")  
        Adodc1.Recordset.MoveNext  
    Wend  
End Sub
```

Esta función permite eliminar un elemento seleccionado en la lista de emisores.

```
Private Sub Eliminar_Click()  
    Dim r As Integer  
    r = List1.ListIndex  
    List1.RemoveItem r  
End Sub
```

Función utilizada para limpiar la lista de emisores seleccionados.

```
Private Sub EliminarTodo_Click()  
    List1.Clear  
End Sub
```

El botón Buscar permitirá al usuario buscar un registro determinado, utilizando el método FindFirst y código SQL. El lenguaje SQL es un lenguaje avanzado para la consulta y modificación de bases de datos.

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo del emisor que quiere buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like '*' & Buscado & '*'"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
        ' Buscar desde el principio  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
        MsgBox ("No encuentro ese modelo")  
    End If  
End Sub
```

Con el botón Volver regresamos al formulario padre donde copiaremos la lista de emisores seleccionados.

---



```
Private Sub Volver_Click()  
Dim i As Integer  
    SeleccionTx.Hide  
    For i = 0 To List1.ListCount  
        DiseñoEnl.List1.AddItem List1.List(i)  
    Next i  
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

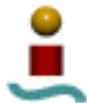
```
Private Sub IncLonda_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(IncLonda) Then  
    Cancel = True  
    MsgBox "El incremento de la longitud de onda debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub LOndaMax_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOndaMax) Then  
    Cancel = True  
    MsgBox "La longitud de onda máxima debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub LOndaMin_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOndaMin) Then  
    Cancel = True  
    MsgBox "La longitud de onda mínima debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Potencia_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Potencia) Then  
    Cancel = True  
    MsgBox "La potencia debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub PrecCirc_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(PrecCirc) Then  
    Cancel = True  
    MsgBox "El precio del circuito adicional debe ser un valor numérico"  
End If  
End Sub
```



```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TRespuesta_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(TRespuesta) Then
    Cancel = True
    MsgBox "El tiempo de respuesta debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```

## SELECCIONAR RX

### Option Explicit

A continuación se describen las funciones que permitirán habilitar y deshabilitar los botones cuando la acción que se pretenda realizar no sea muy coherente, como por ejemplo cambiar de registro en una operación de altas, o pulsar reiteradamente el botón *Nuevo*.

Para habilitar o inhabilitar un control en general, actuaremos sobre su propiedad Enabled, asignando un valor True (habilitado) o False (inhabilitado).

La función *Inhabilitar\_Cajas* me permite inhabilitar todos aquellos controles que sean cajas de texto. El control TextBox (Caja de Texto), es un control de una matriz de controles asociada al formulario denominada Controls. Con la condición `TypeOf Controls(n) Is TextBox` estoy diciendo que para todo control del formulario que sea del tipo TextBox se realice la acción de inhabilitar dicho control: `Controls(n).Enabled = False`. La colección Controls tiene una propiedad, Count, que especifica el número de elementos de la matriz de controles (el primer elemento de la matriz es el cero).

```
Private Sub InhabilitarCajas()
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```



Un razonamiento análogo nos conduce al procedimiento que permite habilitar todos los controles que sean cajas de texto:

```
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

El mismo razonamiento se seguiría para inhabilitar o habilitar botones de pulsación.

```
Private Sub InhabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

Para posicionarnos en el primer registro de la Base de Datos usaremos el botón *Inicio* del formulario. El código asociado a ese botón se muestra a continuación:

```
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub
```

El evento que desarrolla el procedimiento desencadenado por el botón *Inicio* es el evento Click (al pulsar el botón se desarrolla la acción asociada). El método MoveFirst asociado al objeto Adodc1.Recordset me permite posicionarme en el primer registro.

Siguiendo un razonamiento análogo para cada uno de los otros tres botones ( *Anterior*, *Siguiente* y *Final* ), el código asociado a cada uno de ellos sería:

```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```



```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

La función Seleccionar nos permite añadir un nuevo elemento a la lista de receptores que se utilizarán para el diseño del enlace. Habrá que comprobar primero que ese elemento no se encuentra ya en la lista para no volver a añadirlo.

```
Private Sub Seleccionar_Click()  
    Dim i As Integer  
    For i = 0 To List1.ListCount  
        If (RTrim(Modelo.Text) = RTrim(List1.List(i))) Then  
            GoTo fin  
        End If  
    Next i  
    List1.AddItem Modelo.Text  
fin:  
End Sub
```

Con esta función se podrán seleccionar todos los elementos almacenados en la base de datos de Receptores. Para ello lo que hacemos es copiar el campo "Modelo" de cada uno de los elementos a una lista, desde donde se irán referenciando a la hora de proceder a los cálculos para el diseño del enlace.

```
Private Sub SeleccionarTodo_Click()  
    Adodc1.Recordset.MoveFirst  
    While Adodc1.Recordset.EOF = False  
        List1.AddItem Adodc1.Recordset.Fields("Modelo")  
        Adodc1.Recordset.MoveNext  
    Wend  
    'Adodc1.Recordset.MoveLast  
End Sub
```

Esta función permite eliminar un elemento seleccionado en la lista de receptores.

```
Private Sub Eliminar_Click()  
    Dim r As Integer  
    r = List1.ListIndex  
    List1.RemoveItem r  
End Sub
```

Función utilizada para limpiar la lista de receptores seleccionados.

```
Private Sub EliminarTodo_Click()  
    List1.Clear  
End Sub
```



El botón Buscar permitirá al usuario buscar un registro determinado, utilizando el método FindFirst y código SQL. El lenguaje SQL es un lenguaje avanzado para la consulta y modificación de bases de datos.

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo del receptor que quiere buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like '*' & Buscado & '*'"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
        ' Buscar desde el principio  
        Adodc1.Recordset.Find Criterio  
        If Adodc1.Recordset.EOF Then  
            Adodc1.Recordset.MoveLast  
            MsgBox ("No encuentro ese modelo")  
        End If  
    End If  
End Sub
```

Con el botón Volver regresamos al formulario padre donde copiaremos la lista de receptores seleccionados.

```
Private Sub Volver_Click()  
    Dim i As Integer  
    SeleccionarRx.Hide  
    For i = 0 To List1.ListCount  
        DiseñoEnl.List2.AddItem List1.List(i)  
    Next i  
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOndaMax_Validate(Cancel As Boolean)  
    'Si el valor no es un valor numérico, mantener el foco  
    If Not IsNumeric(LOndaMax) Then  
        Cancel = True  
        MsgBox "La longitud de onda máxima debe ser un valor numérico"  
    End If  
End Sub
```



```
Private Sub LOndaMin_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMin) Then
    Cancel = True
    MsgBox "La longitud de onda mínima debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Potencia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Potencia) Then
    Cancel = True
    MsgBox "La potencia debe ser un valor numérico"
End If
End Sub
```

```
Private Sub PrecCirc_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(PrecCirc) Then
    Cancel = True
    MsgBox "El precio del circuito adicional debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TRespuesta_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(TRespuesta) Then
    Cancel = True
    MsgBox "El tiempo de respuesta debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
```



## SELECCIONAR FIBRA

### Option Explicit

A continuación se describen las funciones que permitirán habilitar y deshabilitar los botones cuando la acción que se pretenda realizar no sea muy coherente, como por ejemplo cambiar de registro en una operación de altas, o pulsar reiteradamente el botón *Nuevo*.

Para habilitar o inhabilitar un control en general, actuaremos sobre su propiedad Enabled, asignando un valor True (habilitado) o False (inhabilitado).

La función *Inhabilitar\_Cajas* me permite inhabilitar todos aquellos controles que sean cajas de texto. El control TextBox (Caja de Texto), es un control de una matriz de controles asociada al formulario denominada Controls. Con la condición `TypeOf Controls(n) Is TextBox` estoy diciendo que para todo control del formulario que sea del tipo TextBox se realice la acción de inhabilitar dicho control: `Controls(n).Enabled = False`. La colección Controls tiene una propiedad, Count, que especifica el número de elementos de la matriz de controles (el primer elemento de la matriz es el cero).

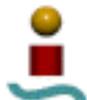
```
Private Sub InhabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

Un razonamiento análogo nos conduce al procedimiento que permite habilitar todos los controles que sean cajas de texto:

```
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

El mismo razonamiento se seguiría para inhabilitar o habilitar botones de pulsación.

```
Private Sub InhabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```



```
Private Sub HabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

Para posicionarnos en el primer registro de la Base de Datos usaremos el botón *Inicio* del formulario. El código asociado a ese botón se muestra a continuación:

```
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub
```

El evento que desarrolla el procedimiento desencadenado por el botón *Inicio* es el evento Click (al pulsar el botón se desarrolla la acción asociada). El método MoveFirst asociado al objeto Adodc1.Recordset me permite posicionarme en el primer registro.

Siguiendo un razonamiento análogo para cada uno de los otros tres botones ( *Anterior*, *Siguiente* y *Final* ), el código asociado a cada uno de ellos sería:

```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```

```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

La función Seleccionar nos permite añadir un nuevo elemento a la lista de fibras ópticas que se utilizarán para el diseño del enlace. Habrá que comprobar primero que ese elemento no se encuentra ya en la lista para no volver a añadirlo.

```
Private Sub Seleccionar_Click()  
    Dim i As Integer  
    For i = 0 To List1.ListCount  
        If (RTrim(Modelo.Text) = RTrim(List1.List(i))) Then  
            GoTo fin  
        End If  
    Next i  
    List1.AddItem Modelo.Text  
End Sub
```

---



Con esta función se podrán seleccionar todos los elementos almacenados en la base de datos de Fibras Ópticas. Para ello lo que hacemos es copiar el campo "Modelo" de cada uno de los elementos a una lista, desde donde se irán referenciando a la hora de proceder a los cálculos para el diseño del enlace.

```
Private Sub SeleccionarTodo_Click()  
    Adodc1.Recordset.MoveFirst  
    While Adodc1.Recordset.EOF = False  
        List1.AddItem Adodc1.Recordset.Fields("Modelo")  
        Adodc1.Recordset.MoveNext  
    Wend  
    'Adodc1.Recordset.MoveLast  
End Sub
```

Esta función permite eliminar un elemento seleccionado en la lista de fibras ópticas.

```
Private Sub Eliminar_Click()  
    Dim r As Integer  
    r = List1.ListIndex  
    List1.RemoveItem r  
End Sub
```

Función utilizada para limpiar la lista de fibras ópticas seleccionadas.

```
Private Sub EliminarTodo_Click()  
    List1.Clear  
End Sub
```

El botón Buscar permitirá al usuario buscar un registro determinado, utilizando el método FindFirst y código SQL. El lenguaje SQL es un lenguaje avanzado para la consulta y modificación de bases de datos.

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo de la fibra óptica que quiere buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like "*" & Buscado & ""*"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
        ' Buscar desde el principio  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
        MsgBox ("No encuentro ese modelo")  
    End If  
End If  
End Sub
```



Con el botón Volver regresamos al formulario padre donde copiaremos la lista de fibras ópticas seleccionadas.

```
Private Sub Volver_Click()  
Dim i As Integer  
    SeleccionarFibras.Hide  
    For i = 0 To List1.ListCount  
        DiseñoEnl.List3.AddItem List1.List(i)  
    Next i  
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOndaMax_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOndaMax) Then  
    Cancel = True  
    MsgBox "La longitud de onda máxima debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub LOndaMin_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOndaMin) Then  
    Cancel = True  
    MsgBox "La longitud de onda mínima debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Precio) Then  
    Cancel = True  
    MsgBox "El precio debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Atenuacion_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Atenuacion) Then  
    Cancel = True  
    MsgBox "La atenuación debe ser un valor numérico"  
End If
```



## SELECCIÓN DEL COMPONENTE ÓPTIMO

### Descripción del formulario

Este formulario nos permitirá, conocidas todas las características del enlace de fibra, obtener una información completa del mismo. De este modo, podremos conocer si el enlace es o no implementable, si podemos ajustar algún parámetro consiguiendo optimizar su comportamiento y tener una idea de que coste asociado tiene su realización.

**ELECCION DEL COMPONENTE OPTIMO**

**CARACTERISTICAS DEL ENLACE**

Longitud de Onda :  ( nm ) Velocidad :  ( Mbps )

Distancia :  ( Km ) Margen de Seguridad:  ( dB )

Número de Regeneradores :

**CARACTERISTICAS DEL CONECTOR**

Tipo :  Perdidas :  ( dB ) Precio  ( € )

**Código**

NRZ  RZ

**SELECCIONA LOS ELEMENTOS CONOCIDOS :**

TRANSMISOR  RECEPTOR  FIBRA OPTICA

**SELECCIONA LOS CRITERIOS DE EVALUACION :**

**Criterios**

COSTE DEL ENLACE  BALANCE DE POTENCIA  BALANCE DE TIEMPO

FIGURA 5.25. Formulario empleado para la selección del componente óptimo.



COMPONENTES ANALIZADOS

laser-prueba-fibra-prueba-pinprueba  
led-prueba-fibra-prueba-pinprueba

CARACTERÍSTICAS DEL ENLACE:

Enlace punto a punto a  $\lambda=850\text{nm}$   
Distancia a enlazar 5Kms  
Velocidad de transmisión 40Mb/s código NRZ  
Número de repetidores optoelectrónicos 0  
Los conectores empleados tienen unas pérdidas de 1dB  
y el Margen de Seguridad utilizado es de 6dB

RESULTADOS DEL ANÁLISIS:

Balance de Potencia  
 $Pot_{Tx} \geq Pot_{Rx} + 2 At_{Conector} + At_{Fibra} \times Long_{Max} + MargSeg$   
 $0 \geq -42 + 2 \times 1 + 3,5 \times BalPotencia.Lmax + 6$

Balance de Tiempos  
 $Tfo^2 = Tmat^2 = (Disp * Long * (Lenl / (1 + NRegt)))^2$   
 $1,00000002980232 = (0,1 * 2 * (5 / (1 + 0)))^2$   
 $Tresp^2 = Ttx^2 + Trx^2 + Tfo^2$   
 $17,0625000298023 = 0,0625 + 16 + 1,00000002980232$   
NRZ =>  $Tresp < 0,7/Velocidad$  RZ =>  $Tresp < 0,35/Velocidad$   
NRZ =>  $4,13067791407201 < 0,7/0,001 * 40 = 17,5$

Presupuesto para el Enlace de Fibra seleccionado 180 €

FIGURA 5.26. Formulario empleado para mostrar los resultados obtenidos.



### Controles utilizados en el formulario Selección del Componente Optimo

El título asignado al formulario en su Propiedad **Nombre** es ComponenteOptimo. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Etiqueta	Caption Name	ENLACE Label2
Etiqueta	Caption Name	Longitud de Onda : Label3
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label7
Etiqueta	Caption Name	Velocidad : Label4
Caja de Texto	Name Text	Velocidad (nada)
Etiqueta	Caption Name	( Mbps ) Label9
Etiqueta	Caption Name	Distancia : Label5
Caja de Texto	Name Text	Distancia (nada)
Etiqueta	Caption Name	( Kms ) Label8
Etiqueta	Caption Name	Número de Regeneradores : Label6
Caja de Texto	Name Text	NRg (nada)
Etiqueta	Caption Name	Margen de Seguridad : Label10
Caja de Texto	Name Text	MargS (nada)
Etiqueta	Caption Name	( dB ) Label11
Etiqueta	Caption Name	CONECTOR Label1
Etiqueta	Caption Name	Tipo : Label20
Caja de Texto	Name Text	TipoConector (nada)
Etiqueta	Caption Name	Pérdidas : Label13
Caja de Texto	Name Text	Perdidas (nada)
Etiqueta	Caption Name	( dB ) Label15
Etiqueta	Caption Name	Precio : Label14
Caja de Texto	Name Text	Precio (nada)



Etiqueta	Caption Name	( € ) Label16
Etiqueta	Caption Name	Código : Label12
	Name Text	NRZ (nada)
	Name Text	RZ Label13
Etiqueta	Caption Name	SELECCIONA LOS ELEMENTOS CONOCIDOS : Label17
Casilla de verificación	Caption Name	TRANSMISOR CheckTx
Casilla de verificación	Caption Name	RECEPTOR CheckRx
Casilla de verificación	Caption Name	FIBRA OPTICA CheckFO
Lista	Name Sorted	Combo1 True
Lista	Name Sorted	Combo2 True
Lista	Name Sorted	Combo3 True
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data1 Transmisor C:\WINDOWS\Escritorio\ProyectoVB\Emisores.mdb Emisores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data2 Receptor C:\WINDOWS\Escritorio\ProyectoVB\Receptores.mdb Receptores False
Control de Datos	Name Caption DatabaseName RecordSource Visible	Data3 Fibra Optica C:\WINDOWS\Escritorio\ProyectoVB\Fibras.mdb Fibras False
Etiqueta	Caption Name	SELECCIONA LOS CRITERIOS DE EVALUACION : Label18
Botón de selección	Caption Name	COSTE DEL ENLACE costenl
Botón de selección	Caption Name	BALANCE DE POTENCIA bpot
Botón de selección	Caption Name	BALANCE DE TIEMPO btime
Botón de pulsación	Caption Name	CALCULAR Command1
Botón de pulsación	Caption Name	GUARDAR GuardarComo

**TABLA 5.13.** Tabla de controles y propiedades asociadas al formulario ComponenteOptimo.



## Descripción del código fuente

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnodeCOpt* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad Index. El valor de índice del último miembro agregado es el valor de la propiedad Count de la colección Nodes.

Dim resulnodeCOpt As Node  
Dim m As Integer

Quando el formulario es cargado ( Private Sub Form\_Load() ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción Set con una variable de tipo Node ( *resulnodeCOpt*). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad Text, asignándole el nombre que queremos que tenga el nodo (COMPONENTES ANALIZADOS).

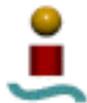
```
Private Sub Form_Load()  
m = 0  
Set resulnodeCOpt = ResultadosCOpt.TreeView1.Nodes.Add()  
resulnodeCOpt.Text = "COMPONENTES ANALIZADOS"  
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub LOnda_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOnda) Then  
Cancel = True  
MsgBox "La longitud de onda debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub MargS_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(MargS) Then  
Cancel = True  
MsgBox "El margen de seguridad debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub NRg_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(NRg) Then  
Cancel = True  
MsgBox "El valor numérico de regeneradores debe ser un valor numérico"  
End If  
End Sub
```



```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub

Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub

Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub

Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```

La función Command1\_Click() se ejecuta cuando el usuario hace "click" sobre el botón Calcular. Dada la extensión de esta función se irán introduciendo comentarios intercalándolos con el código fuente. Las primeras líneas de la función se corresponden con la definición de variables locales, que serán utilizadas en el entorno de la función.

```
Private Sub Command1_Click()
Dim balp As Boolean, balt As Boolean, presup As Single
Dim ptx As Single, prx As Single, afib As Single, acon As Single
Dim MSeg As Single, lenl As Single, NRegt As Single, vel As Single
Dim disp As Single, ilong As Single, ttx As Single, trx As Single
Dim Ctx As Single, cadc As Single, Crx As Single, campf As Single, ccon As Single
Dim cod As Boolean, Criterio As Boolean
Dim cfib As Single, lmax As Single, presupf As Single, distmax As Single, velrest As Single
Dim i As Integer, j As Integer, k As Integer, a As Integer, b As Integer, c As Integer
Dim r As Integer, s As Integer, t As Integer
Dim listatx() As String, listarx() As String, listafo() As String
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominx As Integer, lominfo As Integer
Dim LOenl As Integer
Dim vb As Integer, causa As String
```



Para evitar iniciar los cálculos sin haber seleccionado algún elemento de la base de datos de transmisores, introducimos la siguiente línea de código.

```
If Combo1.Text = "" And CheckTx.Value = 1 Then
  Data1.Recordset.MoveFirst
  Combo1.Text = Data1.Recordset.Fields("Modelo")
  MsgBox "Debe seleccionar un elemento de la lista de transmisores"
End If
```

Lo mismo ocurre con los elementos de las bases de datos de receptores y fibras ópticas.

```
If Combo2.Text = "" And CheckRx.Value = 1 Then
  Data2.Recordset.MoveFirst
  Combo2.Text = Data2.Recordset.Fields("Modelo")
  MsgBox "Debe seleccionar un elemento de la lista de receptores"
End If
If Combo3.Text = "" And CheckFO.Value = 1 Then
  Data3.Recordset.MoveFirst
  Combo3.Text = Data3.Recordset.Fields("Modelo")
  MsgBox "Debe seleccionar un elemento de la lista de fibras ópticas"
End If
```

Para identificar el elemento seleccionado para realizar los cálculos entre los elementos de cada una de las bases de datos ( Transmisores-Receptores-Fibras Ópticas ), dispondremos de un objeto Combo, enlazado con un objeto Data, para acceder a la base de datos.

Al iniciar la función, situamos cada uno de los objetos Data al principio de la base de datos correspondiente y comprobamos que existe algún elemento en la base de datos. En este caso, y hasta que se llegue al último elemento de la base de datos comparamos cada uno de los elementos de la base de datos con el elemento que ha sido seleccionado para realizar los cálculos ( de él hasta ahora tan sólo conocemos el campo "Modelo" que nos da información acerca del nombre del elemento, pero no disponemos de información del resto de sus características hasta que lo ubiquemos en la base de datos ). Esta operación se realiza en cada una de las tres bases de datos que tenemos.

```
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
  Do Until Data1.Recordset.EOF
  If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then
  If Data2.Recordset.RecordCount > 0 Then
  Do Until Data2.Recordset.EOF
  If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo2.Text)) Then
  If Data3.Recordset.RecordCount > 0 Then
  Do Until Data3.Recordset.EOF
  If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo3.Text)) Then
```

Una vez identificados el transmisor, el receptor y la fibra óptica, asociamos el valor correspondiente a cada una de las variables definidas. El primer grupo de variables son pasadas como parámetro a la función *Compatibilidad* para analizar si son compatibles los elementos seleccionados con las características del enlace, basándonos para ello en dos parámetros: la longitud de onda del enlace y los tipo de conectores empleados.

```
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
```



```
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorr, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```

Si resultan compatibles los elementos seleccionados con las características del enlace, inicializamos cada una de las variables definidas, requeridas para los cálculos realizados por la función.

Se asigna el valor de cada una de las variables definidas en la función y que describen las características del enlace, del transmisor, del receptor, de la fibra óptica y de los conectores utilizados.

```
If vb = 0 Then
    ptx = Data1.Recordset.Fields("Potencia")
    prx = Data2.Recordset.Fields("Potencia")
    ilong = Data1.Recordset.Fields("Incremento_LOnda")
    ttx = Data1.Recordset.Fields("Tiempo_Respuesta")
    trx = Data2.Recordset.Fields("Tiempo_Respuesta")
    Ctx = Data1.Recordset.Fields("Precio")
    Crx = Data2.Recordset.Fields("Precio")
    cadc = Data1.Recordset.Fields("Precio_CtoAdicional")
    campf = Data2.Recordset.Fields("Precio_CtoAmplificador")
    ccon = Data3.Recordset.Fields("Precio/metro")
    afib = Data3.Recordset.Fields("Atenuacion")
    disp = Data3.Recordset.Fields("Dispersion")
    vel = Velocidad.Text
    lenl = Distancia.Text
    MSeg = MargS.Text
    NRegt = NRg.Text
    ccon = Precio.Text
    acon = Perdidas.Text
    If NRZ.Value = True Then
        cod = False
    Else
        cod = True
    End If
```

Para evaluar el balance de tiempos se aplicará la ecuación de evaluación del tiempo de subida, que nos dice que el tiempo de subida del sistema al cuadrado, expresado en nanosegundos, lo podemos calcular como la suma de los cuadrados de los tiempos de subida del generador, del receptor y de la fibra.



En función del tipo de código empleado, el tiempo de subida multiplicado por la velocidad de transmisión deberá ser menor que un determinado coeficiente:

```
RZ => Tr x B < 0.35   NRZ => Tr x B < 0.7

TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
Tres = Sqr(TresCuad)
If cod = False Then
cdg = 0.7
codigo = "NRZ"
Else
cdg = 0.35
codigo = "RZ"
End If
restvel = Abs(vel - (cdg / (0.001 * Tres)))
```

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

```
m = m + 1
```

En función de que tengamos seleccionado o no un elemento concreto, transmisor-receptor-fibra óptica, consideraremos sólo el elemento seleccionado para realizar los cálculos del enlace o todos los elementos de ese tipo almacenados en la correspondiente base de datos. En otras palabras, si he adquirido un transmisor y un receptor para implementar mi enlace de fibra y me falta por adquirir la fibra óptica, introduciría como conocidos el transmisor y el receptor, y utilizaría para el cálculo del enlace la base de datos de fibras ópticas que tengo disponible. Para realizar esto introducimos las siguientes líneas de código:

```
'Si el transmisor es marcado como seleccionado
If CheckTx.Value = 1 Then
  i = 1
  ReDim listatx(i - 1)
  listatx(i - 1) = Data1.Recordset.Fields("Modelo")
'En caso contrario
Else
  'Calcula el número de elementos que hay almacenados en la base de datos
  Data1.Recordset.MoveFirst
  i = 1
  While Data1.Recordset.EOF = False
    i = i + 1
    Data1.Recordset.MoveNext
  Wend
  'Especifica el tamaño de la lista donde voy a almacenar los nombres de esos
  elementos
  ReDim listatx(i - 1)
  'Rellena la lista con el campo "Modelo" de la base de datos de transmisores
  Data1.Recordset.MoveFirst
  For a = 0 To i - 2
    listatx(a) = Data1.Recordset.Fields("Modelo")
    If Data1.Recordset.EOF = False Then
      Data1.Recordset.MoveNext
```



```
Else
  a = i - 1
End If
Next a
End If
```

'Lo mismo para el receptor

```
If CheckRx.Value = 1 Then
  j = 1
  ReDim listarx(j - 1)
  listarx(j - 1) = Data2.Recordset.Fields("Modelo")
Else
  Data2.Recordset.MoveFirst
  j = 1
  While Data2.Recordset.EOF = False
    j = j + 1
    Data2.Recordset.MoveNext
  Wend
  ReDim listarx(j - 1)
  Data2.Recordset.MoveFirst
  For b = 0 To j - 2
    listarx(b) = Data2.Recordset.Fields("Modelo")
    If Data2.Recordset.EOF = False Then
      Data2.Recordset.MoveNext
    Else
      b = j - 1
    End If
  Next b
End If
```

'Lo mismo para la fibra óptica

```
If CheckFO.Value = 1 Then
  k = 1
  ReDim listafo(k - 1)
  listafo(k - 1) = Data3.Recordset.Fields("Modelo")
Else
  Data3.Recordset.MoveFirst
  k = 1
  While Data3.Recordset.EOF = False
    k = k + 1
    Data3.Recordset.MoveNext
  Wend
  ReDim listafo(k - 1)
  Data3.Recordset.MoveFirst
  For c = 0 To k - 2
    listafo(c) = Data3.Recordset.Fields("Modelo")
    If Data3.Recordset.EOF = False Then
      Data3.Recordset.MoveNext
    Else
      c = k - 1
    End If
  Next c
End If
```



Llamo a la función *BalancePotencia* que me devolverá una variable lógica, en función de la cuál decidiré si paso al siguiente paso en la evaluación del enlace, o si por el contrario no se cumple el balance de potencia, terminando aquí el análisis del enlace e imprimiendo un mensaje que describe la causa.

```
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
If balp = False Then
    lmax = (ptx - (prx + 2 * acon + MSeg)) / afib

resulEnl(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    vbCrLf & _
    "Los conectores empleados tienen unas pérdidas de " & acon & "dB"
    & vbCrLf & _
    "y el Margen de Seguridad utilizado es de " & MSeg & "dB" &
    vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    vbCrLf & _
    "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " &
vbCrLf & _
    vbCrLf & _
    ptx & " >=" & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax"
    & " + " & MSeg & vbCrLf & _
    "El Balance de Potencias ha resultado negativo" & vbCrLf & _
    "debiendo reducirse la distancia del enlace en " & Abs(lmax - lenl) & "Kms"
    "Falta dirigirlo al final de la funcion
GoTo fin
End If
```

A continuación llamo a la función *BalanceTiempo* que me devolverá una variable lógica, en función de la cuál decidiré si paso al siguiente paso en la evaluación del enlace, o si por el contrario no se cumple el balance de tiempo, terminando aquí el análisis del enlace e imprimiendo un mensaje que describe la causa.

```
balt = baltime(displ, ilong, lenl, ttx, trx, vel, cod, NRegt)
If balt = False Then
    restvel = Abs(vel - (cdg / (0.001 * Tres)))
    resulEnl(m) = vbCrLf & _
        "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
        vbCrLf & _
        "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
        "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
        "Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
        "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
        vbCrLf & _
        "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
        vbCrLf & _
        "Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
        TmatCuad & " = (" & displ & " * " & ilong & " * (" & lenl & " / (1 + " & NRegt &
    " )) ^ 2" & vbCrLf & _
        "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
        TresCuad & " = " & (ttx ^ 2) & " + " & (trx ^ 2) & " + " & TmatCuad & vbCrLf
    & _
```



```
"NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf & _  
& _  
    codigo & " => " & Sqr(TresCuad) & " < " & cdg & "/0.001*" & vel & " = " &  
cdg / (vel * 0.001) & vbCrLf & _  
    vbCrLf & _  
    "El Balance de Tiempo ha resultado negativo" & vbCrLf & _  
    "debiendo reducirse la velocidad del enlace en " & restvel & "Mb/s"  
  
GoTo fin  
End If
```

Finalmente, una vez que se ha comprobado que los elementos que constituyen el enlace son compatibles, y que se verifica el balance de potencia y de tiempo, tan sólo resta por realizar la evaluación económica de su implementación, expresada en Euros.

```
presup = coste(Ctx, cadc, Crx, campf, ccon, lenl, cfib, NRegt)  
presupf = 100000000  
If presup < presupf Then  
    presupf = presup  
    resulEnl(0) = vbCrLf & _  
    vbCrLf & _  
    "EL enlace óptimo es" & vbCrLf & _  
    Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &  
Data2.Recordset.Fields("Modelo") & _  
    vbCrLf & _  
    "con un coste de " & presupf & " €" & vbCrLf & _  
    "requiriendo la solución " & NRegt & " nodos intermedios"  
End If
```

En la variable presupf se irá almacenando el presupuesto más económico de los presupuestos que resulten implementables, y que podremos consultar pinchando sobre el nodo raíz del árbol de resultados.

Por último, tendré que almacenar el resultado obtenido en una variable tipo Cadena de Caracteres, de forma que cuando seleccione el nodo correspondiente en la estructura de árbol, se muestre en el cuadro de texto habilitado para ello el resultado correspondiente. Con este fin se ha definido una variable de carácter Público ( para que sea reconocida en todo el módulo ), como una lista de Cadena de Caracteres, de forma que cada elemento de la lista contendrá el resultado correspondiente al nodo referenciado.

```
resulEnl(m) = vbCrLf & _  
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _  
vbCrLf & _  
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _  
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _  
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _  
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _  
"Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _  
"y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _  
vbCrLf & _  
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _  
vbCrLf & _  
"Balance de Potencia" & vbCrLf & _  
"Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf & _  
ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + " &  
MSeg & vbCrLf & _  
vbCrLf & _  
"Balance de Tiempos" & vbCrLf & _  
"Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
```



```
TmatCuad & " = (" & disp & "*" & ilong & "*" & lenl & "/" & (1 + " & NRegt & "))) ^ 2"
"Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
TresCuad & " = " & (ttx ^ 2) & " + " & (trx ^ 2) & " + " & TmatCuad & vbCrLf & _
"NRZ => Tresp < 0.7/Velocidad  RZ => Tresp < 0.35/Velocidad" & vbCrLf & _

codigo & " => " & Sqr(TresCuad) & " < " & cdg & "/0.001*" & vel & " = " & cdg / (vel *
0.001) & vbCrLf & _
vbCrLf & _
"Presupuesto para el Enlace de Fibra seleccionado " & presup & " €"

fin:
End If
```

En el caso de no ser compatible el enlace con alguno de los elementos que lo constituyen, se mostraría un mensaje de error identificando la causa y el elemento que lo ocasiona.

```
If vb <> 0 Then
Select Case vb
Case 1
causa = "del tipo de conector usado para la fibra óptica"
Case 2
causa = "del tipo de conector usado para el transmisor"
Case 3
causa = "del tipo de conector usado para el receptor"
Case 4
causa = "de la longitud de onda del transmisor"
Case 5
causa = "de la longitud de onda del receptor"
Case 6
causa = "de la longitud de onda de la fibra óptica"
End Select
MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
End If
```

Se terminan los bucles y los condicionales abiertos al principio de la función.

```
End If
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
Next t
Next s
Next r
```

Ejecutamos la función ResultadosCOpt que nos muestra los resultados obtenidos.

```
ResultadosCOpt.Show
End Sub
```



La función *GuardarComo* se utiliza para generar un fichero con la extensión *.enl* que será donde almacenemos los datos empleados para la realización de un determinado cálculo.

```
Private Sub GuardarComo_Click()  
Dim path As String  
Dim puntero  
GuardarCOP.TipoSeleccion = Index  
GuardarCOP.Show vbModal  
If Trim(GuardarCOP.Retorna) <> "" Then  
    path = GuardarCOP.Retorna  
End If  
puntero = FreeFile  
If path <> "" Then  
    Open path For Output As #puntero  
    Write #puntero, LOnda.Text, Velocidad.Text, Distancia.Text, NRg.Text, MargS.Text, _  
    TipoConector.Text, Perdidas.Text, Precio.Text, NRZ.Value, _  
    CheckTx.Value, CheckRx.Value, CheckFO.Value, Combo1.Text, Combo2.Text,  
    Combo3.Text, _  
    costenI.Value, bpot.Value, btime.Value  
End If  
Close #puntero  
End Sub
```

El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario de Balance de Potencia.

```
Private Sub Form_Unload(Cancel As Integer)  
Unload ResultadosCopt  
End Sub
```



## CÁLCULO DEL TIEMPO DE SUBIDA PARA UNA FIBRA ÓPTICA

### Descripción del formulario

Este formulario nos permite calcular el tiempo total de subida de una fibra óptica, que se obtendrá a partir de la ecuación:

$$T^2_{\text{Fibra(ns)}} = T^2_{\text{Sistema(ns)}} - T^2_{\text{Generador(ns)}} - T^2_{\text{Detector(ns)}}$$

**Tiempo de Subida de la Fibra Óptica**

### CALCULO DEL TIEMPO DE SUBIDA DE LA FIBRA

Longitud de Onda de trabajo:  ( nm )

Modelo

Dispersión cromática :  ( ns/nm x Km )

Dispersión cromática material :  ( ns/nm x Km )

Dispersión cromática guía-onda :  ( ns/nm x Km )

Anchura espectral del generador de luz :  ( nm )

Ancho de banda modal de la fibra óptica :  ( MHz x Km )

**CALCULAR** **GUARDAR**

**FIGURA 5.27.** Formulario empleado para el cálculo del tiempo de subida de la fibra óptica.

El tiempo de subida del generador de luz y del detector es el tiempo en nanosegundos que necesita una señal de entrada con forma de onda en escalón para subir entre el 10 y el 90 por 100 de la amplitud total media a la salida. El miembro derecho de la ecuación puede considerarse como el tiempo de subida del equipo.



El tiempo total de subida de la fibra óptica se estimará de una forma u otra dependiendo de si la fibra es multimodo o monomodo. El ancho de banda de transmisión por fibras multimodo está limitado por parámetros de la fibra óptica, como la dispersión modal y la dispersión cromática. El ancho de banda de transmisión por fibras monomodo está limitado por la dispersión cromática material y por la dispersión cromática guía-onda.

Multimodo:  $T^2_{Fibra(ns)} = T^2_{Modal(ns)} + T^2_{Cromática(ns)}$

Monomodo:  $T^2_{Fibra(ns)} = T^2_{Cromática(ns)}$



FIGURA 5.28. Formulario empleado para mostrar los resultados obtenidos.



### Controles utilizados en el formulario Tiempo de Subida para una Fibra Óptica

El título asignado al formulario en su Propiedad **Nombre** es Tsubida. Su Propiedad **BorderStyle** ha sido puesta a 3 (Fixed Dialog), para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

OBJETO	PROPIEDAD	VALOR
Botón de pulsación	Caption Name	Longitud de Onda de trabajo: Command1
Caja de Texto	Name Text	LOnda (nada)
Etiqueta	Caption Name	( nm ) Label2
Etiqueta	Caption Name	Modelo Label1
Caja de Texto	Name Text	Modelo (nada)
Etiqueta	Caption Name	Dispersión cromática : Label3
Caja de Texto	Name Text	DispCrom (nada)
Etiqueta	Caption Name	( ns/nm x Km ) Label4
Etiqueta	Caption Name	Dispersión cromática material : Label5
Caja de Texto	Name Text	DispMat (nada)
Etiqueta	Caption Name	( ns/nm x Km ) Label6
Etiqueta	Caption Name	Dispersión cromática guía-onda : Label7
Caja de Texto	Name Text	DispGuia (nada)
Etiqueta	Caption Name	( ns/nm x Km ) Label8
Etiqueta	Caption Name	Anchura espectral del generador de luz : Label11
Caja de Texto	Name Text	AnchEsp (nada)
Etiqueta	Caption Name	( nm ) Label12
Etiqueta	Caption Name	Ancho de banda modal de la fibra óptica : Label13
Caja de Texto	Name Text	AnchMod (nada)
Etiqueta	Caption Name	( MHz x Km ) Label14
Botón de pulsación	Caption Name	CALCULAR Command1
Botón de pulsación	Caption Name	GUARDAR GuardarComo

**TABLA 5.14.** Tabla de controles y propiedades asociadas al formulario Tsubida.



## Descripción del código fuente

Defino las variables que van a ser globales dentro de este formulario. Estas variables son dos: la variable *resulnodeEnl* que es una variable tipo Node que forma parte de la estructura jerárquica del Control Treeview, y la variable *m* que es definida como un entero y que se utilizará como índice en la colección de objetos Node.

A medida que se agrega un objeto Node se le asigna un número de índice que se almacena en su propiedad Index. El valor de índice del último miembro agregado es el valor de la propiedad Count de la colección Nodes.

```
Dim resulnodeTs As Node
Dim m As Integer
```

Cuando el formulario es cargado ( Private Sub Form\_Load() ) se creará el primer nodo de la estructura jerárquica de resultados, para lo cual se empleará la instrucción Set con una variable de tipo Node ( *resulnodeEnl* ). Una vez creado el primer nodo, tendremos que darle un valor para lo que utilizamos la Propiedad Text, asignándole el nombre que queremos que tenga el nodo (TIEMPO DE RESPUESTA DE LA FIBRA OPTICA).

```
Private Sub Form_Load()
    m = 0
    Set resulnodeTs = ResultadosTs.TreeView1.Nodes.Add()
    resulnodeTs.Text = "TIEMPO DE RESPUESTA DE LA FIBRA OPTICA"
End Sub
```

A la hora de introducir los distintos valores solicitados en el formulario podríamos cometer un error, y por ejemplo, no introducir un valor numérico en caso de que el valor solicitado fuera de este tipo. Para evitar este error se han implementado las siguientes funciones, que en el caso de introducir un valor incorrecto en la casilla habilitada para tal efecto, se produciría un mensaje de error y no nos permitiría continuar hasta que el nuevo valor fuera introducido.

```
Private Sub AnchEsp_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(AnchEsp) Then
        Cancel = True
        MsgBox "La anchura espectral del generador de luz debe ser un valor numérico"
    End If
End Sub
```

```
Private Sub AnchMod_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(AnchMod) Then
        Cancel = True
        MsgBox "El ancho de banda modal de la fibra óptica debe ser un valor numérico"
    End If
End Sub
```

```
Private Sub DispCrom_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(DispCrom) Then
        Cancel = True
        MsgBox "La dispersión cromática de la fibra debe ser un valor numérico"
    End If
End Sub
```



```
Private Sub DispGuia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(DispGuia) Then
    Cancel = True
    MsgBox "La dispersión cromática guía-onda debe ser un valor numérico"
End If
End Sub
```

```
Private Sub DispMat_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(DispMat) Then
    Cancel = True
    MsgBox "La dispersión cromática material debe ser un valor numérico"
End If
End Sub
```

```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
    Cancel = True
    MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```

La siguiente función se ejecuta cuando pulsamos el botón de Longitud de onda, y lo que hace es habilitar las casillas que tendremos que rellenar para calcular el tiempo de respuesta en función del caso en que nos encontremos:

```
Private Sub Command1_Click()
If LOnda.Text = "850" Then
    DispMat.Text = ""
    DispGuia.Text = ""
    DispMat.Enabled = False
    DispGuia.Enabled = False
    DispCrom.Enabled = True
    AnchMod.Enabled = True
Else
    DispCrom.Text = ""
    AnchMod = ""
    DispCrom.Enabled = False
    AnchMod.Enabled = False
    DispMat.Enabled = True
    DispGuia.Enabled = True
End If
End Sub
```



Con la función Calcular podemos determinar el valor del Tiempo de subida requerido según las siguientes ecuaciones:

```
Private Sub Calcular_Click()  
Dim Tfo As Single, Tcrom As Single, Wesp As Single  
Wesp = AnchEsp.Text  
If LOnda.Text = "850" Then  
Dim Tmod As Single, Dcrom As Single  
Dim Bmodelec As Single, Bmodinst As Single, Bmod As Single  
Bmod = AnchMod.Text  
Dcrom = DispCrom.Text  
Bmodelect = 0.71 * Bmod  
Tmod = 0.35 / Bmodelect  
Tcrom = Dcrom * Wesp  
Tfo = Sqr((Tmod ^ 2) + (Tcrom ^ 2))  
Else  
Dim Dmat As Single, Dgo As Single, Dcromat As Single  
Dmat = DispMat.Text  
Dgo = DispGuia.Text  
Dcromat = Dmat + Dgo  
Tcrom = Dcromat * Wesp  
Tfo = Tcrom  
End If
```

Defino el siguiente nodo del árbol de nodos utilizado para mostrar los resultados obtenidos.

```
m = m + 1
```

Por último, tendré que almacenar el resultado obtenido en una variable tipo Cadena de Caracteres, de forma que cuando seleccione el nodo correspondiente en la estructura de árbol, se muestre en el cuadro de texto habilitado para ello el resultado correspondiente. Con este fin se ha definido una variable de carácter Público ( para que sea reconocida en todo el módulo ), como una lista de Cadena de Caracteres, de forma que cada elemento de la lista contendrá el resultado correspondiente al nodo referenciado.

```
Set resulnodeTs = ResultadosTs.TreeView1.Nodes.Add(m)  
resulnodeTs.Text = Modelo.Text  
If LOnda.Text = "850" Then  
resulTs(m) = "FIBRAS MULTIMODO" & vbCrLf & _  
"Bmodalelect = 0.71 * Bmodal" & vbCrLf & _  
"Bmodalelect = 0.71*" & Bmod & " = " & (0.71 * Bmod) & vbCrLf & _  
"Tmodal = 0.35 / Bmodalelect" & vbCrLf & _  
"Tmodal = 0.35/" & Bmodelect & " = " & (0.35 / Bmodelect) & vbCrLf & _  
"Tcromatica = Dispcromatica * Wspectral" & vbCrLf & _  
"Tcromatica = " & Dcrom & "*" & Wesp & " = " & (Dcrom * Wesp) & vbCrLf & _  
"Tfibra^2 = Tmodal^2 + Tcromatica^2" & vbCrLf & _  
Tfo ^ 2 & " = " & Tmod ^ 2 & " + " & Tcrom ^ 2 & " => Tfibra = " & Tfo & "ns"
```



```
Else
    resulTs(m) = "FIBRAS MONOMODO" & vbCrLf & _
        "Dispcromat = Dispmat + Dispguia" & vbCrLf & _
        "Dispcromat = " & Dmat & " + " & Dgo & " = " & (Dmat + Dgo) & vbCrLf & _
        "Tcromatica = Dispcromatica * Wespectral" & vbCrLf & _
        "Tcromatica = " & Dcromat & "*" & Wesp & " = " & (Dcromat * Wesp) &
vbCrLf & _
        "Tfibra^2 = Tcromatica^2" & vbCrLf & _
        "Tfibra^2 = " & Tcrom ^ 2 & " => Tfo = " & Tfo & "ns"

End If
```

Ejecutamos la función ResultadosCOpt que nos muestra los resultados obtenidos.

```
ResultadosTs.Show
End Sub
```

La función *GuardarComo* se utiliza para generar un fichero con la extensión *.enl* que será donde almacenemos los datos empleados para la realización de un determinado cálculo.

```
Private Sub GuardarComo_Click()
    Dim path As String
    Dim puntero
    GuardarTSB.TipoSeleccion = Index
    GuardarTSB.Show vbModal
    If Trim(GuardarTSB.Retorna) <> "" Then
        path = GuardarTSB.Retorna
    End If
    puntero = FreeFile
    If path <> "" Then
        Open path For Output As #puntero
        Write #puntero, LOnda.Text, Modelo.Text, DispCrom.Text, DispMat.Text,
        DispGuia.Text, _
        Distancia.Text, AnchEsp.Text, AnchMod.Text
    End If
    Close #puntero
End Sub
```

El módulo concluye con la función *Form\_Unload* que se utiliza para cerrar la pantalla de resultados una vez que hemos cerrado el formulario de Balance de Potencia.

```
Private Sub Form_Unload(Cancel As Integer)
    Unload ResultadosTs
End Sub
```



## CASO PRÁCTICO

### EJERCICIO

Una empresa de comunicaciones ópticas está al borde de la quiebra por culpa de anteriores diseños mal calculados. ( NOTA: los ingenieros responsables fueron despedidos ). Se contrata a un alumno de último curso de la escuela para que haga un último intento por salvar la empresa, diseñando un enlace con las características óptimas.

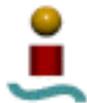
Características del enlace:

- Enlace punto a punto a  $\lambda = 850\text{nm}$
- Distancia a enlazar, 5 Kms, pudiéndose incluir repetidores optoelectrónicos en tramos intermedios en caso necesario.
- Velocidad de transmisión  $B = 40\text{Mb/s}$ , código NRZ.

Se dispone de:

1. Un LED a 850nm,  $P_{tr} = -13\text{dBm}$ ,  $\Delta\lambda = 50\text{nm}$ ,  $T_r = 3\text{ns}$ , Precio = 12.02€
2. Un Laser a 850nm,  $P_{tr} = 0\text{dBm}$ ,  $\Delta\lambda = 2\text{nm}$ ,  $T_r = 0.25\text{ns}$ , Precio = 120.2€
3. Un diodo PIN,  $P_{rec} = -42\text{dBm}$ ,  $T_r = 4\text{ns}$ , Precio = 6.01€
4. Conectores para fibra,  $P_{con} = 1\text{dB}$ , Precio = 6.01€
5. Circuitería driver para LED, Precio = 12.02€
6. Circuitería driver para Laser, Precio = 30.05€
7. Circuitería amplificadora para diodo PIN, Precio = 12.02€
8. Fibra MM con características dispersivas genéricas  $D = 0.1 \text{ ns/nm} \cdot \text{Km}$  y atenuación  $\alpha = 3.5 \text{ dB/Km}$  ( NOTA: Margen de Seguridad 6dB )

Suponiendo que fueses ese alumno, analiza las soluciones posibles y elige la solución más rentable.



## SOLUCIÓN OBTENIDA APLICANDO LAS ECUACIONES

Analizamos primero la solución sin repetidores optoelectrónicos intermedios:

### LED-PIN

Balance de Potencias:  $Pot_{Tx} \geq Pot_{Rx} + 2 * \alpha_c + \alpha_{fo} * L_{enl} + MS$   
 $-13 \geq -42 + 2 * 1 + 3.5 * L_{max} + 6 \Rightarrow L_{max} = 6 \text{ Km} > L = 5\text{Km}$

Por lo que el Balance de Potencias se verifica.

Balance de Tiempos: Fibra MM, approximo el tiempo de subida de la fibra como

$$T_F^2 = T_{MAT}^2 = (0.1 * 50 * 5)^2 = 625$$
$$T_r^2 = T_{tr}^2 + T_{rec}^2 + T_F^2 = 9 + 16 + 625 = 650 \Rightarrow T_r = 25.49\text{ns}$$

Código NRZ  $\Rightarrow T_r < 0.7/B$

$$25.49 * 10^{-9} * 40 * 10^6 = 1.02 > 0.7$$

Por lo tanto el Balance de Tiempo no se verifica.

Queda descartada la solución LED-PIN al no verificarse el Balance de Tiempos.

### LASER-PIN

Balance de Potencias:  $Pot_{Tx} \geq Pot_{Rx} + 2 * \alpha_c + \alpha_{fo} * L_{enl} + MS$   
 $0 \geq -42 + 2 * 1 + 3.5 * L_{max} + 6 \Rightarrow L_{max} = 9.71 \text{ Km} > L = 5\text{Km}$

Por lo que el Balance de Potencias se verifica.

Balance de Tiempos: Fibra MM, approximo el tiempo de subida de la fibra como

$$T_F^2 = T_{MAT}^2 = (0.1 * 2 * 5)^2 = 1$$
$$T_r^2 = T_{tr}^2 + T_{rec}^2 + T_F^2 = 0.0625 + 16 + 1 = 17.0625 \Rightarrow$$

$$T_r = 4.1307\text{ns}$$

Código NRZ  $\Rightarrow T_r < 0.7/B$

$$4.1307 * 10^{-9} * 40 * 10^6 = 0.165 > 0.7$$

Por lo tanto el Balance de Tiempos se verifica.



Coste del enlace:  $\text{Coste} = 2 * \text{Precio\_conector} + \text{Precio\_emisor} + \text{Precio\_driver\_emisor} + \text{Precio\_receptor} + \text{Precio\_amplificador}$   
 $\text{Coste} = 2 * 6.01 + 120.2 + 30.05 + 6.01 + 12.02 = 180.75\text{€}$

Analizamos ahora la solución con repetidores optoelectrónicos intermedios:

### LED-PIN

NR = 1

Balance de Potencias:  $\text{Pot}_{\text{Tx}} \geq \text{Pot}_{\text{Rx}} + 2 * \alpha_c + \alpha_{\text{fo}} * L_{\text{enl}} + \text{MS}$   
 $-13 \geq -42 + 2 * 1 + 3.5 * L_{\text{max}} + 6 \Rightarrow$   
 $L_{\text{max}} = 6 \text{ Km} > L / (1 + \text{NR}) = 5/2 \text{ Km} = 2.5 \text{ Km}$   
Por lo que el Balance de Potencias se verifica.

Balance de Tiempos: Fibra MM, aproximamos el tiempo de subida de la fibra como

$$T_F^2 = T_{\text{MAT}}^2 = (0.1 * 50 * 2.5)^2 = 156.25$$
$$T_r^2 = T_{\text{tr}}^2 + T_{\text{rec}}^2 + T_F^2 = 9 + 16 + 156.25 = 181.25 \Rightarrow$$
$$T_r = 13.46 \text{ ns}$$

Código NRZ  $\Rightarrow T_r < 0.7/B$

$$13.46 * 10^{-9} * 40 * 10^6 = 0.53 < 0.7$$

Por lo tanto el Balance de Tiempos se verifica.

Coste del enlace:  $\text{Coste} = 4 * \text{Precio\_conector} + 2 * \text{Precio\_emisor} + 2 * \text{Precio\_driver\_emisor} + 2 * \text{Precio\_receptor} + 2 * \text{Precio\_amplificador}$   
 $\text{Coste} = 4 * 6.01 + 2 * 12.02 + 2 * 12.02 + 2 * 6.01 + 2 * 12.02 = 108.18\text{€}$

Por tanto la solución óptima para el problema es la obtenida utilizando el LED y el diodo PIN incluyendo un repetidor optoelectrónico con un coste asociado de 108.18€.



## SOLUCIÓN 1 OBTENIDA UTILIZANDO EL PROGRAMA: DISEÑO DEL ENLACE

Lo primero que se tendrá que hacer será dar de alta en las bases de datos los elementos que se utilizarán para el diseño del enlace de fibra en el caso de que no estuvieran almacenados.

Comenzamos por ejemplo creando los emisores ópticos en la Base de Datos de emisores, para lo cual abrimos la Base de Datos y pulsamos sobre la opción Nuevo.

CARACTERISTICAS DEL EMISOR OPTICO	
Modelo :	LED
Tipo de conector :	SC
Longitud de Onda Mínima :	850 (nm)
Longitud de Onda Máxima :	850 (nm)
Velocidad Máxima :	40 (Mbps)
Potencia :	-13 (dBm)
Tiempo de Respuesta :	3 (ns)
Incremento de Longitud de Onda :	50 (nm)
Precio :	12,02 (€)
Precio Circuitería Adicional :	12,02 (€)

Buttons: Inicio, Anterior, Siguiete, Final, Buscar

Buttons: Nuevo, Editar, Grabar, Borrar, Cancelar, Refrescar

**FIGURA 6.1.** Creamos un nuevo elemento en la Base de Datos de Emisores al que identificaremos con el nombre de LED.

A continuación introducimos los datos solicitados para dar de alta un nuevo emisor y lo grabamos con el nombre que deseemos. Si quisiéramos realizar alguna modificación sobre alguno de los elementos de la base de datos ya creados bastará con Editar el elemento y modificar el campo susceptible a ser cambiado. No olvidar salvar los cambios pulsando cuando finalicemos el botón Grabar.



A continuación creamos el elemento LASER siguiendo los mismos pasos que para el elemento anterior.

CARACTERISTICAS DEL EMISOR OPTICO	
Modelo :	LASER
Tipo de conector :	SC
Longitud de Onda Mínima :	850 (nm)
Longitud de Onda Máxima :	850 (nm)
Velocidad Máxima :	40 (Mbps)
Potencia :	0 (dBm)
Tiempo de Respuesta :	0,25 (ns)
Incremento de Longitud de Onda :	2 (nm)
Precio :	120,2 (€)
Precio Circuitería Adicional :	30,05 (€)

Buttons: Inicio, Anterior, Siguiente, Final, Buscar

Buttons: Nuevo, Editar, Grabar, Borrar, Cancelar, Refrescar

**FIGURA 6.2.** Creamos un nuevo elemento en la Base de Datos de Emisores al que identificaremos con el nombre de LASER.

De igual modo crearemos el elemento PIN en la Base de Datos de Receptores, y el elemento FIBRA OPTICA en la Base de Datos de Fibras. Una vez creados los distintos elementos bastará con salirnos de la Base de Datos salvando las modificaciones realizadas.

A partir de este momento cada vez que necesitemos recurrir a la utilización de cualquiera de estos elementos bastará con referenciarlo por su campo Modelo desde cualquiera del resto de los formularios.



Base de Datos Receptores

### CARACTERÍSTICAS DEL RECEPTOR OPTICO

Modelo :	PIN		Nuevo
Tipo de conector :	SC		Editar
Longitud de Onda Mínima :	850	( nm )	Grabar
Longitud de Onda Máxima :	850	( nm )	Borrar
Velocidad Máxima :	40	( Mbps )	Cancelar
Sensibilidad :	-42	( dBm )	Refrescar
Tiempo de Respuesta :	4	( ns )	
Precio :	6,01	( € )	
Precio Circuitería Amplificadora :	12,02	( € )	

Inicio Anterior Siguiente Final Buscar

**FIGURA 6.3.** Creamos un nuevo elemento en la Base de Datos de Receptores al que identificaremos con el nombre de PIN.

Base de Datos Fibras Opticas

### CARACTERÍSTICAS DE LA FIBRA OPTICA

Modelo :	FIBRA OPTICA		Nuevo
Tipo de Conector :	SC		Editar
Longitud de Onda Mínima :	850	( nm )	Grabar
Longitud de Onda Máxima :	850	( nm )	Borrar
Atenuación :	3,5	( dB / Km )	Cancelar
Dispersión :	0,1	( ns / nm * Km )	Refrescar
Precio / metro :	0	( € )	

Inicio Anterior Siguiente Final Buscar

**FIGURA 6.4.** Creamos un nuevo elemento en la Base de Datos de Fibras al que identificaremos con el nombre de FIBRA OPTICA.

Abrimos la opción Diseño del Enlace( nos aparecerá la pantalla de la FIGURA 6.9 ) e introducimos los datos necesarios para el diseño del enlace. Pulsando sobre el botón



Descripción del Enlace abrimos la siguiente pantalla donde deberemos introducir los valores que caracterizarán el enlace y los conectores utilizados. Una vez introducidos todos estos datos pulsaremos el botón Volver.

CARACTERISTICAS DEL ENLACE	
Longitud de Onda	850 (nm)
Velocidad	40 (Mbps)
Distancia	5 (Km)
Margen de Seguridad	6 (dB)
Código	<input checked="" type="radio"/> NRZ <input type="radio"/> RZ
CARACTERISTICAS DEL CONECTOR	
Tipo de conector disponible	SC
Perdidas	1 (dB)
Precio	6.01 (€)
<input type="button" value="VOLVER"/>	

**FIGURA 6.5.** Introducimos los valores que caracterizarán el enlace y los conectores.

A continuación tendremos que seleccionar los elementos que deseemos utilizar para el diseño del enlace. Para ello tendremos que ir accediendo a las distintas pantallas que nos permitirán seleccionar dichos elementos. Los botones que abren dichas pantallas son Transmisores, Receptores y Fibra Óptica.



Para seleccionar un elemento bastará con referenciarlo, utilizando los controles que aparecen a la derecha de la pantalla y pulsar el botón Seleccionar.

**Selección de Emisor**

### CARACTERÍSTICAS DEL EMISOR

Modelo :

Tipo de conector :

Longitud de Onda Mínima :  (nm)

Longitud de Onda Máxima :  (nm)

Velocidad Máxima :  (Mbps)

Potencia :  (dBm)

Tiempo de Respuesta :  (ns)

Incremento Longitud Onda :  (nm)

Precio :  (€)

Precio Circuitería Adicional :  (€)

**LASER**

LED

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR EMISOR

SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

VOLVER

**FIGURA 6.6.** Seleccionamos los emisores que se utilizarán en el diseño del enlace.

Si lo que queremos es seleccionar todos los elementos disponibles en la Base de Datos tendremos que pulsar el botón Seleccionar Todos. Si deseamos eliminar algún elemento de los seleccionados bastará con marcarlo en la lista y pulsar el botón Eliminar. Para limpiar la lista pulsar Eliminar Todos. Una vez seleccionados los elementos regresaremos a la pantalla principal con el botón Volver. Hacemos lo mismo para seleccionar los receptores y las fibras.



**Seleccionar fibras ópticas**

### CARACTERÍSTICAS DE LA FIBRA OPTICA

Modelo : FIBRA OPTICA

Tipo de Conector : SC

Longitud de Onda Mínima : 850 (nm)

Longitud de Onda Máxima : 850 (nm)

Atenuación : 3,5 (dB/Km)

Dispersión : 0,1 (ns/nm\*Km)

Precio/metro : 0 (€)

FIBRA OPTICA

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR FIBRAS OPTICAS

SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

VOLVER

**FIGURA 6.7.** Seleccionamos los receptores que se utilizarán en el diseño.

**Seleccionar Receptores**

### CARACTERÍSTICAS DEL RECEPTOR

Modelo : PIN

Tipo de conector : SC

Longitud de Onda Mínima : 850 (nm)

Longitud de Onda Máxima : 850 (nm)

Velocidad Máxima : 40 (Mbps)

Sensibilidad : -42 (dBm)

Tiempo de Respuesta : 4 (ns)

Precio : 6,01 (€)

Precio Circuitería Ampf. : 12,02 (€)

PIN

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR RECEPTOR

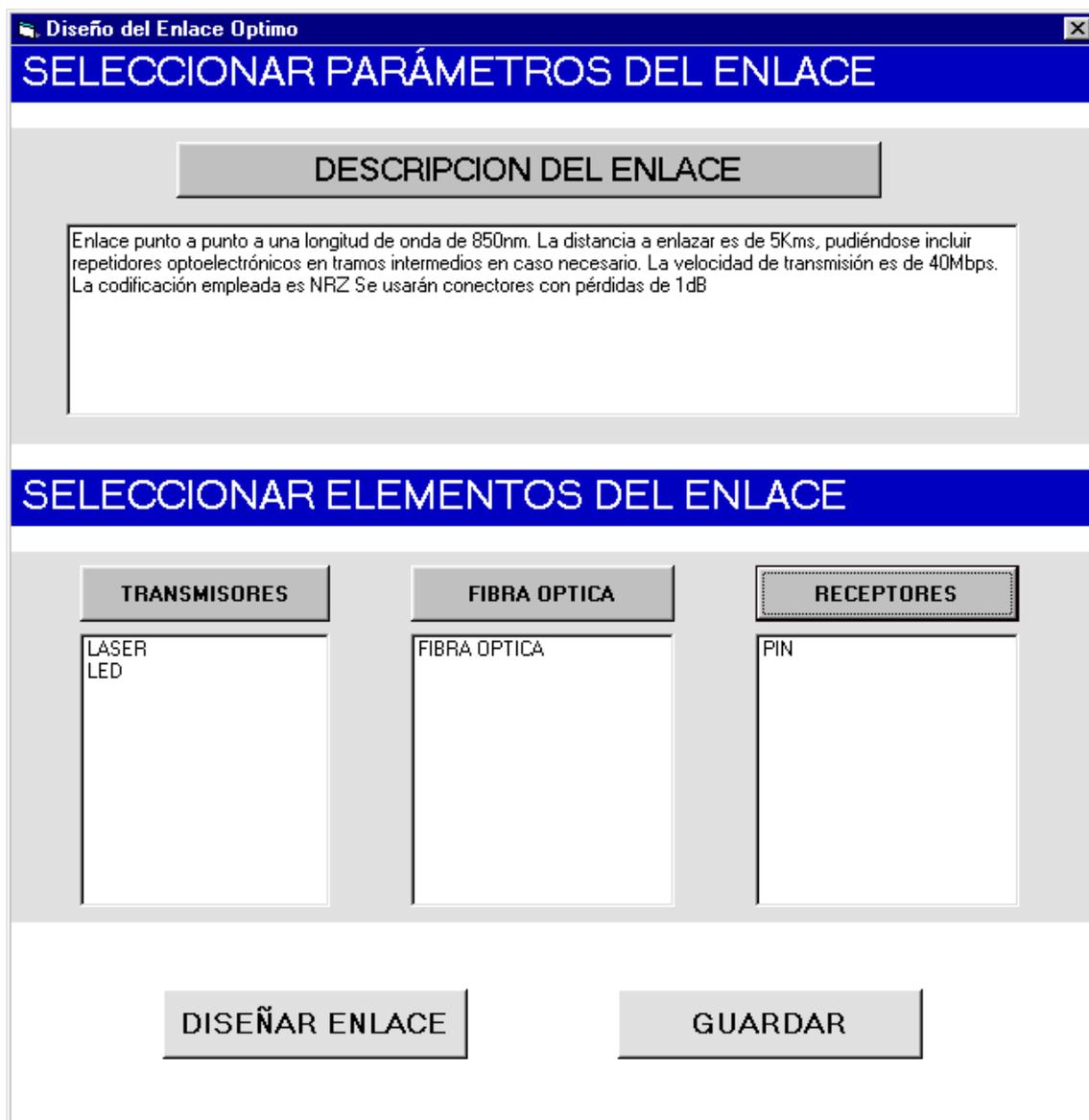
SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

VOLVER

**FIGURA 6.8.** Seleccionamos las fibras que se utilizarán en el diseño.



Finalmente regresaremos a la pantalla principal donde podremos ver las características que he definido para el enlace y los conectores, así como los elementos que he seleccionado y que formarán parte en el diseño del enlace. Para proceder con el mismo pulsaremos el botón Diseñar Enlace.



**FIGURA 6.9.** Diseño del Enlace.

El programa calculará para cada combinación de los posibles elementos que constituyen el enlace la solución óptima ( desde el punto de vista económico ), introduciendo para ello en el caso de ser necesario repetidores optoelectrónicos, simulando tramos intermedios.



Para ver los resultados correspondientes a una determinada combinación de los posibles elementos que pueden constituir el enlace, basta con seleccionarla en el árbol que aparece en la parte superior del interfaz gráfico de resultados.

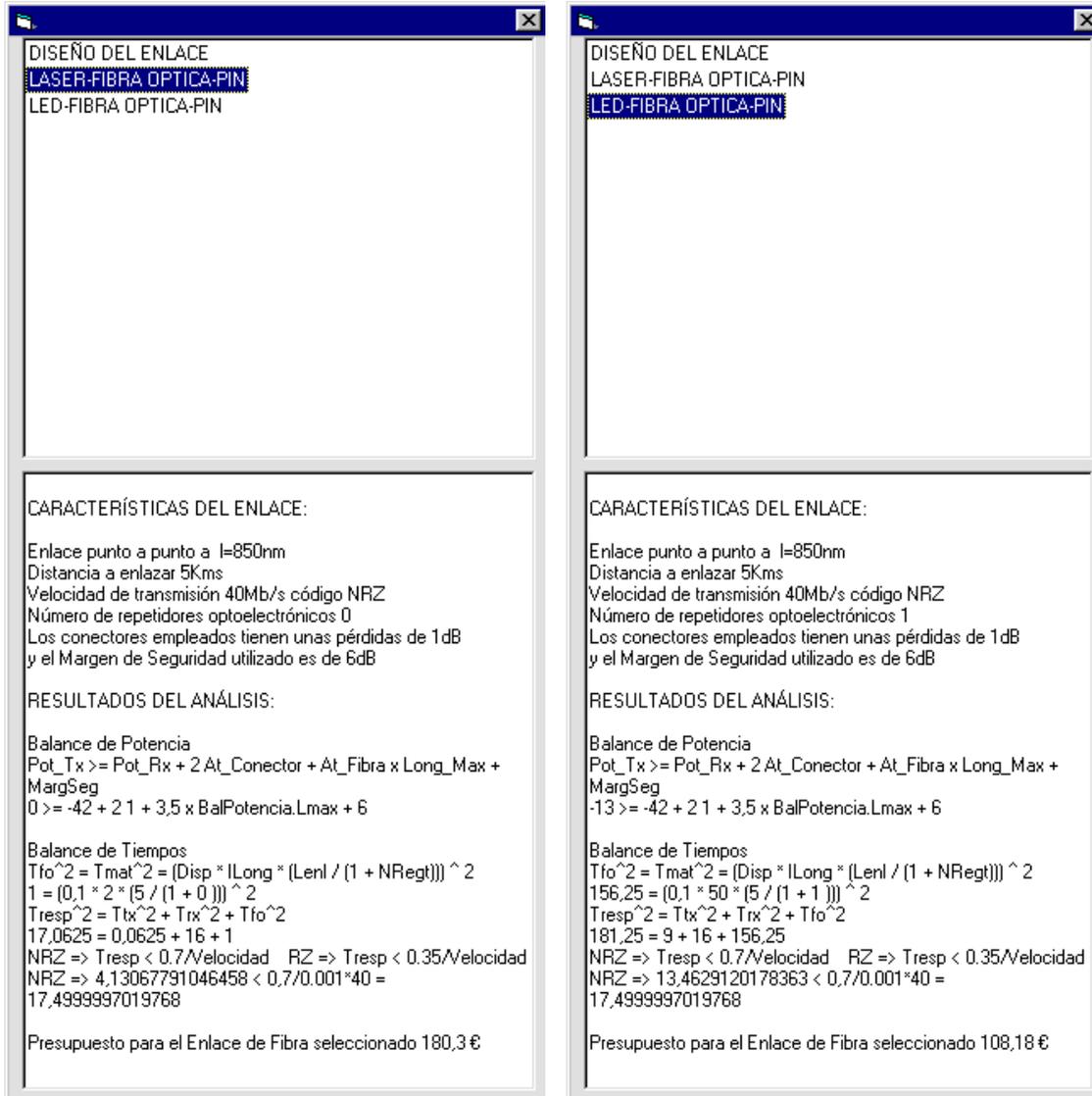
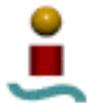


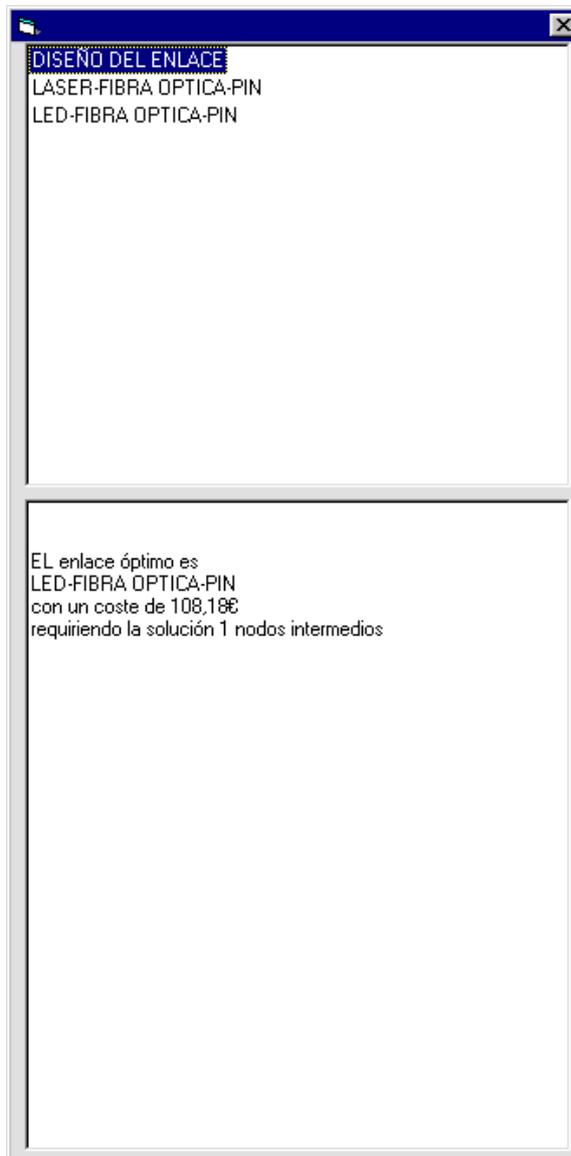
FIGURA 6.10. Resultados obtenidos para el diseño del enlace punto a punto.

En el cuadro inferior podemos ver tanto los datos que hemos utilizado para caracterizar el enlace, como las distintas ecuaciones empleadas para el diseño del mismo, así como si se ha precisado o no de la utilización de regeneradores optoelectrónicos para los tramos intermedios.

Al final del cuadro podremos ver la valoración económica implícita en el diseño del enlace.



Si pulsamos sobre Diseño del Enlace en el cuadro superior podremos ver la mejor opción para el diseño del enlace.



**FIGURA 6.11.** Solución óptima.



## SOLUCIÓN 2 OBTENIDA UTILIZANDO EL PROGRAMA: ELECCIÓN DEL COMPONENTE OPTIMO

Suponemos que los elementos que formarán parte en el diseño de la solución óptima ya se encuentran creados en sus correspondientes Bases de Datos. Abrimos la opción Componente Optimo y nos aparecerá una pantalla como la que se muestra a continuación.

**ELECCION DEL COMPONENTE OPTIMO**

**CARACTERISTICAS DEL ENLACE**

Longitud de Onda : 850 ( nm ) Velocidad : 40 ( Mbps )  
Distancia : 5 ( Km ) Margen de Seguridad 6 ( dB )  
Número de Regeneradores 0

**CARACTERISTICAS DEL CONECTOR**

Tipo : SC Perdidas : 1 ( dB ) Precio 6.01 ( € )

**NRZ**

NRZ  RZ

**SELECCIONA LOS ELEMENTOS CONOCIDOS :**

TRANSMISOR  RECEPTOR  FIBRA OPTICA

[Empty field] PIN FIBRA OPTICA

Transmisor Receptor Fibra Optica

**SELECCIONA LOS CRITERIOS DE EVALUACION :**

**Criterios**

COSTE DEL ENLACE  BALANCE DE POTENCIA  BALANCE DE TIEMPO

**CALCULAR** **GUARDAR**

**FIGURA 6.12.** Tendremos que rellenar los distintos campos que aparecen en la pantalla de Elección del Componente Optimo.

A continuación tendremos que introducir los datos que caracterizarán al enlace, los conectores, así como el código utilizado.



La siguiente opción será la de especificar los elementos que constituirán el enlace en el caso de que estos sean conocidos. Esta opción me permitirá seleccionar el mejor elemento para el diseño de mi enlace entre todos aquellos que tenga almacenados en la Base de Datos. Para escoger esta opción no se deberá marcar el cuadro de selección que aparece junto a los elementos ( para nuestro ejemplo no marcamos Transmisor, pues tengo dos opciones entre las que poder elegir ). Puesto que conocemos el receptor y la fibra óptica que vamos a utilizar en el diseño del enlace, marcamos el elemento correspondiente y los seleccionamos con los botones de desplazamiento entre los distintos elementos de la Base de Datos.

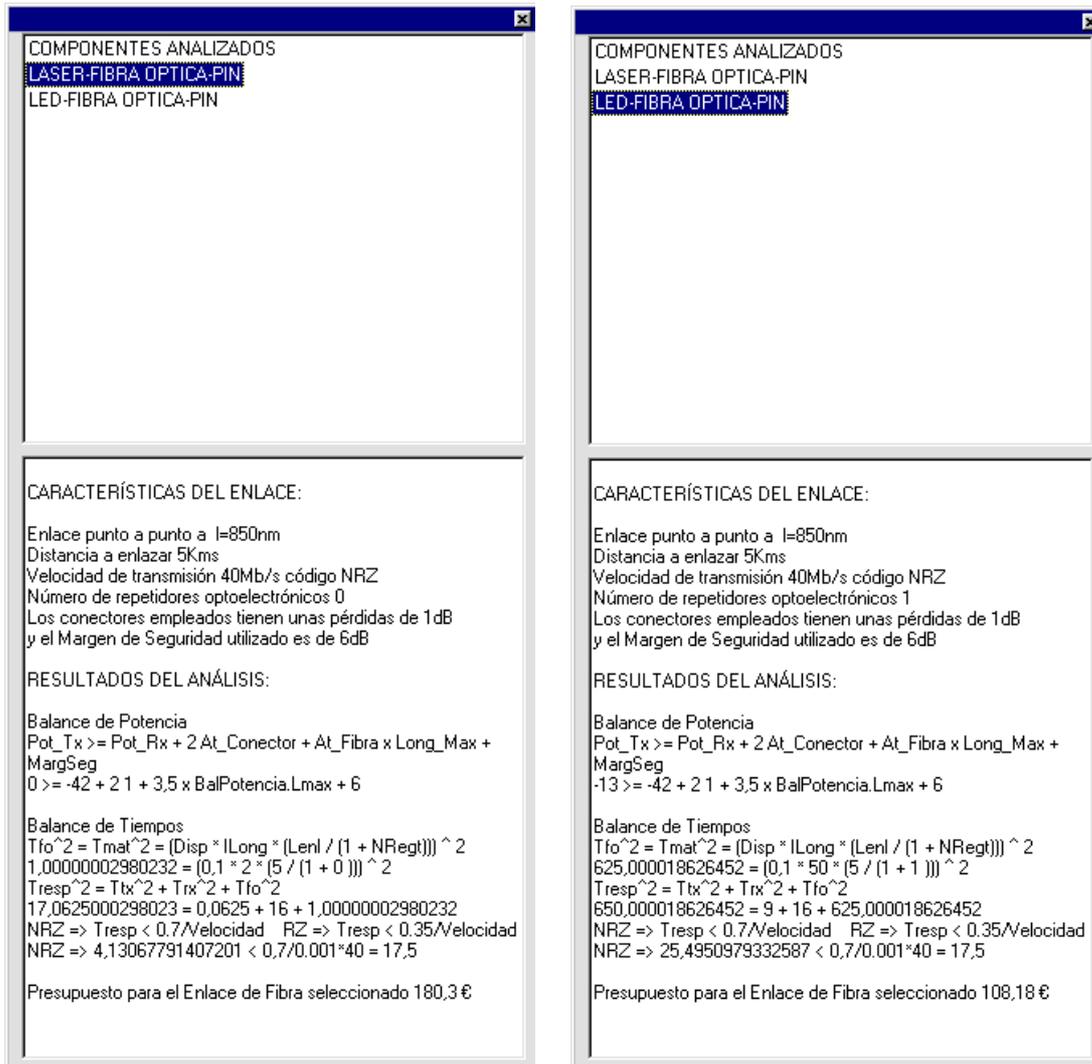
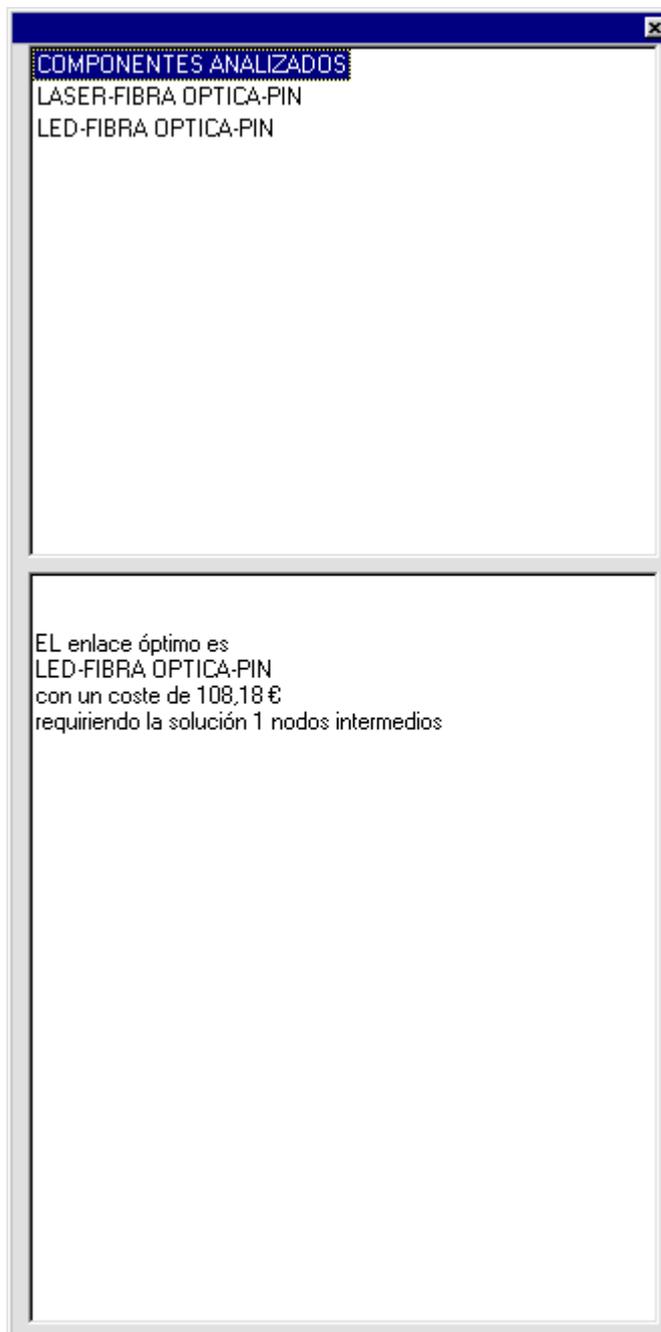


FIGURA 6.13. Resultados obtenidos para el diseño del enlace punto a punto.

Por último tendré que seleccionar el criterio utilizado para la elección del componente óptimo, que podrá ser el mejor desde el punto de vista económico para el diseño del enlace, o simplemente el mejor componente desde el punto de vista del balance de potencias o del balance de tiempos.



Para obtener los resultados deberemos pulsar el botón Calcular obteniéndose los resultados que se reflejan en la FIGURA 6.13. Si lo que queremos es ver directamente el resultado óptimo tendremos que pulsar la opción Componentes Analizados, que aparece en el cuadro superior.



**FIGURA 6.14.** Resultado óptimo.

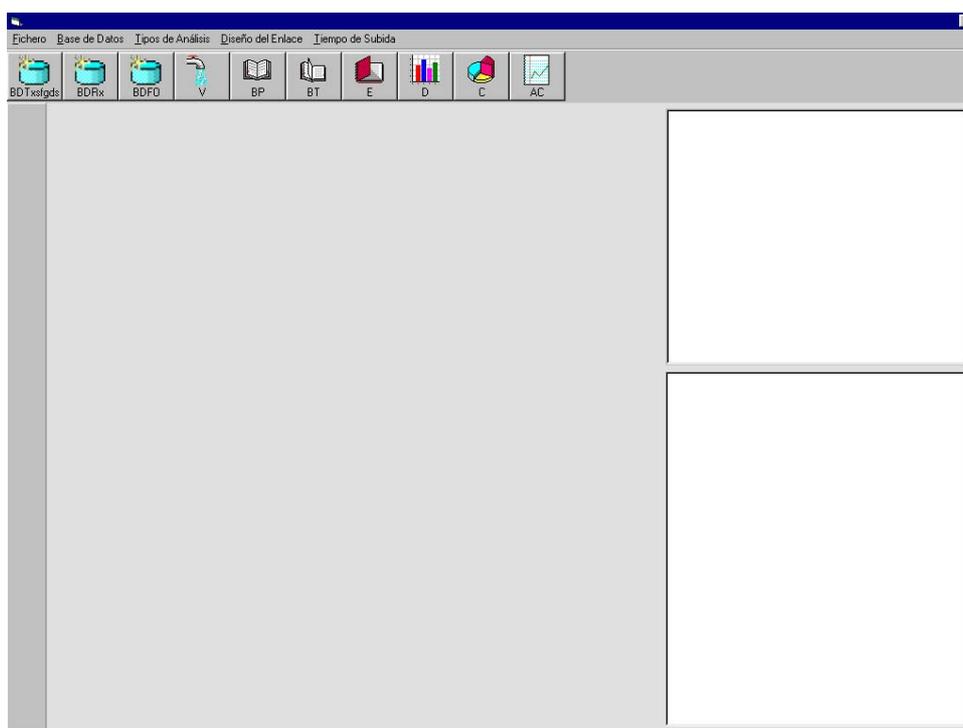
Si deseamos salvar los resultados obtenidos en un fichero lo podremos hacer utilizando el botón Guardar.



## MANUAL DE USUARIO

### PANTALLA PRINCIPAL: SELECCIÓN DE MENÚ

La figura muestra la pantalla principal que nos aparecerá cuando ejecutemos el programa. Desde ella podremos acceder a las distintas funciones que desarrolla la aplicación, bien desde la serie de menús desplegables que nos aparecen en la parte superior de la pantalla, bien utilizando los botones de pulsación que aparecen justo debajo ( si situamos el ratón sobre ellos nos aparece una breve descripción de su funcionalidad ).



Para cerrar una aplicación que se esté ejecutando pulsaremos el botón cerrar que aparece en la esquina superior derecha, una vez hayamos concluido.

La parte derecha de la pantalla muestra el interfaz gráfico de resultados cuyo manejo se explicará más adelante.



## BASES DE DATOS

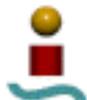
Se han implementado tres bases de datos donde almacenar los distintos elementos que constituyen el enlace: transmisor – fibra óptica – receptor. Estas bases de datos serán accedidas cada vez que se requiera la utilización de cualquiera de sus componentes.

The screenshot shows a software window titled 'Base de Datos Emisores' with a sub-header 'CARACTERÍSTICAS DEL EMISOR OPTICO'. The form contains the following fields and controls:

Modelo :	LASER		Nuevo
Tipo de conector :	SC		
Longitud de Onda Mínima :	850	( nm )	Editar
Longitud de Onda Máxima :	850	( nm )	
Velocidad Máxima :	40	( Mbps )	Grabar
Potencia :	0	( dBm )	Borrar
Tiempo de Respuesta :	0,25	( ns )	
Incremento de Longitud de Onda :	2	( nm )	Cancelar
Precio :	120,2	( € )	
Precio Circuitería Adicional :	30,05	( € )	Refrescar

At the bottom of the form, there are five navigation buttons: Inicio, Anterior, Siguiete, Final, and Buscar.

De forma que, para introducir un nuevo elemento en el diseño del enlace, se tendrá que hacer generándolo previamente en la base de datos correspondiente, introduciendo los datos de los parámetros que lo caracterizan. El valor de dichos datos deberá corresponderse con las unidades que aparecen a la derecha de las cajas habilitadas para la escritura de los mismos.



A continuación se enumeran las distintas acciones que se pueden realizar sobre las bases de datos.

### Creación de un nuevo elemento

Permite generar un nuevo elemento en la base de datos, para lo cual se tendrán que introducir los valores de los parámetros que lo caracterizan, en las unidades especificadas. Para ello se tendrá que pulsar el botón NUEVO, permitiendo introducir a partir de ese momento los datos para el nuevo elemento creado, y pulsar a continuación el botón GRABAR para salvar los cambios introducidos en la base de datos.

Base de Datos Receptores	
CARACTERISTICAS DEL RECEPTOR OPTICO	
Modelo :	PIN
Tipo de conector :	SC
Longitud de Onda Mínima :	850 (nm)
Longitud de Onda Máxima :	850 (nm)
Velocidad Máxima :	40 (Mbps)
Sensibilidad :	-42 (dBm)
Tiempo de Respuesta :	4 (ns)
Precio :	6,01 (€)
Precio Circuitería Amplificadora :	12,02 (€)

Buttons: Nuevo, Editar, Grabar, Borrar, Cancelar, Refrescar

Navigation: Inicio, Anterior, Siguiente, Final, Buscar

### Edición de un elemento existente

Permite modificar los valores asignados a los parámetros que caracterizan a un determinado elemento. Para ello tendremos que seleccionar el elemento en cuestión y pulsar el botón EDITAR. A continuación se podrá modificar el valor del parámetro que deseemos cambiar, y una vez modificado, salvar los cambios pulsando el botón GRABAR. Antes de pulsar el botón EDITAR, si intentamos cambiar el valor de cualquiera de los campos, comprobaremos que no es posible. Esta implantación proporciona a las bases de datos una seguridad contra posibles modificaciones que se pudieran producir por error.



## Deshacer los cambios realizados

En el caso de querer dar marcha atrás con los cambios realizados, se deberá pulsar el botón CANCELAR, lo que nos retornará a la situación inicial.

## Eliminar un elemento de la base de datos

Para eliminar un elemento de la base de datos se deberá seleccionar, posicionándose sobre él, y pulsar el botón BORRAR. Antes de eliminar de forma permanente el elemento de la base de datos se nos pedirá conformidad para llevar a cabo dicha acción.

Base de Datos Fibras Ópticas

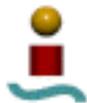
### CARACTERÍSTICAS DE LA FIBRA OPTICA

Modelo :	FIBRA	Nuevo
Tipo de Conector :	SC	Editar
Longitud de Onda Mínima :	850 (nm)	Grabar
Longitud de Onda Máxima :	850 (nm)	Borrar
Atenuación :	3,5 (dB / Km)	Cancelar
Dispersión :	0,1 (ns / nm * Km )	Refrescar
Precio / metro :	0 (€)	

Inicio Anterior Siguiete Final Buscar

## Ordenar los elementos almacenados en la base de datos

Si se han introducido nuevos elementos en la base de datos, estos se añadirán al final de la misma, y no mantendrán el orden alfabético mediante el campo MODELO. Para que ocupen su lugar correspondiente en la base de datos se deberá pulsar el botón REFRESCAR, quedando de este modo ordenados alfabéticamente. Esta operación se deberá realizar de igual modo cuando se modifique el campo MODELO de cualquiera de los elementos existentes. En el caso de no refrescar tras las modificaciones introducidas, los elementos aparecerán ordenados ocupando su posición correspondiente la próxima vez que se vuelva a abrir la base de datos.



## **Seleccionar un elemento de la base de datos**

Para acceder a un elemento determinado de la base de datos se podrá hacer de dos formas distintas:

### Acceso Secuencial

Se irán recorriendo los elementos de la base de datos uno a uno, utilizando para ello los botones ANTERIOR y SIGUIENTE. Con el botón INICIO nos posicionaremos en el primer elemento y con el botón FINAL en el último de la base de datos. Recordar que los elementos se encuentran ordenados alfabéticamente por el campo MODELO.

### Acceso Aleatorio

Podemos posicionarnos directamente sobre el elemento deseado, pulsando el botón BUSCAR e introduciendo el valor del campo MODELO en el cuadro de diálogo que nos aparece. No será necesario introducir el nombre completo en el campo buscar, aproximando la búsqueda al elemento en cuyo campo MODELO se dé el mayor número de coincidencias encontradas.



## COMPATIBILIDAD

Con esta opción se podrá determinar la compatibilidad de los elementos seleccionados para constituir el enlace de fibra ( transmisor – fibra óptica – receptor ) desde el punto de vista de la longitud de onda del enlace y del tipo de conectores disponibles para la fibra.

CARACTERISTICAS DEL ENLACE DE FIBRA OPTICA

Longitud de Onda del Enlace de Fibra 850 ( nm )

Tipo de conector disponible : SQ

TRANSMISOR RECEPTOR FIBRA OPTICA

Lista de transmisores Lista de receptores Lista de fibras ópticas

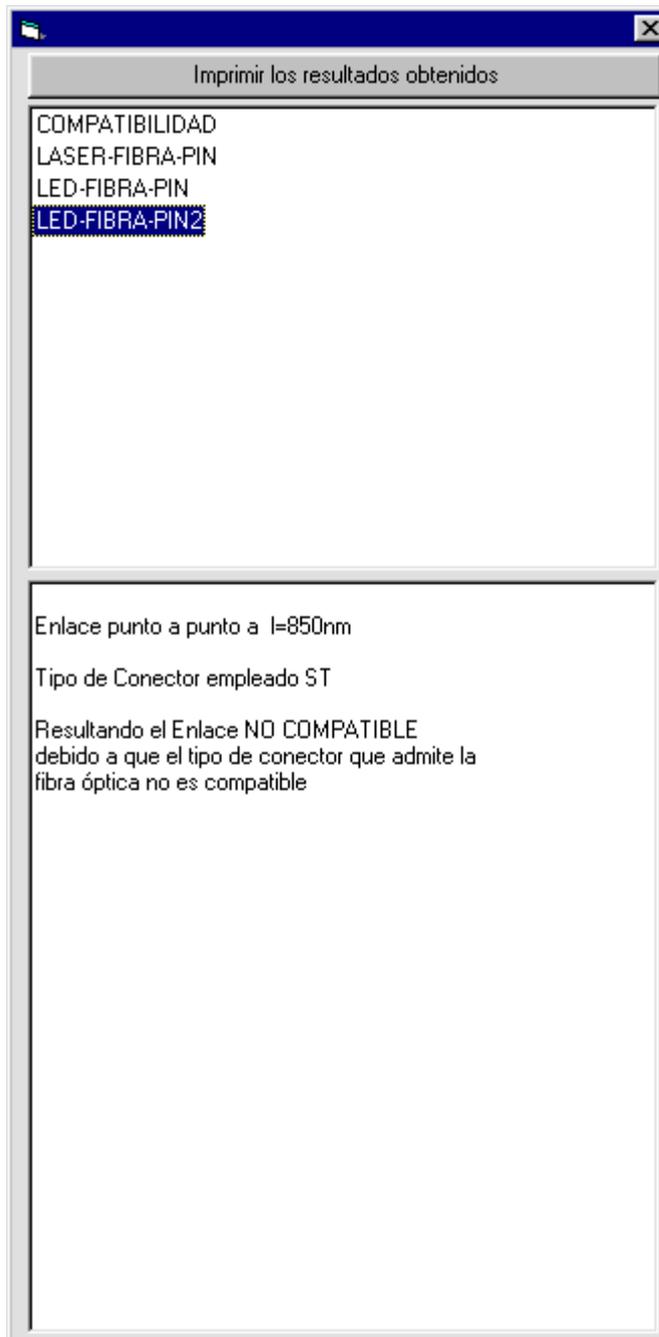
COMPATIBILIDAD GUARDAR

La forma de proceder es muy sencilla; se deberá introducir el valor para la longitud de onda del enlace, expresando éste en nanómetros, y especificar el tipo de conectores disponibles, en las casillas habilitadas para ello. A continuación se deberán escoger los elementos que van a constituir el enlace, para lo cual utilizaremos las listas desplegadas que aparecen en pantalla.

Finalmente, pulsando el botón COMPATIBILIDAD se genera la solución. Esta acción generará el interfaz de resultados, que se encuentra dividido en dos zonas distintas. En la parte superior aparecen los elementos que van a formar parte del enlace, y en la parte inferior aparecen los resultados de la función compatibilidad asociados a dichos elementos. Para visualizarlos se deberá pinchar sobre los distintos elementos del cuadro superior.



Esta función se podrá ejecutar las veces que sea necesario con tan solo modificar los datos requeridos y volver a presionar el botón COMPATIBILIDAD.



Si se desea se puede generar un fichero donde almacenar los datos obtenidos pulsando el botón GUARDAR, generándose un fichero con el nombre que se le haya dado y extensión .com o imprimir los resultados presionando el botón IMPRIMIR.



## BALANCE DE POTENCIA

Esta opción nos permitirá examinar si los elementos seleccionados para el diseño del enlace verifican el balance de potencias. Para ello se tendrán que introducir los valores que caracterizan el enlace de fibra desde el punto de vista del balance de potencia, en las casillas habilitadas para ello:

- Longitud de Onda del enlace, expresada en nanómetros
- Distancia a enlazar, expresada en kilómetros
- Existencia de repetidores optoelectrónicos en tramos intermedios del enlace
- Margen de Seguridad, expresado en decibelios

**BALANCE DE POTENCIA**

### CARACTERISTICAS DEL ENLACE DE FIBRA

Longitud de Onda : 850 ( nm )      Número de Regeneradores : 6

Distancia : 5 ( Km )      Margen de Seguridad: 0 ( dB )

### CARACTERISTICAS DEL CONECTOR

Tipo de conector : SC      Perdidas : 1 ( dB )

**TRANSMISOR      RECEPTOR      FIBRA OPTICA**

Lista de transmisores ▼      Lista de receptores ▼      Lista de fibras ópticas ▼

**CALCULAR      GUARDAR**

De igual modo se deberán introducir los valores que caracterizan a los conectores disponibles:

- Tipo de conector
- Pérdidas introducidas por los conectores, expresado en decibelios



Una vez introducidos estos valores, se seleccionarán los elementos que formarán parte del enlace, y se presionará el botón CALCULAR para obtener los resultados. Pinchando sobre los elementos que aparecen en el cuadro superior del interfaz de resultados, nos aparecerá, en el cuadro inferior, una descripción del enlace junto con los resultados del análisis. De igual modo aparecen las ecuaciones utilizadas para llegar al resultado obtenido, permitiéndonos detectar posibles errores si desarrollamos los cálculos a mano.

En el caso de verificarse el balance de potencia se nos proporcionará información acerca de cuánto podemos aumentar la distancia del enlace, de forma que se siga verificando el balance. En caso contrario se nos informará de en cuántos kilómetros se deberá reducir la distancia del enlace para que el balance de potencia se verifique.



## BALANCE DE TIEMPOS

Esta opción nos permitirá examinar si los elementos seleccionados para el diseño del enlace verifican el balance de tiempos. Para ello se tendrán que introducir los valores que caracterizan el enlace de fibra desde el punto de vista del balance de tiempos, en las casillas habilitadas para ello:

- Longitud de Onda del enlace, expresada en nanómetros
- Distancia a enlazar, expresada en kilómetros
- Velocidad máxima en el enlace, expresada en baudios ( Mb/s )
- Existencia de repetidores optoelectrónicos en tramos intermedios del enlace

**BALANCE DE TIEMPO**

### CARACTERISTICAS DEL ENLACE DE FIBRA

Longitud de Onda : 850 ( nm )      Distancia : 5 ( Kms )

Velocidad : 40 ( Mbps )      Número de Regeneradores : 0

Tipo de conector disponible : SC      Código  NRZ  RZ

**TRANSMISOR**      **RECEPTOR**      **FIBRA OPTICA**

Lista de transmisores ▼      Lista de receptores ▼      Lista de fibras ópticas ▼

**CALCULAR**      **GUARDAR**

De igual modo se deberán introducir el tipo de conectores disponibles y el método de modulación óptica utilizado ( código ).

Una vez introducidos estos valores, se seleccionarán los elementos que formarán parte del enlace, y se presionará el botón CALCULAR para obtener los resultados.



Pinchando sobre los elementos que aparecen en el cuadro superior del interfaz de resultados, nos aparecerá, en el cuadro inferior, una descripción del enlace junto con los resultados del análisis. De igual modo aparecen las ecuaciones utilizadas para llegar al resultado obtenido, permitiéndonos detectar posibles errores si desarrollamos los cálculos a mano.

En el caso de verificarse el balance de tiempos se nos proporcionará información acerca de cuánto podemos aumentar la velocidad del enlace, de forma que se siga verificando el balance. En caso contrario se nos informará de en cuánto se deberá reducir la velocidad del enlace para que el balance de tiempos se verifique.



## ANÁLISIS DEL ENLACE DE FIBRA

Esta opción deberá utilizarse cuando los elementos que constituyen el enlace son conocidos, así como las características principales del mismo. Para un transmisor, fibra óptica y receptor seleccionados, verifica si el enlace es implementable ( no existen incompatibilidades entre los elementos seleccionados, y se verifica el balance de potencia y de tiempos ) y calcula el coste asociado al diseño del mismo. En el caso de que el enlace no sea realizable se mostrará un mensaje en el cual se indicará la causa de porqué no es implementable.

**CARACTERÍSTICAS DEL ENLACE DE FIBRA OPTICA**

**CARACTERÍSTICAS DEL ENLACE**

Longitud de Onda : 850 ( nm    Velocidad : 40 ( Mbps )

Distancia : 5 ( Km )    Margen de Seguridad: 6 ( dB )

Número de Regeneradores : 0

**CARACTERÍSTICAS DEL CONECTOR**

Tipo : SC    Perdidas : 1 ( dB )    Precio : 6.01 ( € )

Código     NRZ     RZ

**TRANSMISOR    RECEPTOR    FIBRA OPTICA**

Lista de transmisores    Lista de receptores    Lista de fibras ópticas

**CALCULAR    GUARDAR**

Para ejecutarlo bastará con rellenar los campos solicitados y pulsar CALCULAR.



## DISEÑO DEL ENLACE DE FIBRA

Esta opción deberá utilizarse cuando se disponga de un conjunto de transmisores, receptores y fibras ópticas, y se desee calcular que combinación de los elementos anteriores constituyen el enlace óptimo, desde el punto de vista de los costes asociados al diseño del enlace.

Para cualquier combinación de los elementos seleccionados, el programa calculará la opción, que siendo implementable, resulte económicamente más favorable, aunque para ello se tengan que introducir repetidores optoelectrónicos en los tramos intermedios del enlace.

Diseño del Enlace Optimo

### SELECCIONAR PARÁMETROS DEL ENLACE

**DESCRIPCION DEL ENLACE**

Enlace punto a punto a una longitud de onda de 850nm. La distancia a enlazar es de 5Kms, pudiéndose incluir repetidores optoelectrónicos en tramos intermedios en caso necesario. La velocidad de transmisión es de 40Mbps. La codificación empleada es RZ Se usarán conectores con pérdidas de 1dB

### SELECCIONAR ELEMENTOS DEL ENLACE

TRANSMISORES	FIBRA OPTICA	RECEPTORES
LASER LED	FIBRA	PIN

**DISEÑAR ENLACE**      **GUARDAR**



Deberemos ir seleccionando cada una de los requisitos que se muestran en la pantalla principal pulsando el botón correspondiente y rellenar los datos solicitados. Una vez hecho esto pulsaremos el botón VOLVER y regresaremos al menú principal, quedando registrado en él los datos introducidos para el diseño del enlace.

**CARACTERISTICAS DEL ENLACE DE FIBRA OPTICA**

**CARACTERISTICAS DEL ENLACE**

Longitud de Onda: 850 (nm) Velocidad: 40 (Mbps)

Distancia: 5 (Km) Margen de Seguridad: 6 (dB)

**Código**  NRZ  RZ

**CARACTERISTICAS DEL CONECTOR**

Tipo de conector disponible: SC

Perdidas: 1 (dB) Precio: 6,01 (€)

VOLVER

Una vez introducidos todos los datos requeridos pulsaremos el botón CALCULAR para obtener los resultados. Para ver el resultado óptimo deberemos pulsar en el nodo Diseño del Enlace del árbol de resultados.



Para seleccionar qué emisores, receptores y fibras ópticas queremos para el diseño del enlace nos aparecen unas pantallas como las que se muestran a continuación donde podremos ir seleccionando elemento a elemento de las bases de datos con el botón SELECCIONAR o por el contrario seleccionar todos los elementos almacenados en la base de datos utilizando el botón SELECCIONAR TODOS.

**Seleccionar Emisor**

### CARACTERISTICAS DEL EMISOR

Modelo :

Tipo de conector :

Longitud de Onda Mínima :  (nm)

Longitud de Onda Máxima :  (nm)

Velocidad Máxima :  (Mbps)

Potencia :  (dBm)

Tiempo de Respuesta :  (ns)

Incremento Longitud Onda :  (nm)

Precio :  (€)

Precio Circuitería Adicional :  (€)

**LASER**

LED

Inicio

Anterior

Siguiente

Final

Buscar

### SELECCIONAR EMISOR

SELECCIONAR SELECCIONAR TODOS ELIMINAR TODOS ELIMINAR

VOLVER

Si queremos eliminar alguno de los elementos seleccionados bastará con marcarlo en la lista y pulsar ELIMINAR. Si deseamos limpiar la lista completa y hacer la selección nuevamente usaremos el botón ELIMINAR TODOS. Una vez concluida la selección pulsaremos el botón VOLVER registrando de este modo las modificaciones.



## SELECCIÓN DEL COMPONENTE ÓPTIMO

Esta opción deberá seleccionarse en aquellos casos en los que se tiene perfectamente clara la composición del enlace, salvo alguno de los elementos constitutivos del mismo. El programa determinará cuál es la mejor opción de entre todas las que tiene almacenadas en las bases de datos, en base al criterio especificado, que podrá ser un criterio del tipo económico, coste del enlace, o un criterio referente al diseño del enlace, balance de potencia o balance de tiempos.

ELECCION DEL COMPONENTE OPTIMO							
CARACTERISTICAS DEL ENLACE							
Longitud de Onda :	850	( nm )	Velocidad :	40	( Mbps )		
Distancia :	5	( Km )	Margen de Seguridad :	6	( dB )		
Número de Regeneradores :	0						
CARACTERISTICAS DEL CONECTOR							
Tipo :	SC	Perdidas :	1	( dB )	Precio :	6,01	( € )
Código		<input checked="" type="radio"/> NRZ <input type="radio"/> RZ					
SELECCIONA LOS ELEMENTOS CONOCIDOS :							
<input type="checkbox"/> TRANSMISOR	<input checked="" type="checkbox"/> RECEPTOR	<input checked="" type="checkbox"/> FIBRA OPTICA					
LASER	PIN	FIBRA					
Transmisor	Receptor	Fibra Optica					
SELECCIONA LOS CRITERIOS DE EVALUACION :							
Criterios							
<input checked="" type="radio"/> COSTE DEL ENLACE	<input type="radio"/> BALANCE DE POTENCIA	<input type="radio"/> BALANCE DE TIEMPO					
CALCULAR		GUARDAR					



## CÁLCULO DEL TIEMPO DE SUBIDA

Esta opción nos permitirá calcular el tiempo de subida asociado a una fibra óptica, en el caso de no venir especificado en las características del fabricante.

Para calcular el tiempo de subida introduciremos en primer lugar la longitud de onda del enlace y pulsaremos el botón creado para tal efecto. Esto habilitará unas casillas u otras en el formulario correspondientes a los parámetros de la fibra óptica, ya que para el cálculo del tiempo de subida se deberá especificar si la fibra es multimodo o monomodo.

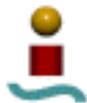
CALCULO DEL TIEMPO DE SUBIDA DE LA FIBRA		
Longitud de Onda de trabajo:	850	( nm )
Modelo	FibraPrueba	
Dispersión cromática :	0.1	( ns/nm x Km )
Dispersión cromática material :		( ns/nm x Km )
Dispersión cromática guía-onda :		( ns/nm x Km )
Anchura espectral del generador de luz :	850	( nm )
Ancho de banda modal de la fibra óptica :	40	( MHz x Km )
<b>CALCULAR</b> <b>GUARDAR</b>		

Seguidamente se rellenarán los campos del formulario y pulsaremos el botón CALCULAR. Los campos que deberán rellenarse son:

Fibras Multimodo: Dispersión cromática

Fibras Monomodo: Dispersión cromática material y dispersión cromática guía-onda

En los resultados obtenidos aparecen las ecuaciones utilizadas para el cálculo según el caso en el que nos encontremos.



## CÓDIGO FUENTE

### BDEmisores.frm

Option Explicit

```
Private Sub InhabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

```
Private Sub InhabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

```
Private Sub FinBuscar_Click()  
    Adodc1.Recordset.Close  
    Adodc1.Refresh  
End Sub
```



```
Private Sub Form_Load()  
    Grabar.Enabled = False  
    InhabilitarCajas  
    ChDir App.path  
End Sub
```

```
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub
```

```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```

```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

```
Private Sub Nuevo_Click()  
    HabilitarCajas  
    InhabilitarBotones  
    Grabar.Enabled = True  
    Cancelar.Enabled = True  
    Adodc1.Recordset.AddNew 'añadir un nuevo registro  
    Modelo.SetFocus      'poner el cursor en la caja del "Modelo"  
End Sub
```

```
Private Sub Editar_Click()  
    HabilitarCajas  
    InhabilitarBotones  
    Grabar.Enabled = True  
    Cancelar.Enabled = True  
    Modelo.SetFocus      'poner el cursor en la caja del "Modelo"  
End Sub
```

```
Private Sub Grabar_Click()  
    Adodc1.Recordset.Update  
    HabilitarBotones  
    Grabar.Enabled = False  
    InhabilitarCajas  
End Sub
```



```
Private Sub Cancelar_Click()
    Adodc1.Recordset.CancelUpdate
    HabilitarBotones
    Grabar.Enabled = False
    InhabilitarCajas
End Sub

Private Sub Borrar_Click()
    Dim R As Integer
    On Error GoTo RutinaDeError
    R = MsgBox("¿Desea borrar el registro?", vbYesNo, "Atención")
    If R <> vbYes Then Exit Sub
    Adodc1.Recordset.Delete 'borrar el registro actual
    Adodc1.Recordset.MoveNext 'situarse en el registro siguiente
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
    Exit Sub
RutinaDeError:
    R = MsgBox(Error, vbOKOnly, "Se ha producido un error:")
    Adodc1.Recordset.CancelUpdate
End Sub

Private Sub Refrescar_Click()
    Adodc1.Recordset.Requery
    HabilitarBotones
    Grabar.Enabled = False
End Sub

Private Sub Buscar_Click()
    FrmBuscar.Nombres = "Modelo;Longitud_Onda_Min;Longitud_Onda_Max;Anchura
Espectral;Frecuencia_Max;Potencia_Optica;Tiempo_Respuesta;Tipo_Conector;Precio;Precio_
CtoAdicional"
    FrmBuscar.Campos =
"Modelo;Longitud_Onda_Min;Longitud_Onda_Max;Incremento_LOnda;Velocidad_Max;Potenci
a;Tiempo_Respuesta;Tipo_Conector;Precio;Precio_CtoAdicional"
    FrmBuscar.Rs = Adodc1
    FrmBuscar.Show vbModal
    Set Adodc1.Recordset = FrmBuscar.Rs.Recordset
End Sub

Private Sub InclOnda_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(InclOnda) Then
        Cancel = True
        MsgBox "El incremento de la longitud de onda debe ser un valor numérico"
    End If
End Sub

Private Sub LOndaMax_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(LOndaMax) Then
        Cancel = True
        MsgBox "La longitud de onda máxima debe ser un valor numérico"
    End If
End Sub
```

---



```
Private Sub LOndaMin_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMin) Then
    Cancel = True
    MsgBox "La longitud de onda mínima debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Potencia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Potencia) Then
    Cancel = True
    MsgBox "La potencia debe ser un valor numérico"
End If
End Sub
```

```
Private Sub PrecCirc_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(PrecCirc) Then
    Cancel = True
    MsgBox "El precio del circuito adicional debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TRespuesta_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(TRespuesta) Then
    Cancel = True
    MsgBox "El tiempo de respuesta debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```



## **BDRectores.frm**

Option Explicit

```
Private Sub InhabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

```
Private Sub InhabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

```
Private Sub FinBuscar_Click()  
    Adodc1.Recordset.Close  
    Adodc1.Refresh  
End Sub
```

```
Private Sub Form_Load()  
    Grabar.Enabled = False  
    InhabilitarCajas  
    ChDir App.path  
End Sub
```

```
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub
```



```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```

```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

```
Private Sub Nuevo_Click()  
    HabilitarCajas  
    InhabilitarBotones  
    Grabar.Enabled = True  
    Cancelar.Enabled = True  
    Adodc1.Recordset.AddNew 'añadir un nuevo registro  
    Modelo.SetFocus      'poner el cursor en la caja del "Modelo"  
End Sub
```

```
Private Sub Editar_Click()  
    HabilitarCajas  
    InhabilitarBotones  
    Grabar.Enabled = True  
    Cancelar.Enabled = True  
    Modelo.SetFocus      'poner el cursor en la caja del "Modelo"  
End Sub
```

```
Private Sub Grabar_Click()  
    Adodc1.Recordset.Update  
    HabilitarBotones  
    Grabar.Enabled = False  
    InhabilitarCajas  
End Sub
```

```
Private Sub Cancelar_Click()  
    Adodc1.Recordset.CancelUpdate  
    HabilitarBotones  
    Grabar.Enabled = False  
    InhabilitarCajas  
End Sub
```



```
Private Sub Borrar_Click()
    Dim R As Integer
    On Error GoTo RutinaDeError
    R = MsgBox("¿Desea borrar el registro?", vbYesNo, "Atención")
    If R <> vbYes Then Exit Sub
    Adodc1.Recordset.Delete 'borrar el registro actual
    Adodc1.Recordset.MoveNext 'situarse en el registro siguiente
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
    Exit Sub
RutinaDeError:
    R = MsgBox(Error, vbOKOnly, "Se ha producido un error:")
    Adodc1.Recordset.CancelUpdate
End Sub

Private Sub Refrescar_Click()
    Adodc1.Recordset.Requery
    HabilitarBotones
    Grabar.Enabled = False
End Sub

Private Sub Buscar_Click()
    FrmBuscar.Nombres =
"Modelo;Longitud_Onda_Min;Longitud_Onda_Max;Frecuencia_Max;Sensibilidad;Tiempo_Resp
uesta;Tipo_Conector;Precio;Precio_CtoAmplificador"
    FrmBuscar.Campos =
"Modelo;Longitud_Onda_Min;Longitud_Onda_Max;Velocidad_Max;Potencia;Tiempo_Respuest
a;Tipo_Conector;Precio;Precio_CtoAmplificador"
    FrmBuscar.Rs = Adodc1
    FrmBuscar.Show vbModal
    Set Adodc1.Recordset = FrmBuscar.Rs.Recordset
End Sub

Private Sub LOndaMax_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMax) Then
    Cancel = True
    MsgBox "La longitud de onda máxima debe ser un valor numérico"
End If
End Sub

Private Sub LOndaMin_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMin) Then
    Cancel = True
    MsgBox "La longitud de onda mínima debe ser un valor numérico"
End If
End Sub
```



```
Private Sub Potencia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Potencia) Then
    Cancel = True
    MsgBox "La potencia debe ser un valor numérico"
End If
End Sub
```

```
Private Sub PrecCirc_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(PrecCirc) Then
    Cancel = True
    MsgBox "El precio del circuito adicional debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TRespuesta_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(TRespuesta) Then
    Cancel = True
    MsgBox "El tiempo de respuesta debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```



## **BDFibras.frm**

Option Explicit

```
Private Sub InhabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarBotones()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is CommandButton Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

```
Private Sub InhabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub
```

```
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub
```

```
Private Sub FinBuscar_Click()  
    Adodc1.Recordset.Close  
    Adodc1.Refresh  
End Sub
```

```
Private Sub Form_Load()  
    Grabar.Enabled = False  
    InhabilitarCajas  
    ChDir App.path  
End Sub
```

```
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub
```



```
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub
```

```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

```
Private Sub Nuevo_Click()  
    HabilitarCajas  
    InhabilitarBotones  
    Grabar.Enabled = True  
    Cancelar.Enabled = True  
    Adodc1.Recordset.AddNew 'añadir un nuevo registro  
    Modelo.SetFocus      'poner el cursor en la caja del "Modelo"  
End Sub
```

```
Private Sub Editar_Click()  
    HabilitarCajas  
    InhabilitarBotones  
    Grabar.Enabled = True  
    Cancelar.Enabled = True  
    Modelo.SetFocus      'poner el cursor en la caja del "Modelo"  
End Sub
```

```
Private Sub Grabar_Click()  
    Adodc1.Recordset.Update  
    HabilitarBotones  
    Grabar.Enabled = False  
    InhabilitarCajas  
End Sub
```

```
Private Sub Cancelar_Click()  
    Adodc1.Recordset.CancelUpdate  
    HabilitarBotones  
    Grabar.Enabled = False  
    InhabilitarCajas  
End Sub
```



```
Private Sub Borrar_Click()
    Dim R As Integer
    On Error GoTo RutinaDeError
    R = MsgBox("¿Desea borrar el registro?", vbYesNo, "Atención")
    If R <> vbYes Then Exit Sub
    Adodc1.Recordset.Delete 'borrar el registro actual
    Adodc1.Recordset.MoveNext 'situarse en el registro siguiente
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
    Exit Sub
RutinaDeError:
    R = MsgBox(Error, vbOKOnly, "Se ha producido un error:")
    Adodc1.Recordset.CancelUpdate
End Sub

Private Sub Refrescar_Click()
    Adodc1.Recordset.Requery
    HabilitarBotones
    Grabar.Enabled = False
End Sub

Private Sub Buscar_Click()
    FrmBuscar.Nombres =
"Modelo;Longitud_Onda_Min;Longitud_Onda_Max;Atenuacion;Dispersion;Tipo_Conector;Precio/metro"
    FrmBuscar.Campos =
"Modelo;Longitud_Onda_Min;Longitud_Onda_Max;Atenuacion;Dispersion;Tipo_Conector;Precio/metro"
    FrmBuscar.Rs = Adodc1
    FrmBuscar.Show vbModal
    Set Adodc1.Recordset = FrmBuscar.Rs.Recordset
End Sub

Private Sub LOndaMax_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMax) Then
    Cancel = True
    MsgBox "La longitud de onda máxima debe ser un valor numérico"
End If
End Sub

Private Sub LOndaMin_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMin) Then
    Cancel = True
    MsgBox "La longitud de onda mínima debe ser un valor numérico"
End If
End Sub
```

---



```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
'Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
'If Not IsNumeric(Velocidad) Then
'Cancel = True
'MsgBox "La velocidad debe ser un valor numérico"
'End If
'End Sub
```

```
Private Sub Atenuacion_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Atenuacion) Then
    Cancel = True
    MsgBox "La atenuación debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Dispersion_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Dispersion) Then
    Cancel = True
    MsgBox "La dispersión debe ser un valor numérico"
End If
End Sub
```



## Compatibilidad.frm

Dim resulnodeV As Node

Dim m As Integer

```
Private Sub Combo1_GotFocus()
```

```
Data1.Recordset.MoveFirst
```

```
While Data1.Recordset.EOF = False
```

```
    Combo1.AddItem Data1.Recordset.Fields("Modelo")
```

```
Data1.Recordset.MoveNext
```

```
Wend
```

```
End Sub
```

```
Private Sub Combo2_GotFocus()
```

```
Data2.Recordset.MoveFirst
```

```
While Data2.Recordset.EOF = False
```

```
    Combo2.AddItem Data2.Recordset.Fields("Modelo")
```

```
Data2.Recordset.MoveNext
```

```
Wend
```

```
End Sub
```

```
Private Sub Combo3_GotFocus()
```

```
Data3.Recordset.MoveFirst
```

```
While Data3.Recordset.EOF = False
```

```
    Combo3.AddItem Data3.Recordset.Fields("Modelo")
```

```
Data3.Recordset.MoveNext
```

```
Wend
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Unload ResultadosV
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
m = 0
```

```
Set resulnodeV = ResultadosV.TreeView1.Nodes.Add()
```

```
    resulnodeV.Text = "COMPATIBILIDAD"
```

```
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\emisores.mdb", vbCritical: Exit Sub
```

```
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\receptores.mdb", vbCritical: Exit Sub
```

```
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\fibras.mdb", vbCritical: Exit Sub
```

```
Data1.DatabaseName = App.path & "\emisores.mdb"
```

```
Data2.DatabaseName = App.path & "\receptores.mdb"
```

```
Data3.DatabaseName = App.path & "\fibras.mdb"
```

```
Data1.Refresh
```

```
Data2.Refresh
```

```
Data3.Refresh
```

```
End Sub
```



```
Private Sub GuardarComo_Click()
Dim path As String
Dim puntero
GuardarVBL.TipoSeleccion = Index
GuardarVBL.Show vbModal
If Trim(GuardarVBL.Retorna) <> "" Then
    path = GuardarVBL.Retorna
End If
puntero = FreeFile
If path <> "" Then
Open path For Output As #puntero
Write #puntero, LOnda.Text, TipoConector.Text, Combo1.Text, Combo2.Text, Combo3.Text
End If
Close #puntero
End Sub
```

```
Private Sub TipoConector_Change()
    TipoConector = UCase(TipoConector)
    TipoConector.SelStart = Len(TipoConector)
End Sub
```

```
Private Sub Viabilidad_Click()
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominrx As Integer, lominfo As Integer
Dim LOenl As Integer
Dim vb As Integer, causa As String
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo2.Text)) Then
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo3.Text)) Then
'Viabilidad referente al tipo de conector
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
'm = m + 1
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
```



```
Select Case vb
Case 0
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "COMPATIBLE" & vbCrLf
Case 1
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "NO COMPATIBLE" & vbCrLf & "debido a que el tipo de conector que admite el
equipo" & vbCrLf _
  & "transmisor no es compatible" & vbCrLf
Case 2
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "NO COMPATIBLE" & vbCrLf & "debido a que el tipo de conector que admite el
equipo" & vbCrLf _
  & "receptor no es compatible" & vbCrLf
Case 3
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "NO COMPATIBLE" & vbCrLf & "debido a que el tipo de conector que admite la" &
vbCrLf _
  & "fibra óptica no es compatible" & vbCrLf
Case 4
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "NO COMPATIBLE" & vbCrLf & "debido a que la longitud de onda del equipo" &
vbCrLf _
  & "transmisor no es compatible" & vbCrLf
Case 5
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "NO COMPATIBLE" & vbCrLf & "debido a que la longitud de onda del equipo" &
vbCrLf _
  & "receptor no es compatible" & vbCrLf
Case 6
  'Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
  'resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  causa = "NO COMPATIBLE" & vbCrLf & "debido a que la longitud de onda de la" & vbCrLf _
  & "fibra óptica no es compatible" & vbCrLf
End Select
'ResultadosV.Show
m = m + 1
resulV(m) = vbCrLf & _
  "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
  vbCrLf & _
  "Tipo de Conector empleado " & conectortipo & vbCrLf & _
  vbCrLf
```



```
resulVR(m) = "Resultando el Enlace " & causa
Set resulnodeV = ResultadosV.TreeView1.Nodes.Add(m)
resulnodeV.Text = Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo")
& "-" & Data2.Recordset.Fields("Modelo")
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
ResultadosV.Show
End Sub
```

```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
Cancel = True
MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```

## **GuardarVBL.frm**

Option Explicit

```
Dim Ret
Dim Tipo
```

```
Public Property Get Retorna() As Variant
Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
Ret = Retorna
End Property
```

```
Private Sub Command1_Click()
Dim extension As String
extension = ".vbl"
If Right(Dir1.path, 1) = "\" Then
Ret = Dir1.path & Text1.Text & extension
If Tipo = 1 Then Ret = Dir1.path
Else
Ret = Dir1.path & "\" & Text1.Text & extension
If Tipo = 1 Then Ret = Dir1.path
End If
Unload Me
End Sub
```



```
Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub

Private Sub File1_Click()
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Load()
    If App.PrevInstance Then Ret = "": Unload Me
    Ret = ""
    Dir1.path = Drive1.Drive
    File1.path = Dir1.path
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Retorna = Ret
End Sub

Public Property Get TipoSeleccion() As Variant
    TipoSeleccion = Tipo
End Property

Public Property Let TipoSeleccion(ByVal vNewValue As Variant)
    Tipo = vNewValue
End Property
```

### **ResultadosV.frm**

```
Private Sub TreeView1_NodeClick(ByVal Node As Node)
    Text1.Text = resulV(Node.Index - 1)
    Text2.Text = resulVR(Node.Index - 1)
End Sub
```



```
Private Sub Imprimir_Click()
    Dim Cabecera As String
    Dim Cuerpo As String
    Dim K As Integer
    Screen.MousePointer = vbHourglass
    Printer.Print vbCrLf & vbCrLf & vbCrLf
    Printer.Font.Size = 14
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf
    For K = 2 To TreeView1.Nodes.Count
        Printer.Font.Size = 10
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf
        Printer.Font.Size = 8
        Printer.Print resulV(K - 1)
    Next
    Printer.EndDoc
    Screen.MousePointer = vbNormal
End Sub
```

### **BalancePotencia.frm**

```
Dim resulnodeBP As Node
Dim m As Integer
```

```
Private Sub Combo4_GotFocus()
    Data1.Recordset.MoveFirst
    While Data1.Recordset.EOF = False
        Combo4.AddItem Data1.Recordset.Fields("Modelo")
        Data1.Recordset.MoveNext
    Wend
End Sub
```

```
Private Sub Combo5_GotFocus()
    Data2.Recordset.MoveFirst
    While Data2.Recordset.EOF = False
        Combo5.AddItem Data2.Recordset.Fields("Modelo")
        Data2.Recordset.MoveNext
    Wend
End Sub
```

```
Private Sub Combo6_GotFocus()
    Data3.Recordset.MoveFirst
    While Data3.Recordset.EOF = False
        Combo6.AddItem Data3.Recordset.Fields("Modelo")
        Data3.Recordset.MoveNext
    Wend
End Sub
```

```
Private Sub GuardarComo_Click()
```

---



```
Dim path As String
Dim puntero
GuardarBPT.TipoSeleccion = Index
GuardarBPT.Show vbModal
If Trim(GuardarBPT.Retorna) <> "" Then
    path = GuardarBPT.Retorna
End If
puntero = FreeFile
If path <> "" Then
    Open path For Output As #puntero
    Write #puntero, LOnda.Text, Distancia.Text, NRg.Text, MargS.Text, TipoConector.Text, _
    Perdidas.Text, Combo4.Text, Combo5.Text, Combo6.Text
End If
Close #puntero
End Sub

Private Sub Form_Unload(Cancel As Integer)
Unload ResultadosBP
End Sub

Private Sub Form_Load()
m = 0
Set resulnodeBP = ResultadosBP.TreeView1.Nodes.Add()
    resulnodeBP.Text = "BALANCE DE POTENCIAS"
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\fibras.mdb", vbCritical: Exit Sub
Data1.DatabaseName = App.path & "\emisores.mdb"
Data2.DatabaseName = App.path & "\receptores.mdb"
Data3.DatabaseName = App.path & "\fibras.mdb"
Data1.Refresh
Data2.Refresh
Data3.Refresh
End Sub

Private Sub BPotencia_Click()
Dim balp As Boolean, ptx As Single, prx As Single, afib As Single, acon As Single
Dim MSeg As Single, lenl As Single, lmax As Single, NRegt As Single
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominx As Integer, lominfo As Integer
Dim LOenl As Integer
Dim vb As Integer, vbl As Boolean, causa As String
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo4.Text)) Then
If Data2.Recordset.RecordCount > 0 Then

Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo5.Text)) Then
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
```



```
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo6.Text)) Then
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
If vb = 0 Then
vbl = True
Else
vbl = False
End If
If vbl = True Then
ptx = Data1.Recordset.Fields("Potencia")
prx = Data2.Recordset.Fields("Potencia")
afib = Data3.Recordset.Fields("Atenuacion")
lenl = Distancia.Text
MSeg = MargS.Text
NRegt = NRg.Text
acon = Perdidas.Text
m = m + 1
lmax = (ptx - (prx + 2 * acon + MSeg)) / afib
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
If balp = False Then
'Set resulnodeBP = ResultadosBP.TreeView1.Nodes.Add(m)
'resulnodeBP.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
resulBP(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
"Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
"y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
vbCrLf & _
"Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf & _
_
_
ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + " &
MSeg & vbCrLf & _
vbCrLf

resulBPR(m) = "El Balance de Potencias ha resultado negativo" & vbCrLf & _
"debiendo reducirse la distancia del enlace en " & Redondeo(Abs(lmax - lenl)) &
"Kms"
"El Balance de Potencias KK" & vbCrLf
Else
'Set resulnodeBP = ResultadosBP.TreeView1.Nodes.Add(m)
```



```
'resulnodeBP.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
resulBP(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    "Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
    "y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    vbCrLf & _
    "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf &
_
    ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + " &
MSeg & vbCrLf & _
    vbCrLf
resulBPR(m) = "El Balance de Potencias ha resultado positivo" & vbCrLf & _
    "pudiendo aumentar la distancia del enlace en " & Redondeo(Abs(lmax - lenl)) &
"Kms"
End If
'ResultadosBP.Show
End If
If vbl = False Then
Select Case vb
Case 1
    causa = "del tipo de conector usado para la fibra óptica"
Case 2
    causa = "del tipo de conector usado para el transmisor"
Case 3
    causa = "del tipo de conector usado para el receptor"
Case 4
    causa = "de la longitud de onda del transmisor"
Case 5
    causa = "de la longitud de onda del receptor"
Case 6
    causa = "de la longitud de onda de la fibra óptica"
End Select
m = m + 1
resulBP(m) = vbCrLf & _
    vbCrLf & _
    "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
'MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
End If
Set resulnodeBP = ResultadosBP.TreeView1.Nodes.Add(m)
resulnodeBP.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
End If

Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
```



```
Loop
End If
ResultadosBP.Show
End Sub
```

```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
    Cancel = True
    MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```

```
Private Sub MargS_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(MargS) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub NRg_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(NRg) Then
    Cancel = True
    MsgBox "El valor numérico de regeneradores debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TipoConector_Change()
    TipoConector = UCase(TipoConector)
    TipoConector.SelStart = Len(TipoConector)
End Sub
```



## GuardarBPT.frm

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
    Dim extension As String
    extension = ".bpt"
    If Right(Dir1.path, 1) = "\" Then
        Ret = Dir1.path & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    Else
        Ret = Dir1.path & "\" & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    End If
    Unload Me
End Sub

Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub
Private Sub File1_Click()
    Ret = Dir1.path & Text1.Text
End Sub
```



```
Private Sub Form_Load()  
    If App.PreviousInstance Then Ret = "": Unload Me  
    Ret = ""  
    Dir1.path = Drive1.Drive  
    File1.path = Dir1.path  
    Ret = Dir1.path & Text1.Text  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    Retorna = Ret  
End Sub  
  
Public Property Get TipoSeleccion() As Variant  
    TipoSeleccion = Tipo  
End Property  
  
Public Property Let TipoSeleccion(ByVal vNewValue As Variant)  
    Tipo = vNewValue  
End Property
```

## **ResultadosBP.frm**

```
Private Sub Imprimir_Click()  
    Dim Cabecera As String  
    Dim Cuerpo As String  
    Dim K As Integer  
    Screen.MousePointer = vbHourglass  
    Printer.Print vbCrLf & vbCrLf & vbCrLf  
    Printer.Font.Size = 14  
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf  
    For K = 2 To TreeView1.Nodes.Count  
        Printer.Font.Size = 10  
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf  
        Printer.Font.Size = 8  
        Printer.Print resulV(K - 1)  
    Next  
    Printer.EndDoc  
    Screen.MousePointer = vbNormal  
End Sub  
  
Private Sub TreeView1_NodeClick(ByVal Node As Node)  
    Text1.Text = resulBP(Node.Index - 1)  
    Text2.Text = resulBPR(Node.Index - 1)  
End Sub
```



## BalanceTiempo.frm

Dim resulnodeBT As Node  
Dim m As Integer

```
Private Sub Combo4_GotFocus()  
Data1.Recordset.MoveFirst  
While Data1.Recordset.EOF = False  
    Combo4.AddItem Data1.Recordset.Fields("Modelo")  
    Data1.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo2_GotFocus()  
Data2.Recordset.MoveFirst  
While Data2.Recordset.EOF = False  
    Combo2.AddItem Data2.Recordset.Fields("Modelo")  
    Data2.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo3_GotFocus()  
Data3.Recordset.MoveFirst  
While Data3.Recordset.EOF = False  
    Combo3.AddItem Data3.Recordset.Fields("Modelo")  
    Data3.Recordset.MoveNext  
Wend  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
Unload ResultadosBT  
End Sub
```

```
Private Sub Form_Load()  
m = 0  
Set resulnodeBT = ResultadosBT.TreeView1.Nodes.Add()  
    resulnodeBT.Text = "BALANCE DE TIEMPO"  
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\emisores.mdb", vbCritical: Exit Sub  
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\receptores.mdb", vbCritical: Exit Sub  
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\fibras.mdb", vbCritical: Exit Sub  
Data1.DatabaseName = App.path & "\emisores.mdb"  
Data2.DatabaseName = App.path & "\receptores.mdb"  
Data3.DatabaseName = App.path & "\fibras.mdb"  
Data1.Refresh  
Data2.Refresh  
Data3.Refresh  
End Sub
```



```
Private Sub BTiempo_Click()
Dim balt As Boolean
Dim lenl As Single, NRegt As Single, vel As Single, restvel As Single
Dim cod As Boolean, cdg As Single, cogigo As String
Dim disp As Single, ilong As Single, ttx As Single, trx As Single
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominrx As Integer, lominfo As Integer
Dim LOenl As Integer
Dim vb As Integer, vbl As Boolean, causa As String
Dim TmatCuad As Single, TresCuad As Single, Tres As Single
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo4.Text)) Then
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo2.Text)) Then
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo3.Text)) Then
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
If vb = 0 Then
ttx = Data1.Recordset.Fields("Tiempo_Respuesta")
trx = Data2.Recordset.Fields("Tiempo_Respuesta")
ilong = Data1.Recordset.Fields("Incremento_LOnda")
disp = Data3.Recordset.Fields("Dispersion")
vel = Velocidad.Text
lenl = Distancia.Text
NRegt = NRg.Text
If NRZ.Value = True Then
cod = False
Else
cod = True
End If
TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
Tres = Sqr(TresCuad)
```



```
If cod = False Then
  cdg = 0.7
  codigo = "NRZ"
Else
  cdg = 0.35
  codigo = "RZ"
End If
restvel = Abs(vel - (cdg / (0.001 * Tres)))
m = m + 1
balt = baltime(displ, ilong, lenl, ttx, trx, vel, cod, NRegt)
If balt = False Then
  'Set resulnodeBT = ResultadosBT.TreeView1.Nodes.Add(m)
  'resulnodeBT.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  resulBT(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    "Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
    Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " & NRegt
& " )) ^ 2" & vbCrLf & _
    vbCrLf & _
    "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
    Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & " +
" & Redondeo(TmatCuad) & vbCrLf & _
    vbCrLf & _
    "NRZ => Tresp < 0.7/Velocidad  RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
    codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = " &
Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
    vbCrLf
  resulBTR(m) = "El Balance de Tiempo ha resultado negativo" & vbCrLf & _
    "debiendo reducirse la velocidad del enlace en " & Redondeo(restvel) & "Mb/s"

Else
  'Set resulnodeBT = ResultadosBT.TreeView1.Nodes.Add(m)
  'resulnodeBT.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo")
  resulBT(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    "Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
    Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " & NRegt
& " )) ^ 2" & vbCrLf & _
```



```
vbCrLf & _
  "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
  Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & " +
" & Redondeo(TmatCuad) & vbCrLf & _
  vbCrLf & _
  "NRZ => Tresp < 0.7/Velocidad  RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
  codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = " &
Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
  vbCrLf
  resulBTR(m) = "El Balance de Tiempo ha resultado positivo" & vbCrLf & _
  "pudiendo aumentar la velocidad del enlace en " & Redondeo(restvel) & "Mb/s"
End If
'ResultadosBT.Show
Else
Select Case vb
Case 1
  causa = "del tipo de conector usado para la fibra óptica"
Case 2
  causa = "del tipo de conector usado para el transmisor"
Case 3
  causa = "del tipo de conector usado para el receptor"
Case 4
  causa = "de la longitud de onda del transmisor"
Case 5
  causa = "de la longitud de onda del receptor"
Case 6
  causa = "de la longitud de onda de la fibra óptica"
End Select
m = m + 1
resulBT(m) = vbCrLf & _
  vbCrLf & _
  "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
'MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
End If
Set resulnodeBT = ResultadosBT.TreeView1.Nodes.Add(m)
resulnodeBT.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
ResultadosBT.Show
End Sub
```



```
Private Sub GuardarComo_Click()  
Dim path As String  
Dim puntero  
GuardarBTP.TipoSeleccion = Index  
GuardarBTP.Show vbModal  
If Trim(GuardarBTP.Retorna) <> "" Then  
    path = GuardarBTP.Retorna  
End If  
puntero = FreeFile  
If path <> "" Then  
    Open path For Output As #puntero  
    Write #puntero, Londa.Text, Velocidad.Text, Distancia.Text, NRg.Text, TipoConector.Text, _  
    NRZ.Value, Combo4.Text, Combo2.Text, Combo3.Text  
End If  
Close #puntero  
End Sub
```

```
Private Sub Londa_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Londa) Then  
    Cancel = True  
    MsgBox "La longitud de onda debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub NRg_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(NRg) Then  
    Cancel = True  
    MsgBox "El valor numérico de regeneradores debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Distancia_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Distancia) Then  
    Cancel = True  
    MsgBox "La distancia del enlace debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub TipoConector_Change()  
    TipoConector = UCase(TipoConector)  
    TipoConector.SelStart = Len(TipoConector)  
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Velocidad) Then  
    Cancel = True  
    MsgBox "La velocidad debe ser un valor numérico"  
End If  
End Sub
```



## GuardarBTP.frm

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
    Dim extension As String
    extension = ".btp"
    If Right(Dir1.path, 1) = "\" Then
        Ret = Dir1.path & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    Else
        Ret = Dir1.path & "\" & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    End If
    Unload Me
End Sub

Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub
Private Sub File1_Click()
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Load()
    If App.PreviousInstance Then Ret = "": Unload Me
    Ret = ""
    Dir1.path = Drive1.Drive
    File1.path = Dir1.path
    Ret = Dir1.path & Text1.Text
End Sub
```



```
Private Sub Form_Unload(Cancel As Integer)
```

```
    Retorna = Ret
```

```
End Sub
```

```
Public Property Get TipoSeleccion() As Variant
```

```
    TipoSeleccion = Tipo
```

```
End Property
```

```
Public Property Let TipoSeleccion(ByVal vNewValue As Variant)
```

```
    Tipo = vNewValue
```

```
End Property
```

## **ResultadosBT.frm**

```
Private Sub Imprimir_Click()
```

```
    Dim Cabecera As String
```

```
    Dim Cuerpo As String
```

```
    Dim K As Integer
```

```
    Screen.MousePointer = vbHourglass
```

```
    Printer.Print vbCrLf & vbCrLf & vbCrLf
```

```
    Printer.Font.Size = 14
```

```
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf
```

```
    For K = 2 To TreeView1.Nodes.Count
```

```
        Printer.Font.Size = 10
```

```
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf
```

```
        Printer.Font.Size = 8
```

```
        Printer.Print resulV(K - 1)
```

```
    Next
```

```
    Printer.EndDoc
```

```
    Screen.MousePointer = vbNormal
```

```
End Sub
```

```
Private Sub TreeView1_NodeClick(ByVal Node As Node)
```

```
    Text1.Text = resulBT(Node.Index - 1)
```

```
    Text2.Text = resulBTR(Node.Index - 1)
```

```
End Sub
```



## Enlace.frm

```
Dim resulnodeEnl As Node
Dim m As Integer
Dim presupf As Single
```

```
Private Sub Combo1_GotFocus()
Data1.Recordset.MoveFirst
While Data1.Recordset.EOF = False
'If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then
Combo1.AddItem Data1.Recordset.Fields("Modelo")
Data1.Recordset.MoveNext
Wend
End Sub
```

```
Private Sub Combo2_GotFocus()
Data2.Recordset.MoveFirst
While Data2.Recordset.EOF = False
Combo2.AddItem Data2.Recordset.Fields("Modelo")
Data2.Recordset.MoveNext
Wend
End Sub
```

```
Private Sub Combo3_GotFocus()
Data3.Recordset.MoveFirst
While Data3.Recordset.EOF = False
Combo3.AddItem Data3.Recordset.Fields("Modelo")
Data3.Recordset.MoveNext
Wend
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
Unload ResultadosEnl
End Sub
```

```
Private Sub Form_Load()
m = 0
presupf = 1000000
Set resulnodeEnl = ResultadosEnl.TreeView1.Nodes.Add()
resulnodeEnl.Text = "ENLACES ANALIZADOS"
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\fibras.mdb", vbCritical: Exit Sub
Data1.DatabaseName = App.path & "\emisores.mdb"
Data2.DatabaseName = App.path & "\receptores.mdb"
Data3.DatabaseName = App.path & "\fibras.mdb"
Data1.Refresh
Data2.Refresh
Data3.Refresh
End Sub
```



```
Private Sub Command1_Click()
Dim balp As Boolean, balt As Boolean, cod As Boolean, codigo As String
Dim ptx As Single, prx As Single, afib As Single, acon As Single
Dim MSeg As Single, lenl As Single, NRegt As Single, vel As Single, restvel As Single
Dim disp As Single, ilong As Single, ttx As Single, trx As Single
Dim Ctx As Single, cadc As Single, Crx As Single, campf As Single
Dim cfib As Single, presup As Single, ccon As Single
Dim conectortx As String, conectorrx As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominrx As Integer, lominfo As Integer
Dim LOenl As Integer
Dim vb As Integer, vbl As Boolean, causa As String
Dim TmatCuad As Single, TresCuad As Single, Tres As Single
If Combo1.Text = "Lista de transmisores" Then
  MsgBox "Seleccione un elemento de la lista de transmisores"
End If
If Combo2.Text = "Lista de receptores" Then
  MsgBox "Seleccione un elemento de la lista de receptores"
End If
If Combo3.Text = "Lista de fibras ópticas" Then
  MsgBox "Seleccione un elemento de la lista de fibras ópticas"
End If
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(Combo1.Text)) Then
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(Combo2.Text)) Then
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(Combo3.Text)) Then
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorrx = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominrx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOenl)
If vb = 0 Then
vbl = True
Else
vbl = False
End If
```



```
If vbl = True Then
    ptx = Data1.Recordset.Fields("Potencia")
    prx = Data2.Recordset.Fields("Potencia")
    ttx = Data1.Recordset.Fields("Tiempo_Respuesta")
    trx = Data2.Recordset.Fields("Tiempo_Respuesta")
    Ctx = Data1.Recordset.Fields("Precio")
    Crx = Data2.Recordset.Fields("Precio")
    cadc = Data1.Recordset.Fields("Precio_CtoAdicional")
    campf = Data2.Recordset.Fields("Precio_CtoAmplificador")
    cfib = Data3.Recordset.Fields("Precio/metro")
    ilong = Data1.Recordset.Fields("Incremento_LOnda")
    afib = Data3.Recordset.Fields("Atenuacion")
    disp = Data3.Recordset.Fields("Dispersion")
    vel = Velocidad.Text
    lenl = Distancia.Text
    MSeg = MargS.Text
    NRegt = NRg.Text
    ccon = Precio.Text
    acon = Perdidas.Text
    If NRZ.Value = True Then
        cod = False
    Else
        cod = True
    End If
    TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
    TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
    Tres = Sqr(TresCuad)
    If cod = False Then
        cdg = 0.7
        codigo = "NRZ"
    Else
        cdg = 0.35
        codigo = "RZ"
    End If
    m = m + 1
    'Set resulnodeEnl = TreeView1.Nodes.Add()
    'resulnodeEnl.Text = "ASOCIACIONES POSIBLES"
    balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
    If balp = False Then
        lmax = (ptx - (prx + 2 * acon + MSeg)) / afib
        resulEnl(m) = vbCrLf & _
            "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
            vbCrLf & _
            "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
            "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
            "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
            vbCrLf & _
            "Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
            "y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
            vbCrLf & _
            "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
            vbCrLf & _
            "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf &
```



```
vbCrLf & _
ptx & " >=" & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + " &
MSeg & vbCrLf
resulEnIR(m) = "El Balance de Potencias ha resultado negativo" & vbCrLf & _
"debiendo reducirse la distancia del enlace en " & Redondeo(Abs(lmax - lenl)) &
"Kms"
'Falta dirigirlo al final de la funcion
GoTo fin
End If
balt = baltime(displ, ilong, lenl, ttx, trx, vel, cod, NRegt)
If balt = False Then
restvel = Abs(vel - (cdg / (0.001 * Tres)))
resulEnI(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
vbCrLf & _
"Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " &
NRegt & " )) ^ 2" & vbCrLf & _
"Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & "
+ " & Redondeo(TmatCuad) & vbCrLf & _
"NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = "
& Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
vbCrLf
resulEnIR(m) = "El Balance de Tiempo ha resultado negativo" & vbCrLf & _
"debiendo reducirse la velocidad del enlace en " & Redondeo(restvel) & "Mb/s"
'Falta dirigirlo al final de la funcion
GoTo fin
End If
presup = coste(Ctx, cadc, Crx, campf, ccon, lenl, cfib, NRegt)
If presup < presupf Then
presupf = presup
resulEnIR(0) = vbCrLf & _
"EL enlace óptimo es" & vbCrLf & _
Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo") & _
vbCrLf & _
"con un coste de " & Redondeo(presupf) & " €" & vbCrLf & _
"requiriendo la solución " & NRegt & " nodos intermedios"
End If
'Set resulenl = Menus.TreeView1.Nodes.Add(m)
'resulenl.Text = Data1.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
```



```
resulEnl(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    "Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
    "y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    vbCrLf & _
    "Balance de Potencia" & vbCrLf & _
    "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf
& _
    ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + "
& MSeg & vbCrLf & _
    vbCrLf & _
    "Balance de Tiempos" & vbCrLf & _
    "Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
    Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " &
NRegt & " )) ^ 2" & vbCrLf & _
    "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
    Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & "
+ " & Redondeo(TmatCuad) & vbCrLf & _
    "NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
    codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = "
& Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
    vbCrLf
resulEnlR(m) = "Presupuesto para el Enlace de Fibra seleccionado " & Redondeo(presup) & "
€"
fin:
End If
If vbl = False Then
Select Case vb
Case 1
    causa = "del tipo de conector usado para la fibra óptica"
Case 2
    causa = "del tipo de conector usado para el transmisor"
Case 3
    causa = "del tipo de conector usado para el receptor"
Case 4
    causa = "de la longitud de onda del transmisor"
Case 5
    causa = "de la longitud de onda del receptor"
Case 6
    causa = "de la longitud de onda de la fibra óptica"
End Select
m = m + 1
resulEnl(m) = vbCrLf & _
    vbCrLf & _
    "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
'MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
End If
```



```
Set resulnodeEnl = ResultadosEnl.TreeView1.Nodes.Add(m)
resulnodeEnl.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
ResultadosEnl.Show
End Sub
```

```
Private Sub GuardarComo_Click()
Dim path As String
Dim puntero
GuardarENL.TipoSeleccion = Index
GuardarENL.Show vbModal
If Trim(GuardarENL.Retorna) <> "" Then
path = GuardarENL.Retorna
End If
puntero = FreeFile
If path <> "" Then
Open path For Output As #puntero
Write #puntero, LOnda.Text, Velocidad.Text, Distancia.Text, NRg.Text, MargS.Text, _
TipoConector.Text, Perdidas.Text, Precio.Text, NRZ.Value, _
Combo1.Text, Combo2.Text, Combo3.Text
End If
Close #puntero
End Sub
```

```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
Cancel = True
MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```

```
Private Sub MargS_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(MargS) Then
Cancel = True
MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```



```
Private Sub NRg_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(NRg) Then
    Cancel = True
    MsgBox "El valor numérico de regeneradores debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TipoConector_Change()
    TipoConector = UCase(TipoConector)
    TipoConector.SelStart = Len(TipoConector)
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```



## GuardarENL.frm

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
    Dim extension As String
    extension = ".enl"
    If Right(Dir1.path, 1) = "\" Then
        Ret = Dir1.path & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    Else
        Ret = Dir1.path & "\" & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    End If
    Unload Me
End Sub

Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub
Private Sub File1_Click()
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Load()
    If App.PreviousInstance Then Ret = "": Unload Me
    Ret = ""
    Dir1.path = Drive1.Drive
    File1.path = Dir1.path
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Retorna = Ret
End Sub
```

---



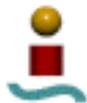
```
Public Property Get TipoSeleccion() As Variant
    TipoSeleccion = Tipo
End Property
```

```
Public Property Let TipoSeleccion(ByVal vNewValue As Variant)
    Tipo = vNewValue
End Property
```

## **ResultadosEnl.frm**

```
Private Sub Imprimir_Click()
    Dim Cabecera As String
    Dim Cuerpo As String
    Dim K As Integer
    Screen.MousePointer = vbHourglass
    Printer.Print vbCrLf & vbCrLf & vbCrLf
    Printer.Font.Size = 14
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf
    For K = 2 To TreeView1.Nodes.Count
        Printer.Font.Size = 10
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf
        Printer.Font.Size = 8
        Printer.Print resulV(K - 1)
    Next
    Printer.EndDoc
    Screen.MousePointer = vbNormal
End Sub

Private Sub TreeView1_NodeClick(ByVal Node As Node)
    Text1.Text = resulEnl(Node.Index - 1)
    Text2.Text = resulEnlR(Node.Index - 1)
End Sub
```



## ComponenteOptimo.frm

```
Dim resulnodeCOpt As Node
Dim m As Integer
```

```
Private Sub Form_Unload(Cancel As Integer)
Unload ResultadosCOpt
End Sub
```

```
Private Sub Form_Load()
m = 0
Set resulnodeCOpt = ResultadosCOpt.TreeView1.Nodes.Add()
resulnodeCOpt.Text = "COMPONENTES ANALIZADOS"
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &
App.path & "\fibras.mdb", vbCritical: Exit Sub
Data1.DatabaseName = App.path & "\emisores.mdb"
Data2.DatabaseName = App.path & "\receptores.mdb"
Data3.DatabaseName = App.path & "\fibras.mdb"
Data1.Refresh
Data2.Refresh
Data3.Refresh
End Sub
```

```
Private Sub Command1_Click()
Dim balp As Boolean, balt As Boolean, presup As Single
Dim ptx As Single, prx As Single, afib As Single, acon As Single
Dim MSeg As Single, lenl As Single, NRegt As Single, vel As Single
Dim disp As Single, ilong As Single, ttx As Single, trx As Single
Dim Ctx As Single, cadc As Single, Crx As Single, campf As Single, ccon As Single
Dim cod As Boolean, Criterio As Boolean
Dim cfib As Single, lmax As Single, presupf As Single, distmax As Single, velrest As Single
Dim i As Integer, j As Integer, K As Integer, a As Integer, b As Integer, c As Integer
Dim R As Integer, s As Integer, t As Integer
Dim listatx() As String, listarx() As String, listafo() As String
Dim conectortx As String, conectorr As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominx As Integer, lominfo As Integer
Dim LOenl As Integer
Dim vb As Integer, causa As String
presupf = 100000000
distmax = 0
velrest = 0
'Si no se visualiza ningún elemento, se selecciona el primero
If Combo1.Text = "" And CheckTx.Value = 1 Then
Data1.Recordset.MoveFirst
Combo1.Text = Data1.Recordset.Fields("Modelo")
MsgBox "Debe seleccionar un elemento de la lista de transmisores"
End If
```



```
If Combo2.Text = "" And CheckRx.Value = 1 Then
    Data2.Recordset.MoveFirst
    Combo2.Text = Data2.Recordset.Fields("Modelo")
    MsgBox "Debe seleccionar un elemento de la lista de receptores"
End If
If Combo3.Text = "" And CheckFO.Value = 1 Then
    Data3.Recordset.MoveFirst
    Combo3.Text = Data3.Recordset.Fields("Modelo")
    MsgBox "Debe seleccionar un elemento de la lista de fibras ópticas"
End If
If CheckTx.Value = 1 Then
    i = 1
    ReDim listatx(i - 1)
    listatx(i - 1) = Data1.Recordset.Fields("Modelo")
Else
    'Calcula el número de elementos que hay almacenados en la base de datos
    Data1.Recordset.MoveFirst
    i = 1
    While Data1.Recordset.EOF = False
        i = i + 1
        Data1.Recordset.MoveNext
    Wend
    'Especifica el tamaño de la lista
    ReDim listatx(i - 1)
    'Rellena la lista con el campo "Modelo" de la base de datos de transmisores
    Data1.Recordset.MoveFirst
    For a = 0 To i - 2
        listatx(a) = Data1.Recordset.Fields("Modelo")
        If Data1.Recordset.EOF = False Then
            Data1.Recordset.MoveNext
        Else
            a = i - 1
        End If
    Next a
End If
If CheckRx.Value = 1 Then
    j = 1
    ReDim listarx(j - 1)
    listarx(j - 1) = Data2.Recordset.Fields("Modelo")
Else
    Data2.Recordset.MoveFirst
    j = 1
    While Data2.Recordset.EOF = False
        j = j + 1
        Data2.Recordset.MoveNext
    Wend
    ReDim listarx(j - 1)
    Data2.Recordset.MoveFirst
    For b = 0 To j - 2
        listarx(b) = Data2.Recordset.Fields("Modelo")
        If Data2.Recordset.EOF = False Then
            Data2.Recordset.MoveNext
        Else
            b = j - 1
        End If
    End If
```



```
Next b
End If
If CheckFO.Value = 1 Then
    K = 1
    ReDim listafo(K - 1)
    listafo(K - 1) = Data3.Recordset.Fields("Modelo")
Else
    Data3.Recordset.MoveFirst
    K = 1
    While Data3.Recordset.EOF = False
        K = K + 1
        Data3.Recordset.MoveNext
    Wend
    ReDim listafo(K - 1)
    Data3.Recordset.MoveFirst
    For c = 0 To K - 2
        listafo(c) = Data3.Recordset.Fields("Modelo")
        If Data3.Recordset.EOF = False Then
            Data3.Recordset.MoveNext
        Else
            c = K - 1
        End If
    Next c
End If
Dim mat() As String
ReDim mat((i * j * K) - 1, 3)
    For R = 0 To i - 1
        Data1.Recordset.MoveFirst
        For s = 0 To j - 1
            Data1.Recordset.MoveFirst
            Data2.Recordset.MoveFirst
            For t = 0 To K - 1
                Data1.Recordset.MoveFirst
                Data2.Recordset.MoveFirst
                Data3.Recordset.MoveFirst
                ' Busco el Transmisor en la Base de Datos de Emisores
                If Data1.Recordset.RecordCount > 0 Then
                    Do Until Data1.Recordset.EOF
                        If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(listatx(R))) Then
                            'Busco el Receptor
                            If Data2.Recordset.RecordCount > 0 Then
                                Do Until Data2.Recordset.EOF
                                    If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(listarx(s))) Then
                                        'Busco la Fibra
                                        If Data3.Recordset.RecordCount > 0 Then
                                            Do Until Data3.Recordset.EOF
                                                If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(listafo(t))) Then
                                                    conectortx = Data1.Recordset.Fields("Tipo_Conector")
                                                    conectorrax = Data2.Recordset.Fields("Tipo_Conector")
                                                    conectorfo = Data3.Recordset.Fields("Tipo_Conector")
                                                    conectortipo = TipoConector.Text
```



```
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = LOnda.Text
vb = viab(conectortipo, conectortx, conectorr, conectorfo, _
lomaxtx, lomaxrx, lomaxfo, lomintx, lominx, lominfo, LOenl)
If vb = 0 Then
    ptx = Data1.Recordset.Fields("Potencia")
    prx = Data2.Recordset.Fields("Potencia")
    ilong = Data1.Recordset.Fields("Incremento_LOnda")
    ttx = Data1.Recordset.Fields("Tiempo_Respuesta")
    trx = Data2.Recordset.Fields("Tiempo_Respuesta")
    Ctx = Data1.Recordset.Fields("Precio")
    Crx = Data2.Recordset.Fields("Precio")
    cadc = Data1.Recordset.Fields("Precio_CtoAdicional")
    campf = Data2.Recordset.Fields("Precio_CtoAmplificador")
    ccon = Data3.Recordset.Fields("Precio/metro")
    afib = Data3.Recordset.Fields("Atenuacion")
    disp = Data3.Recordset.Fields("Dispersion")
    vel = Velocidad.Text
    lenl = Distancia.Text
    MSeg = MargS.Text
    NRegt = NRg.Text
    ccon = Precio.Text
    acon = Perdidas.Text
    If NRZ.Value = True Then
        cod = False
    Else
        cod = True
    End If
    TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
    TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
    Tres = Sqr(TresCuad)
    If cod = False Then
        cdg = 0.7
        codigo = "NRZ"
    Else
        cdg = 0.35
        codigo = "RZ"
    End If
    restvel = Abs(vel - (cdg / (0.001 * Tres)))
    m = m + 1
    If bpot.Value = True Then
        balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
        While balp = False
            NRegt = NRegt + 1
            balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
        Wend
        lmax = (ptx - (prx + 2 * acon + MSeg)) / afib
```



```
If lmax > distmax Then
  distmax = lmax
  resulcompR(0) = vbCrLf & _
  "EL enlace óptimo es" & vbCrLf & _
  Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
  Data2.Recordset.Fields("Modelo") & _
  vbCrLf & _
  "permitiéndonos aumentar la distancia del enlace en " & Redondeo(Abs(distmax - lenl))
  & " Km" & vbCrLf & _
  "requiriendo la solución " & NRegt & " nodos intermedios"
End If
resulcomp(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
"Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
"y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
vbCrLf & _
"Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf &
-
Data1.Recordset.Fields("Potencia") & " >= " & prx & " + 2 x " & acon & " + " & afib & "
x " & "Lmax" & " + " & MSeg & vbCrLf & _
vbCrLf
resulcompR(m) = "El Balance de Potencias ha resultado positivo " & vbCrLf & _
"pudiendo aumentar la distancia del enlace en " & Redondeo(Abs(lmax - lenl)) &
"Kms"
End If
If btime.Value = True Then
  balt = baltime(displ, ilong, lenl, ttx, trx, vel, cod, NRegt)
  While balt = False
    NRegt = NRegt + 1
    balt = baltime(displ, ilong, lenl, ttx, trx, vel, cod, NRegt)
  Wend
  If restvel > velrest Then
    velrest = restvel
    resulcompR(0) = vbCrLf & _
    "EL enlace óptimo es" & vbCrLf & _
    Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
    Data2.Recordset.Fields("Modelo") & _
    vbCrLf & _
    "permitiéndonos aumentar la velocidad de transmisión en " & Redondeo(velrest) & "
    Mb/s" & vbCrLf & _
    "requiriendo la solución " & NRegt & " nodos intermedios"
  End If
```



```
resulcomp(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
"Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " & NRegt
& " )) ^ 2" & vbCrLf & _
vbCrLf & _
"Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & " +
" & Redondeo(TmatCuad) & vbCrLf & _
vbCrLf & _
"NRZ => Tresp < 0.7/Velocidad  RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = " &
Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
vbCrLf
resulcompR(m) = "El Balance de Tiempo ha resultado positivo" & vbCrLf & _
"pudiendo aumentar la velocidad del enlace en " & Redondeo(restvel) & "Mb/s"
End If
If costenl.Value = True Then
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
While balp = False
NRegt = NRegt + 1
balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)
Wend
balt = baltime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)
While balt = False
NRegt = NRegt + 1
balt = baltime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)
Wend
presup = coste(Ctx, cadc, Crx, campf, ccon, lenl, cfib, NRegt)
If presup < presupf Then
presupf = presup
resulcompR(0) = vbCrLf & _
"EL enlace óptimo es" & vbCrLf & _
Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
Data2.Recordset.Fields("Modelo") & _
vbCrLf & _
"con un coste de " & Redondeo(presupf) & " €" & vbCrLf & _
"requiriendo la solución " & NRegt & " nodos intermedios"
End If
```



```
resulcomp(m) = vbCrLf & _
"CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
vbCrLf & _
"Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
"Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
"Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
"Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
"Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
"y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
vbCrLf & _
"RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
vbCrLf & _
"Balance de Potencia" & vbCrLf & _
"Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf
& _
    ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + "
& MSeg & vbCrLf & _
    vbCrLf & _
    "Balance de Tiempos" & vbCrLf & _
    "Tfo^2 = Tmat^2 = (Disp * lLong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
    Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " &
NRegt & " )) ^ 2" & vbCrLf & _
    "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
    Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & "
+ " & Redondeo(TmatCuad) & vbCrLf & _
    "NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
    codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = "
& Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
    vbCrLf
    resulcompR(m) = "Presupuesto para el Enlace de Fibra seleccionado " &
Redondeo(presup) & " €"
End If
Set resulnodeCOpt = ResultadosCOpt.TreeView1.Nodes.Add(m)
resulnodeCOpt.Text = Data1.Recordset.Fields("Modelo") & "-" &
Data3.Recordset.Fields("Modelo") & "-" & Data2.Recordset.Fields("Modelo")
End If
End If
Data3.Recordset.MoveNext
Loop
End If
End If
Data2.Recordset.MoveNext
Loop
End If
End If
Data1.Recordset.MoveNext
Loop
End If
Next t
Next s
Next R
ResultadosCOpt.Show
```



```
If vb <> 0 Then
Select Case vb
Case 1
causa = "del tipo de conector usado para la fibra óptica"
Case 2
causa = "del tipo de conector usado para el transmisor"
Case 3
causa = "del tipo de conector usado para el receptor"
Case 4
causa = "de la longitud de onda del transmisor"
Case 5
causa = "de la longitud de onda del receptor"
Case 6
causa = "de la longitud de onda de la fibra óptica"
End Select
MsgBox "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
End If
End Sub

Private Sub GuardarComo_Click()
Dim path As String
Dim puntero
GuardarCOP.TipoSeleccion = Index
GuardarCOP.Show vbModal
If Trim(GuardarCOP.Retorna) <> "" Then
path = GuardarCOP.Retorna
End If
puntero = FreeFile
If path <> "" Then
Open path For Output As #puntero
Write #puntero, LOnda.Text, Velocidad.Text, Distancia.Text, NRg.Text, MargS.Text, _
TipoConector.Text, Perdidas.Text, Precio.Text, NRZ.Value, _
CheckTx.Value, CheckRx.Value, CheckFO.Value, Combo1.Text, Combo2.Text, Combo3.Text,
costenl.Value, bpot.Value, btime.Value
End If
Close #puntero
End Sub

Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
Cancel = True
MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub

Private Sub MargS_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(MargS) Then
Cancel = True
MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

---



```
Private Sub NRg_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(NRg) Then
    Cancel = True
    MsgBox "El valor numérico de regeneradores debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Distancia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Distancia) Then
    Cancel = True
    MsgBox "La distancia del enlace debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TipoConector_Change()
    TipoConector = UCase(TipoConector)
    TipoConector.SelStart = Len(TipoConector)
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```



## GuardarCOP.frm

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
    Dim extension As String
    extension = ".cop"
    If Right(Dir1.path, 1) = "\" Then
        Ret = Dir1.path & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    Else
        Ret = Dir1.path & "\" & Text1.Text & extension
        If Tipo = 1 Then Ret = Dir1.path
    End If
    Unload Me
End Sub

Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub
Private Sub File1_Click()
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Load()
    If App.PreviousInstance Then Ret = "": Unload Me
    Ret = ""
    Dir1.path = Drive1.Drive
    File1.path = Dir1.path
    Ret = Dir1.path & Text1.Text
End Sub
```



```
Private Sub Form_Unload(Cancel As Integer)
```

```
    Retorna = Ret
```

```
End Sub
```

```
Public Property Get TipoSeleccion() As Variant
```

```
    TipoSeleccion = Tipo
```

```
End Property
```

```
Public Property Let TipoSeleccion(ByVal vNewValue As Variant)
```

```
    Tipo = vNewValue
```

```
End Property
```

## **ResultadosCOpt.frm**

```
Private Sub Imprimir_Click()
```

```
    Dim Cabecera As String
```

```
    Dim Cuerpo As String
```

```
    Dim K As Integer
```

```
    Screen.MousePointer = vbHourglass
```

```
    Printer.Print vbCrLf & vbCrLf & vbCrLf
```

```
    Printer.Font.Size = 14
```

```
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf
```

```
    For K = 2 To TreeView1.Nodes.Count
```

```
        Printer.Font.Size = 10
```

```
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf
```

```
        Printer.Font.Size = 8
```

```
        Printer.Print resulV(K - 1)
```

```
    Next
```

```
    Printer.EndDoc
```

```
    Screen.MousePointer = vbNormal
```

```
End Sub
```

```
Private Sub TreeView1_NodeClick(ByVal Node As Node)
```

```
    Text1.Text = resulcomp(Node.Index - 1)
```

```
    Text2.Text = resulcompR(Node.Index - 1)
```

```
End Sub
```



## DiseñoEnl.frm

Option Explicit

Dim resulnode As Node  
Dim m As Integer  
Dim presupf As Single

Private Sub Form\_Unload(Cancel As Integer)  
Unload Resultados  
End Sub

Private Sub Form\_Load()  
m = 0  
presupf = 1000000  
Set resulnode = Resultados.TreeView1.Nodes.Add()  
resulnode.Text = "DISEÑO DEL ENLACE"  
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\emisores.mdb", vbCritical: Exit Sub  
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\receptores.mdb", vbCritical: Exit Sub  
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\fibras.mdb", vbCritical: Exit Sub  
Data1.DatabaseName = App.path & "\emisores.mdb"  
Data2.DatabaseName = App.path & "\receptores.mdb"  
Data3.DatabaseName = App.path & "\fibras.mdb"  
Data1.Refresh  
Data2.Refresh  
Data3.Refresh  
End Sub  
Private Sub Command1\_Click()  
SeleccionarEnlace.Show vbModal, Me  
End Sub

Private Sub Command2\_Click()  
If Not Existe(App.path & Tipo0) Then MsgBox "No se ha localizado el fichero .:" &  
App.path & Tipo0, vbCritical: Exit Sub  
SeleccionTx.Adodc1.ConnectionString = "File Name=" & App.path & Tipo0 & ";"  
SeleccionTx.Adodc1.RecordSource = "select \* from emisores order by Modelo"  
SeleccionTx.Adodc1.Refresh  
SeleccionTx.Show vbModal, Me  
End Sub

Private Sub Command3\_Click()  
If Not Existe(App.path & Tipo1) Then MsgBox "No se ha localizado el fichero .:" &  
App.path & Tipo1, vbCritical: Exit Sub  
SeleccionarRx.Adodc1.ConnectionString = "File Name=" & App.path & Tipo1 & ";"  
SeleccionarRx.Adodc1.RecordSource = "select \* from receptores order by Modelo"  
SeleccionarRx.Adodc1.Refresh  
SeleccionarRx.Show vbModal, Me  
End Sub



```
Private Sub Command4_Click()
'Dim resulnode As Node
Dim i As Integer, j As Integer, K As Integer
'Dim m As Integer
Dim balp As Boolean, balt As Boolean, presup As Single
Dim balpp As Boolean, balpt As Boolean, baltt As Boolean
Dim ptx As Single, prx As Single, afib As Single, acon As Single
Dim MSeg As Single, lenl As Single, NRegt As Single, vel As Single, cod As Boolean
Dim disp As Single, ilong As Single, ttx As Single, trx As Single
Dim Ctx As Single, cadc As Single, Crx As Single, campf As Single, ccon As Single
Dim cfib As Single, cdg As Single, restvel As Single
'Dim resul(100) As String
Dim conectortx As String, conectorr As String, conectorfo As String, conectortipo As String
Dim lomaxtx As Integer, lomaxrx As Integer, lomaxfo As Integer
Dim lomintx As Integer, lominx As Integer, lominfo As Integer
Dim LOenl As Integer, codigo As String
Dim vb As Integer, vbl As Boolean, causa As String
Dim TmatCuad As Single, TresCuad As Single, Tres As Single
'm = 1
For i = 0 To List1.ListCount - 1
Data1.Recordset.MoveFirst
For j = 0 To List2.ListCount - 1
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
For K = 0 To List3.ListCount - 1
Data1.Recordset.MoveFirst
Data2.Recordset.MoveFirst
Data3.Recordset.MoveFirst
'Set resulnode = Resultados.TreeView1.Nodes.Add(m)
'resulnode.Text = List1.List(i) & "-" & List2.List(j)
' Busco el Transmisor en la Base de Datos de Emisores
If Data1.Recordset.RecordCount > 0 Then
Do Until Data1.Recordset.EOF
If (RTrim(Data1.Recordset.Fields("Modelo")) = RTrim(List1.List(i - 1))) Then
'Busco el Receptor
If Data2.Recordset.RecordCount > 0 Then
Do Until Data2.Recordset.EOF
If (RTrim(Data2.Recordset.Fields("Modelo")) = RTrim(List2.List(j - 1))) Then
'Busco la Fibra
If Data3.Recordset.RecordCount > 0 Then
Do Until Data3.Recordset.EOF
If (RTrim(Data3.Recordset.Fields("Modelo")) = RTrim(List3.List(K - 1))) Then
conectortx = Data1.Recordset.Fields("Tipo_Conector")
conectorr = Data2.Recordset.Fields("Tipo_Conector")
conectorfo = Data3.Recordset.Fields("Tipo_Conector")
conectortipo = SeleccionarEnlace.TipoConector.Text
lomaxtx = Data1.Recordset.Fields("Longitud_Onda_Max")
lomaxrx = Data2.Recordset.Fields("Longitud_Onda_Max")
lomaxfo = Data3.Recordset.Fields("Longitud_Onda_Max")
lomintx = Data1.Recordset.Fields("Longitud_Onda_Min")
lominx = Data2.Recordset.Fields("Longitud_Onda_Min")
lominfo = Data3.Recordset.Fields("Longitud_Onda_Min")
LOenl = SeleccionarEnlace.LOnda.Text
```



```
vb = viab(conectortipo, conectortx, conectorrx, conectorfo, _  
lomaxtx, lomaxrx, lomaxfo, lomintx, lominrx, lominfo, LOeni)  
  If vb = 0 Then  
    vbl = True  
  Else  
    vbl = False  
  End If  
  If vbl = True Then  
    ptx = Data1.Recordset.Fields("Potencia")  
    prx = Data2.Recordset.Fields("Potencia")  
    ilong = Data1.Recordset.Fields("Incremento_LOnda")  
    ttx = Data1.Recordset.Fields("Tiempo_Respuesta")  
    trx = Data2.Recordset.Fields("Tiempo_Respuesta")  
    Ctx = Data1.Recordset.Fields("Precio")  
    Crx = Data2.Recordset.Fields("Precio")  
    cadc = Data1.Recordset.Fields("Precio_CtoAdicional")  
    campf = Data2.Recordset.Fields("Precio_CtoAmplificador")  
    ccon = Data3.Recordset.Fields("Precio/metro")  
    afib = Data3.Recordset.Fields("Atenuacion")  
    disp = Data3.Recordset.Fields("Dispersion")  
    vel = SeleccionarEnlace.Velocidad.Text  
    lenl = SeleccionarEnlace.Distancia.Text  
    MSeg = SeleccionarEnlace.MargS.Text  
    ccon = SeleccionarEnlace.Precio.Text  
    acon = SeleccionarEnlace.Perdidas.Text  
    NRegt = 0  
    If SeleccionarEnlace.NRZ.Value = True Then  
      cod = False  
    Else  
      cod = True  
    End If  
    balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)  
    If balp = False Then  
      NRegt = NRegt + 1  
      balpp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)  
    End If  
    balt = balttime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)  
    If balt = False Then  
      NRegt = NRegt + 1  
      balp = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)  
      If balp = False Then  
        NRegt = NRegt + 1  
        balpt = balpot(ptx, prx, afib, acon, MSeg, lenl, NRegt)  
      End If  
      baltt = balttime(disp, ilong, lenl, ttx, trx, vel, cod, NRegt)  
    End If  
    presup = coste(Ctx, cadc, Crx, campf, ccon, lenl, cfib, NRegt)  
    If presup < presupf Then  
      presupf = presup  
      resulR(0) = vbCrLf & _  
        "EL enlace óptimo es" & vbCrLf & _  
        Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo") & "-" &
```



```
Data2.Recordset.Fields("Modelo") & _
    vbCrLf & _
    "con un coste de " & Redondeo(presupf) & "€" & vbCrLf & _
    "requiriendo la solución " & NRegt & " nodos intermedios"
End If
m = m + 1
TmatCuad = (disp * ilong * (lenl / (1 + NRegt))) ^ 2
TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
Tres = Sqr(TresCuad)
If cod = False Then
    cdg = 0.7
    codigo = "NRZ"
Else
    cdg = 0.35
    codigo = "RZ"
End If
restvel = Abs(vel - (cdg / (0.001 * Tres)))
resul(m) = vbCrLf & _
    "CARACTERÍSTICAS DEL ENLACE:" & vbCrLf & _
    vbCrLf & _
    "Enlace punto a punto a l=" & LOenl & "nm" & vbCrLf & _
    "Distancia a enlazar " & lenl & "Kms" & vbCrLf & _
    "Velocidad de transmisión " & vel & "Mb/s" & " código " & codigo & vbCrLf & _
    "Número de repetidores optoelectrónicos " & NRegt & vbCrLf & _
    "Los conectores empleados tienen unas pérdidas de " & acon & "dB" & vbCrLf & _
    "y el Margen de Seguridad utilizado es de " & MSeg & "dB" & vbCrLf & _
    vbCrLf & _
    "RESULTADOS DEL ANÁLISIS:" & vbCrLf & _
    vbCrLf & _
    "Balance de Potencia" & vbCrLf & _
    "Pot_Tx >= Pot_Rx + 2 At_Conector + At_Fibra x Long_Max + MargSeg " & vbCrLf
& _
    ptx & " >= " & prx & " + 2 " & acon & " + " & afib & " x " & "BalPotencia.Lmax" & " + "
& MSeg & vbCrLf & _
    vbCrLf & _
    "Balance de Tiempos" & vbCrLf & _
    "Tfo^2 = Tmat^2 = (Disp * ILong * (Lenl / (1 + NRegt))) ^ 2" & vbCrLf & _
    Redondeo(TmatCuad) & " = (" & disp & " * " & ilong & " * (" & lenl & " / (1 + " &
NRegt & " )) ^ 2" & vbCrLf & _
    "Tresp^2 = Ttx^2 + Trx^2 + Tfo^2" & vbCrLf & _
    Redondeo(TresCuad) & " = " & Redondeo((ttx ^ 2)) & " + " & Redondeo((trx ^ 2)) & "
+ " & Redondeo(TmatCuad) & vbCrLf & _
    "NRZ => Tresp < 0.7/Velocidad RZ => Tresp < 0.35/Velocidad" & vbCrLf & _
    codigo & " => " & Redondeo(Sqr(TresCuad)) & " < " & cdg & "/0.001*" & vel & " = "
& Redondeo(cdg / (vel * 0.001)) & vbCrLf & _
    vbCrLf
    resul(R(m)) = "Presupuesto para el Enlace de Fibra seleccionado " & Redondeo(presup)
& " €"
End If
If vbl = False Then
```



```
Select Case vb
Case 1
  causa = "del tipo de conector usado para la fibra óptica"
Case 2
  causa = "del tipo de conector usado para el transmisor"
Case 3
  causa = "del tipo de conector usado para el receptor"
Case 4
  causa = "de la longitud de onda del transmisor"
Case 5
  causa = "de la longitud de onda del receptor"
Case 6
  causa = "de la longitud de onda de la fibra óptica"
End Select
  m = m + 1
  resul(m) = vbCrLf & _
    vbCrLf & _
    "El enlace de fibra óptica no es viable debido a una incompatibilidad" & causa
  'resulnode(m) = resul(m)
End If
  Set resulnode = Resultados.TreeView1.Nodes.Add(m)
  resulnode.Text = Data1.Recordset.Fields("Modelo") & "-" & Data3.Recordset.Fields("Modelo")
  & "-" & Data2.Recordset.Fields("Modelo")
  End If
  Data3.Recordset.MoveNext
  Loop
  End If
  End If
  Data2.Recordset.MoveNext
  Loop
  End If
  End If
  Data1.Recordset.MoveNext
  Loop
  End If
  Next K
  Next j
  Next i
  Resultados.Show
  'Resultados.Show vbModal, Me
End Sub

Private Sub Command5_Click()
  If Not Existe(App.path & Tipo2) Then MsgBox "No se ha localizado el fichero .: " &
  App.path & Tipo2, vbCritical: Exit Sub
  SeleccionarFibras.Adodc1.ConnectionString = "File Name=" & App.path & Tipo2 & ";"
  SeleccionarFibras.Adodc1.RecordSource = "select * from fibras order by Modelo"
  SeleccionarFibras.Adodc1.Refresh
  SeleccionarFibras.Show vbModal, Me
End Sub
```



```
Private Sub GuardarComo_Click()
Dim path As String, Index As Variant
Dim puntero
Dim i As Integer, j As Integer, K As Integer
GuardarDEN.TipoSeleccion = Index
GuardarDEN.Show vbModal
If Trim(GuardarDEN.Retorna) <> "" Then
    path = GuardarDEN.Retorna
End If
puntero = FreeFile
If path <> "" Then
Open path For Output As #puntero
Write #puntero, SeleccionarEnlace.LOnda.Text, SeleccionarEnlace.Velocidad.Text,
SeleccionarEnlace.Distancia.Text, _
SeleccionarEnlace.MargS.Text, SeleccionarEnlace.NRZ.Value,
SeleccionarEnlace.TipoConector.Text, _
SeleccionarEnlace.Perdidas.Text, SeleccionarEnlace.Precio.Text, _
List1.ListCount - 2, List2.ListCount - 2, List3.ListCount - 2
For i = 0 To List1.ListCount - 2
    Write #puntero, List1.List(i)
Next i
For j = 0 To List2.ListCount - 2
    Write #puntero, List2.List(j)
Next j
For K = 0 To List3.ListCount - 2
    Write #puntero, List3.List(K)
Next K
End If
Close #puntero
End Sub
```



## SeleccionarEnl.frm

```
Private Sub TipoConector_Change()  
    TipoConector = UCase(TipoConector)  
    TipoConector.SelStart = Len(TipoConector)  
End Sub  
  
Private Sub Volver_Click()  
Dim vel As String, lenl As String, MSeg As String, lond As String  
Dim ccon As String, acon As String  
Dim codigo As String  
vel = Velocidad.Text  
lenl = Distancia.Text  
MSeg = MargS.Text  
lond = LOnda.Text  
ccon = Precio.Text  
acon = Perdidas.Text  
If NRZ.Value = True Then  
cod = False  
codigo = "NRZ"  
Else  
cod = True  
codigo = "RZ"  
End If  
DiseñoEnl.Text1 = "Enlace punto a punto a una longitud de onda de " & LOnda.Text & "nm." _  
& " La distancia a enlazar es de " & Distancia.Text & "Kms, pudiéndose incluir repetidores" _  
& " optoelectrónicos en tramos intermedios en caso necesario." _  
& " La velocidad de transmisión es de " & Velocidad.Text & "Mbps." _  
& " La codificación empleada es " & codigo _  
& " Se usarán conectores con pérdidas de " & acon & "dB"  
SeleccionarEnlace.Hide  
End Sub  
  
Private Sub LOnda_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOnda) Then  
    Cancel = True  
    MsgBox "La longitud de onda debe ser un valor numérico"  
End If  
End Sub  
  
Private Sub MargS_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(MargS) Then  
    Cancel = True  
    MsgBox "El margen de seguridad debe ser un valor numérico"  
End If  
End Sub  
  
Private Sub Distancia_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Distancia) Then  
    Cancel = True  
    MsgBox "La distancia del enlace debe ser un valor numérico"  
End If  
End Sub
```

---



```
Private Sub Perdidas_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Perdidas) Then
    Cancel = True
    MsgBox "El margen de seguridad debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```

## **SeleccionarTx.frm**

Option Explicit

```
Private Sub InhabilitarBotones()
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is CommandButton Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```

```
Private Sub HabilitarBotones()
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is CommandButton Then
            Controls(n).Enabled = True
        End If
    Next n
End Sub
```

```
Private Sub InhabilitarCajas()
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```



```
Private Sub HabilitarCajas()
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is TextBox Then
            Controls(n).Enabled = True
        End If
    Next n
End Sub

Private Sub Eliminar_Click()
    Dim R As Integer
    R = List1.ListIndex
    List1.RemoveItem R
End Sub

Private Sub EliminarTodo_Click()
    List1.Clear
End Sub

Private Sub Inicio_Click()
    Adodc1.Recordset.MoveFirst
End Sub

Private Sub Anterior_Click()
    Adodc1.Recordset.MovePrevious
    If Adodc1.Recordset.BOF Then
        Adodc1.Recordset.MoveFirst
    End If
End Sub

Private Sub Seleccionar_Click()
    Dim i As Integer
    For i = 0 To List1.ListCount
        If (RTrim(Modelo.Text) = RTrim(List1.List(i))) Then
            GoTo fin
        End If
    Next i
    List1.AddItem Modelo.Text
fin:
End Sub

Private Sub SeleccionarTodo_Click()
    Adodc1.Recordset.MoveFirst
    While Adodc1.Recordset.EOF = False
        List1.AddItem Adodc1.Recordset.Fields("Modelo")
        Adodc1.Recordset.MoveNext
    Wend
End Sub

Private Sub Siguiente_Click()
    Adodc1.Recordset.MoveNext
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
End Sub

Private Sub Final_Click()
    Adodc1.Recordset.MoveLast
End Sub
```

---



```
Private Sub Buscar_Click()  
Dim Buscado As String, Criterio As String  
Buscado = InputBox("Introduzca el modelo del emisor que quiere buscar")  
If Buscado = "" Then Exit Sub  
Criterio = "Modelo Like "*" & Buscado & "*"  
Adodc1.Recordset.MoveNext  
If Not Adodc1.Recordset.EOF Then  
    Adodc1.Recordset.Find Criterio  
End If  
If Adodc1.Recordset.EOF Then  
    Adodc1.Recordset.MoveFirst  
    ' Buscar desde el principio  
    Adodc1.Recordset.Find Criterio  
If Adodc1.Recordset.EOF Then  
    Adodc1.Recordset.MoveLast  
    MsgBox ("No encuentro ese modelo")  
End If  
End If  
End Sub
```

```
Private Sub Volver_Click()  
Dim i As Integer  
    SeleccionTx.Hide  
    For i = 0 To List1.ListCount  
        DiseñoEnl.List1.AddItem List1.List(i)  
    Next i  
End Sub
```

```
Private Sub InclOnda_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(InclOnda) Then  
    Cancel = True  
    MsgBox "El incremento de la longitud de onda debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub LOndaMax_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOndaMax) Then  
    Cancel = True  
    MsgBox "La longitud de onda máxima debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub LOndaMin_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(LOndaMin) Then  
    Cancel = True  
    MsgBox "La longitud de onda mínima debe ser un valor numérico"  
End If  
End Sub
```

```
Private Sub Potencia_Validate(Cancel As Boolean)  
'Si el valor no es un valor numérico, mantener el foco  
If Not IsNumeric(Potencia) Then  
    Cancel = True  
    MsgBox "La potencia debe ser un valor numérico"  
End If  
End Sub
```

---



```
Private Sub PrecCirc_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(PrecCirc) Then
    Cancel = True
    MsgBox "El precio del circuito adicional debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TRespuesta_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(TRespuesta) Then
    Cancel = True
    MsgBox "El tiempo de respuesta debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```

## **SeleccionarRx.frm**

Option Explicit

```
Private Sub InhabilitarBotones()
Dim n As Integer
For n = 0 To Controls.Count - 1
    If TypeOf Controls(n) Is CommandButton Then
        Controls(n).Enabled = False
    End If
Next n
End Sub
```

```
Private Sub HabilitarBotones()
Dim n As Integer
For n = 0 To Controls.Count - 1
    If TypeOf Controls(n) Is CommandButton Then
        Controls(n).Enabled = True
    End If
Next n
End Sub
```



```
Private Sub InhabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = False  
        End If  
    Next n  
End Sub  
  
Private Sub HabilitarCajas()  
    Dim n As Integer  
    For n = 0 To Controls.Count - 1  
        If TypeOf Controls(n) Is TextBox Then  
            Controls(n).Enabled = True  
        End If  
    Next n  
End Sub  
  
Private Sub Eliminar_Click()  
    Dim R As Integer  
    R = List1.ListIndex  
    List1.RemoveItem R  
End Sub  
  
Private Sub EliminarTodo_Click()  
    List1.Clear  
End Sub  
  
Private Sub Inicio_Click()  
    Adodc1.Recordset.MoveFirst  
End Sub  
  
Private Sub Anterior_Click()  
    Adodc1.Recordset.MovePrevious  
    If Adodc1.Recordset.BOF Then  
        Adodc1.Recordset.MoveFirst  
    End If  
End Sub  
  
Private Sub Seleccionar_Click()  
    Dim i As Integer  
    For i = 0 To List1.ListCount  
        If (RTrim(Modelo.Text) = RTrim(List1.List(i))) Then  
            GoTo fin  
        End If  
    Next i  
    List1.AddItem Modelo.Text  
fin:  
End Sub  
  
Private Sub SeleccionarTodo_Click()  
    Adodc1.Recordset.MoveFirst  
    While Adodc1.Recordset.EOF = False  
        List1.AddItem Adodc1.Recordset.Fields("Modelo")  
        Adodc1.Recordset.MoveNext  
    Wend  
    'Adodc1.Recordset.MoveLast  
End Sub
```

---



```
Private Sub Siguiente_Click()  
    Adodc1.Recordset.MoveNext  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveLast  
    End If  
End Sub
```

```
Private Sub Final_Click()  
    Adodc1.Recordset.MoveLast  
End Sub
```

```
Private Sub Buscar_Click()  
    Dim Buscado As String, Criterio As String  
    Buscado = InputBox("Introduzca el modelo del receptor que quiere buscar")  
    If Buscado = "" Then Exit Sub  
    Criterio = "Modelo Like "*" & Buscado & "*"  
    ' Buscar desde el siguiente registro a la posición actual  
    Adodc1.Recordset.MoveNext  
    If Not Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.Find Criterio  
    End If  
    If Adodc1.Recordset.EOF Then  
        Adodc1.Recordset.MoveFirst  
        ' Buscar desde el principio  
        Adodc1.Recordset.Find Criterio  
        If Adodc1.Recordset.EOF Then  
            Adodc1.Recordset.MoveLast  
            MsgBox ("No encuentro ese modelo")  
        End If  
    End If  
End Sub
```

```
Private Sub Volver_Click()  
    Dim i As Integer  
    SeleccionarRx.Hide  
    For i = 0 To List1.ListCount  
        DiseñoEnl.List2.AddItem List1.List(i)  
    Next i  
End Sub
```

```
Private Sub LOndaMax_Validate(Cancel As Boolean)  
    'Si el valor no es un valor numérico, mantener el foco  
    If Not IsNumeric(LOndaMax) Then  
        Cancel = True  
        MsgBox "La longitud de onda máxima debe ser un valor numérico"  
    End If  
End Sub
```

```
Private Sub LOndaMin_Validate(Cancel As Boolean)  
    'Si el valor no es un valor numérico, mantener el foco  
    If Not IsNumeric(LOndaMin) Then  
        Cancel = True  
        MsgBox "La longitud de onda mínima debe ser un valor numérico"  
    End If  
End Sub
```



```
Private Sub Potencia_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Potencia) Then
    Cancel = True
    MsgBox "La potencia debe ser un valor numérico"
End If
End Sub
```

```
Private Sub PrecCirc_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(PrecCirc) Then
    Cancel = True
    MsgBox "El precio del circuito adicional debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub TRespuesta_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(TRespuesta) Then
    Cancel = True
    MsgBox "El tiempo de respuesta debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Velocidad_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Velocidad) Then
    Cancel = True
    MsgBox "La velocidad debe ser un valor numérico"
End If
End Sub
```

## **SeleccionarFibras.frm**

Option Explicit

```
Private Sub InhabilitarBotones()
    Dim n As Integer
    For n = 0 To Controls.Count - 1
        If TypeOf Controls(n) Is CommandButton Then
            Controls(n).Enabled = False
        End If
    Next n
End Sub
```



```
Private Sub HabilitarBotones()  
Dim n As Integer  
For n = 0 To Controls.Count - 1  
If TypeOf Controls(n) Is CommandButton Then  
Controls(n).Enabled = True  
End If  
Next n  
End Sub
```

```
Private Sub InhabilitarCajas()  
Dim n As Integer  
For n = 0 To Controls.Count - 1  
If TypeOf Controls(n) Is TextBox Then  
Controls(n).Enabled = False  
End If  
Next n  
End Sub
```

```
Private Sub HabilitarCajas()  
Dim n As Integer  
For n = 0 To Controls.Count - 1  
If TypeOf Controls(n) Is TextBox Then  
Controls(n).Enabled = True  
End If  
Next n  
End Sub
```

```
Private Sub Eliminar_Click()  
Dim R As Integer  
R = List1.ListIndex  
List1.RemoveItem R  
End Sub
```

```
Private Sub EliminarTodo_Click()  
List1.Clear  
End Sub
```

```
Private Sub Inicio_Click()  
Adodc1.Recordset.MoveFirst  
End Sub
```

```
Private Sub Anterior_Click()  
Adodc1.Recordset.MovePrevious  
If Adodc1.Recordset.BOF Then  
Adodc1.Recordset.MoveFirst  
End If  
End Sub
```

```
Private Sub Seleccionar_Click()  
Dim i As Integer  
For i = 0 To List1.ListCount  
If (RTrim(Modelo.Text) = RTrim(List1.List(i))) Then  
GoTo fin  
End If  
Next i  
List1.AddItem Modelo.Text
```

```
fin:  
End Sub
```

---



```
Private Sub SeleccionarTodo_Click()
    Adodc1.Recordset.MoveFirst
    While Adodc1.Recordset.EOF = False
        List1.AddItem Adodc1.Recordset.Fields("Modelo")
        Adodc1.Recordset.MoveNext
    Wend
End Sub

Private Sub Siguiente_Click()
    Adodc1.Recordset.MoveNext
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveLast
    End If
End Sub

Private Sub Final_Click()
    Adodc1.Recordset.MoveLast
End Sub

Private Sub Buscar_Click()
    Dim Buscado As String, Criterio As String
    Buscado = InputBox("Introduzca el modelo de la fibra óptica que quiere buscar")
    If Buscado = "" Then Exit Sub
    Criterio = "Modelo Like "*" & Buscado & "*""
    ' Buscar desde el siguiente registro a la posición actual
    Adodc1.Recordset.MoveNext
    If Not Adodc1.Recordset.EOF Then
        Adodc1.Recordset.Find Criterio
    End If
    If Adodc1.Recordset.EOF Then
        Adodc1.Recordset.MoveFirst
        ' Buscar desde el principio
        Adodc1.Recordset.Find Criterio
        If Adodc1.Recordset.EOF Then
            Adodc1.Recordset.MoveLast
            MsgBox ("No encuentro ese modelo")
        End If
    End If
End Sub

Private Sub Volver_Click()
    Dim i As Integer
    SeleccionarFibras.Hide
    For i = 0 To List1.ListCount
        DiseñoEnl.List3.AddItem List1.List(i)
    Next i
End Sub

Private Sub LOndaMax_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(LOndaMax) Then
        Cancel = True
        MsgBox "La longitud de onda máxima debe ser un valor numérico"
    End If
End Sub
```



```
Private Sub LOndaMin_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOndaMin) Then
    Cancel = True
    MsgBox "La longitud de onda mínima debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Precio_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Precio) Then
    Cancel = True
    MsgBox "El precio debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Atenuacion_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Atenuacion) Then
    Cancel = True
    MsgBox "La atenuación debe ser un valor numérico"
End If
End Sub
```

```
Private Sub Dispersion_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(Dispersion) Then
    Cancel = True
    MsgBox "La dispersión debe ser un valor numérico"
End If
End Sub
```

## **GuardarDEN.frm**

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
Dim extension As String
extension = ".den"
If Right(Dir1.path, 1) = "\" Then
    Ret = Dir1.path & Text1.Text & extension
    If Tipo = 1 Then Ret = Dir1.path
Else
    Ret = Dir1.path & "\" & Text1.Text & extension
    If Tipo = 1 Then Ret = Dir1.path
End If
Unload Me
End Sub
```

---



```
Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub

Private Sub File1_Click()
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Load()
    If App.PreviousInstance Then Ret = "": Unload Me
    Ret = ""
    Dir1.path = Drive1.Drive
    File1.path = Dir1.path
    Ret = Dir1.path & Text1.Text
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Retorna = Ret
End Sub

Public Property Get TipoSeleccion() As Variant
    TipoSeleccion = Tipo
End Property

Public Property Let TipoSeleccion(ByVal vNewValue As Variant)
    Tipo = vNewValue
End Property
```

## **Resultados.frm**

```
Private Sub TreeView1_NodeClick(ByVal Node As Node)
    Text1.Text = resul(Node.Index - 1)
    Text2.Text = resulR(Node.Index - 1)
End Sub
```



```
Private Sub Imprimir_Click()
    Dim Cabecera As String
    Dim Cuerpo As String
    Dim K As Integer
    Screen.MousePointer = vbHourglass
    Printer.Print vbCrLf & vbCrLf & vbCrLf
    Printer.Font.Size = 14
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf
    For K = 2 To TreeView1.Nodes.Count
        Printer.Font.Size = 10
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf
        Printer.Font.Size = 8
        Printer.Print resulV(K - 1)
    Next
    Printer.EndDoc
    Screen.MousePointer = vbNormal
End Sub
```

### **Tsubida.frm**

```
Dim resulnodeTs As Node
Dim m As Integer
```

```
Private Sub Form_Unload(Cancel As Integer)
    Unload ResultadosTs
End Sub
```

```
Private Sub Form_Load()
    m = 0
    Set resulnodeTs = ResultadosTs.TreeView1.Nodes.Add()
    resulnodeTs.Text = "TIEMPO DE RESPUESTA DE LA FIBRA OPTICA"
End Sub
```

```
Private Sub AnchEsp_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(AnchEsp) Then
        Cancel = True
        MsgBox "La anchura espectral del generador de luz debe ser un valor numérico"
    End If
End Sub
```

```
Private Sub AnchMod_Validate(Cancel As Boolean)
    'Si el valor no es un valor numérico, mantener el foco
    If Not IsNumeric(AnchMod) Then
        Cancel = True
        MsgBox "El ancho de banda modal de la fibra óptica debe ser un valor numérico"
    End If
End Sub
```



```
Private Sub Calcular_Click()
Dim Tfo As Single, Tcrom As Single, Wesp As Single
Wesp = AnchEsp.Text
If LOnda.Text = "850" Then
Dim Tmod As Single, Dcrom As Single
Dim Bmodelec As Single, Bmodinst As Single, Bmod As Single
Bmod = AnchMod.Text
Dcrom = DispCrom.Text
Bmodelect = 0.71 * Bmod
Tmod = 0.35 / Bmodelect
Tcrom = Dcrom * Wesp
Tfo = Sqr((Tmod ^ 2) + (Tcrom ^ 2))
Else
Dim Dmat As Single, Dgo As Single, Dcromat As Single
Dmat = DispMat.Text
Dgo = DispGuia.Text
Dcromat = Dmat + Dgo
Tcrom = Dcromat * Wesp
Tfo = Tcrom
End If
m = m + 1
Set resulnodeTs = ResultadosTs.TreeView1.Nodes.Add(m)
resulnodeTs.Text = Modelo.Text
If LOnda.Text = "850" Then
resulTs(m) = "FIBRAS MULTIMODO" & vbCrLf & _
"Bmodalelect = 0.71 * Bmodal" & vbCrLf & _
"Bmodalelect = 0.71*" & Bmod & " = " & Redondeo((0.71 * Bmod)) & vbCrLf & _
"Tmodal = 0.35 / Bmodalelect" & vbCrLf & _
"Tmodal = 0.35/" & Bmodelect & " = " & Redondeo((0.35 / Bmodelect)) & vbCrLf & _
"Tcromaticat = Dispromaticat * Wesp" & vbCrLf & _
"Tcromaticat = " & Dcrom & "*" & Wesp & " = " & Redondeo((Dcrom * Wesp)) & vbCrLf
& _
"Tfibra^2 = Tmodal^2 + Tcromaticat^2" & vbCrLf & _
Redondeo(Tfo ^ 2) & " = " & Redondeo(Tmod ^ 2) & " + " & Redondeo(Tcrom ^ 2)
resulTsR(m) = "Tfibra = " & Redondeo(Tfo) & "ns"
Else
resulTs(m) = "FIBRAS MONOMODO" & vbCrLf & _
"Dispromaticat = Dispmat + Dispguia" & vbCrLf & _
"Dispromaticat = " & Dmat & " + " & Dgo & " = " & (Dmat + Dgo) & vbCrLf & _
"Tcromaticat = Dispromaticat * Wesp" & vbCrLf & _
"Tcromaticat = " & Dcromat & "*" & Wesp & " = " & Redondeo((Dcromat * Wesp)) &
vbCrLf & _
"Tfibra^2 = Tcromaticat^2" & vbCrLf & _
"Tfibra^2 = " & Redondeo(Tcrom ^ 2)
resulTsR(m) = "Tfibra = " & Redondeo(Tfo) & "ns"
End If
ResultadosTs.Show
End Sub

Private Sub Command1_Click()
```



```
If LOnda.Text = "850" Then
  DispMat.Text = ""
  DispGuia.Text = ""
  DispMat.Enabled = False
  DispGuia.Enabled = False
  DispCrom.Enabled = True
  AnchMod.Enabled = True
Else
  DispCrom.Text = ""
  AnchMod = ""
  DispCrom.Enabled = False
  AnchMod.Enabled = False
  DispMat.Enabled = True
  DispGuia.Enabled = True
End If
End Sub
```

```
Private Sub DispCrom_Validate(Cancel As Boolean)
If Not IsNumeric(DispCrom) Then
  Cancel = True
  MsgBox "La dispersión cromática de la fibra debe ser un valor numérico"
End If
End Sub
```

```
Private Sub DispGuia_Validate(Cancel As Boolean)
If Not IsNumeric(DispGuia) Then
  Cancel = True
  MsgBox "La dispersión cromática guía-onda debe ser un valor numérico"
End If
End Sub
```

```
Private Sub DispMat_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(DispMat) Then
  Cancel = True
  MsgBox "La dispersión cromática material debe ser un valor numérico"
End If
End Sub
```

```
Private Sub GuardarComo_Click()
Dim path As String
Dim puntero
GuardarTSB.TipoSeleccion = Index
GuardarTSB.Show vbModal
If Trim(GuardarTSB.Retorna) <> "" Then
  path = GuardarTSB.Retorna
End If
puntero = FreeFile
If path <> "" Then
Open path For Output As #puntero
Write #puntero, LOnda.Text, Modelo.Text, DispCrom.Text, DispMat.Text, DispGuia.Text, _
Distancia.Text, AnchEsp.Text, AnchMod.Text
End If
Close #puntero
End Sub
```

---



```
Private Sub LOnda_Validate(Cancel As Boolean)
'Si el valor no es un valor numérico, mantener el foco
If Not IsNumeric(LOnda) Then
    Cancel = True
    MsgBox "La longitud de onda debe ser un valor numérico"
End If
End Sub
```

## **GuardarTSB.frm**

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property
Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
Dim extension As String
extension = ".tsb"
If Right(Dir1.path, 1) = "\" Then
    Ret = Dir1.path & Text1.Text & extension
    If Tipo = 1 Then Ret = Dir1.path
Else
    Ret = Dir1.path & "\" & Text1.Text & extension
    If Tipo = 1 Then Ret = Dir1.path
End If
Unload Me
End Sub

Private Sub Command2_Click()
Ret = ""
Unload Me
End Sub

Private Sub Dir1_Change()
File1.path = Dir1.path
If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
Drive1.Refresh
Dir1.Refresh
Dir1.path = Drive1.Drive
End Sub

Private Sub File1_Click()
Ret = Dir1.path & Text1.Text
End Sub
```



```
Private Sub Form_Load()  
    If App.PreviousInstance Then Ret = "": Unload Me  
    Ret = ""  
    Dir1.path = Drive1.Drive  
    File1.path = Dir1.path  
    Ret = Dir1.path & Text1.Text  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    Retorna = Ret  
End Sub  
  
Public Property Get TipoSeleccion() As Variant  
    TipoSeleccion = Tipo  
End Property  
  
Public Property Let TipoSeleccion(ByVal vNewValue As Variant)  
    Tipo = vNewValue  
End Property
```

## ResultadosTs.frm

```
Private Sub Imprimir_Click()  
    Dim Cabecera As String  
    Dim Cuerpo As String  
    Dim K As Integer  
    Screen.MousePointer = vbHourglass  
    Printer.Print vbCrLf & vbCrLf & vbCrLf  
    Printer.Font.Size = 14  
    Printer.Print TreeView1.Nodes(1).Text & vbCrLf  
    For K = 2 To TreeView1.Nodes.Count  
        Printer.Font.Size = 10  
        Printer.Print UCase(TreeView1.Nodes(K).Text) & vbCrLf  
        Printer.Font.Size = 8  
        Printer.Print resulV(K - 1)  
    Next  
    Printer.EndDoc  
    Screen.MousePointer = vbNormal  
End Sub  
  
Private Sub TreeView1_NodeClick(ByVal Node As Node)  
    Text1.Text = resulTs(Node.Index - 1)  
    Text2.Text = resulTsR(Node.Index - 1)  
End Sub
```



## Abrir.frm

```
Option Explicit
Dim Ret
Dim Tipo

Public Property Get Retorna() As Variant
    Retorna = Ret
End Property

Public Property Let Retorna(ByVal vNewValue As Variant)
    Ret = Retorna
End Property

Private Sub Command1_Click()
    If Right(Dir1.path, 1) = "\" Then
        Ret = Dir1.path & File1.FileName
        If Tipo = 1 Then Ret = Dir1.path
    Else
        Ret = Dir1.path & "\" & File1.FileName
        If Tipo = 1 Then Ret = Dir1.path
    End If
    Unload Me
End Sub

Private Sub Command2_Click()
    Ret = ""
    Unload Me
End Sub

Private Sub Dir1_Change()
    File1.path = Dir1.path
    If File1.ListCount > 0 Then File1.ListIndex = 0
End Sub

Private Sub Drive1_Change()
    Drive1.Refresh
    Dir1.Refresh
    Dir1.path = Drive1.Drive
End Sub

Private Sub Form_Load()
    If App.PreviousInstance Then Ret = "": Unload Me
    Ret = ""
    Dir1.path = Drive1.Drive
    File1.path = Dir1.path
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Retorna = Ret
End Sub

Public Property Get TipoSeleccion() As Variant
    TipoSeleccion = Tipo
End Property
```

---



```
Public Property Let TipoSeleccion(ByVal vNewValue As Variant)
    Tipo = vNewValue
End Property
```

## **Buscar.frm**

```
Option Explicit
Dim R As Adodc
Dim Nom As Variant
Dim TblNombres, TblCampos
Dim K As Integer
```

```
Private Sub CmAceptar_Click()
    Buscamos
    Unload Me
End Sub
```

```
Private Sub CmCancelar_Click()
    Unload Me
End Sub
```

```
Public Property Get Rs() As Adodc
    Set Rs = R
End Property
```

```
Public Property Let Rs(ByVal vNewValue As Adodc)
    Set R = vNewValue
End Property
```

```
Public Property Let Nombres(ByVal vNewValue As Variant)
    Nom = vNewValue
    If Trim(Nom) <> "" Then
        TblNombres = Split(Nom, ";")
        CbBuscar.Clear
        For K = 0 To UBound(TblNombres)
            CbBuscar.AddItem TblNombres(K)
            CbBuscar.ItemData(CbBuscar.NewIndex) = K
        Next
        CbBuscar.ListIndex = 0
    End If
End Property
```

```
Private Sub Buscamos()
    Dim RsTemp As Recordset
    If Not R.Recordset.EOF And Not R.Recordset.BOF Then
        R.Recordset.MoveFirst
        If R.Recordset.Fields(TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex))).Type = 202
Then
            R.Recordset.Find TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex)) & " = " &
TxtBuscar & ""
        Else
            If R.Recordset.Fields(TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex))).Type = 7
Then
```



```
R.Recordset.Find TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex)) & " = #" &
TxtBuscar.Text & "#"
Else
R.Recordset.Find TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex)) & " = " &
TxtBuscar.Text & ""
End If
End If
If R.Recordset.EOF Then
MsgBox " Valor no encontrado"
Else
If R.Recordset.Fields(TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex))).Type = 202
Then
R.Recordset.Filter = TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex)) & " = "" &
TxtBuscar & ""
Else
If R.Recordset.Fields(TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex))).Type = 7
Then
R.Recordset.Filter = TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex)) & " = #"
& TxtBuscar & "#"
Else
R.Recordset.Filter = TblCampos(CbBuscar.ItemData(CbBuscar.ListIndex)) & " = "
& TxtBuscar
End If
End If
End If
End If
End Sub

Public Property Let Campos(ByVal vNewValue As Variant)
TblCampos = Split(vNewValue, ";")
End Property
```

## Menus.frm

```
Private Sub FicheroAbrir_Click()
Dim path As String, var As String
Dim puntero
Dim i As Integer, j As Integer, K As Integer
Dim Tx(10) As String, Rx(10) As String, fo(10) As String
Dim var1 As String, var2 As String, var3 As String, var4 As String, var5 As String
Dim var6 As String, var7 As String, var8 As String, var9 As String, var10 As String
Dim var11 As String, var12 As String, var13 As String, var14 As String, var15 As String
Dim var16 As String, var17 As String, var18 As String
Abrir.TipoSeleccion = Index
Abrir.Show vbModal
If Trim(Abrir.Retorna) <> "" Then
path = Abrir.Retorna
End If
puntero = FreeFile
Open path For Input As #puntero
var = Right(path, 3)
```



```
If var = "bpt" Then
  Input #puntero, var1, var2, var3, var4, var5, var6, var7, var8, var9
  BalancePotencia.LOnda.Text = var1
  BalancePotencia.Distancia.Text = var2
  BalancePotencia.NRg.Text = var3
  BalancePotencia.MargS.Text = var4
  BalancePotencia.TipoConector.Text = var5
  BalancePotencia.Perdidas.Text = var6
  BalancePotencia.Combo4.Text = var7
  BalancePotencia.Combo5.Text = var8
  BalancePotencia.Combo6.Text = var9
  Close #puntero
  BalancePotencia.Show
Elseif var = "btp" Then
  Input #puntero, var1, var2, var3, var4, var5, var6, var7, var8, var9
  BalanceTiempo.LOnda.Text = var1
  BalanceTiempo.Velocidad.Text = var2
  BalanceTiempo.Distancia.Text = var3
  BalanceTiempo.NRg.Text = var4
  BalanceTiempo.TipoConector.Text = var5
  BalanceTiempo.NRZ.Value = var6
  BalanceTiempo.Combo4.Text = var7
  BalanceTiempo.Combo2.Text = var8
  BalanceTiempo.Combo3.Text = var9
  Close #puntero
  BalanceTiempo.Show
Elseif var = "cop" Then
  Input #puntero, var1, var2, var3, var4, var5, var6, var7, var8, var9, _
  var10, var11, var12, var13, var14, var15, var16, var17, var18
  ComponenteOptimo.LOnda.Text = var1
  ComponenteOptimo.Velocidad.Text = var2
  ComponenteOptimo.Distancia.Text = var3
  ComponenteOptimo.NRg.Text = var4
  ComponenteOptimo.MargS.Text = var5
  ComponenteOptimo.TipoConector.Text = var6
  ComponenteOptimo.Perdidas.Text = var7
  ComponenteOptimo.Precio.Text = var8
  ComponenteOptimo.NRZ.Value = var9
  ComponenteOptimo.CheckTx.Value = var10
  ComponenteOptimo.CheckRx.Value = var11
  ComponenteOptimo.CheckFO.Value = var12
  ComponenteOptimo.Combo1.Text = var13
  ComponenteOptimo.Combo2.Text = var14
  ComponenteOptimo.Combo3.Text = var15
  ComponenteOptimo.costenl.Value = var16
  ComponenteOptimo.bpot.Value = var17
  ComponenteOptimo.btime.Value = var18
  Close #puntero
  ComponenteOptimo.Show
```



```
Elsif var = "den" Then
  Input #puntero, var1, var2, var3, var4, var5, var6, var7, var8, var9, var10, var11
  SeleccionarEnlace.LOnda.Text = var1
  SeleccionarEnlace.Velocidad.Text = var2
  SeleccionarEnlace.Distancia.Text = var3
  SeleccionarEnlace.MargS.Text = var4
  SeleccionarEnlace.NRZ.Value = var5
  SeleccionarEnlace.TipoConector.Text = var6
  SeleccionarEnlace.Perdidas.Text = var7
  SeleccionarEnlace.Precio.Text = var8
  For i = 0 To var9
    Input #puntero, Tx(i + 1)
  Next i
  For j = 0 To var10
    Input #puntero, Rx(j + 1)
  Next j
  For K = 0 To var11
    Input #puntero, fo(K + 1)
  Next K
  Close #puntero
  If SeleccionarEnlace.NRZ.Value = True Then
    cod = False
    var12 = "NRZ"
  Else
    cod = True
    var12 = "RZ"
  End If
  DiseñoEnl.Text1 = "Enlace punto a punto a una longitud de onda de " &
  SeleccionarEnlace.LOnda.Text & "nm." _
  & " La distancia a enlazar es de " & SeleccionarEnlace.Distancia.Text _
  & "Kms, pudiéndose incluir repetidores optoelectrónicos en tramos intermedios en caso
necesario." _
  & " La velocidad de transmisión es de " & SeleccionarEnlace.Velocidad.Text & "Mbps." _
  & " La codificación empleada es " & var12 _
  & " Se usarán conectores con pérdidas de " & SeleccionarEnlace.Perdidas.Text & "dB"
  For i = 0 To var9 + 1
    DiseñoEnl.List1.AddItem Tx(i + 1)
    SeleccionTx.List1.AddItem Tx(i + 1)
  Next i
  For j = 0 To var10 + 1
    DiseñoEnl.List2.AddItem Rx(j + 1)
    SeleccionRx.List1.AddItem Rx(j + 1)
  Next j
  For K = 0 To var11 + 1
    DiseñoEnl.List3.AddItem fo(K + 1)
    SeleccionFibras.List1.AddItem fo(K + 1)
  Next K
  DiseñoEnl.Show
```



```
Elself var = "en!" Then
  Input #puntero, var1, var2, var3, var4, var5, var6, var7, var8, _
  var9, var10, var11, var12
  Enlace.LOnda.Text = var1
  Enlace.Velocidad.Text = var2
  Enlace.Distancia.Text = var3
  Enlace.NRg.Text = var4
  Enlace.MargS.Text = var5
  Enlace.TipoConector.Text = var6
  Enlace.Perdidas.Text = var7
  Enlace.Precio.Text = var8
  Enlace.NRZ.Value = var9
  Enlace.Combo1.Text = var10
  Enlace.Combo2.Text = var11
  Enlace.Combo3.Text = var12
  Close #puntero
  Enlace.Show
Elself var = "tsb" Then
  Input #puntero, var1, var2, var3, var4, var5, var6, var7
  Tsubida.LOnda.Text = var1
  Tsubida.Modelo.Text = var2
  Tsubida.DispCrom.Text = var3
  Tsubida.DispMat.Text = var4
  Tsubida.DispGuia.Text = var5
  Tsubida.AnchEsp.Text = var6
  Tsubida.AnchMod.Text = var7
  Close #puntero
  Tsubida.Show
Elself var = "vbl" Then
  Input #puntero, var1, var2, var3, var4, var5
  Viabilidad.LOnda.Text = var1
  Viabilidad.TipoConector.Text = var2
  Viabilidad.Combo1.Text = var3
  Viabilidad.Combo2.Text = var4
  Viabilidad.Combo3.Text = var5
  Close #puntero
  Viabilidad.Show
End If
End Sub

Private Sub FicheroSalir_Click()
  Unload Menu
  Unload BDEmisores
  Unload BDRreceptores
  Unload BDFibras
  Unload Viabilidad
  Unload ResultadosV
  Unload BalancePotencia
  Unload ResultadosBP
  Unload BalanceTiempo
  Unload ResultadosBT
  Unload Enlace
  Unload ResultadosEnl
```



```
Unload ComponenteOptimo
Unload ResultadosCOpt
Unload DiseñoEnl
Unload Resultados
Unload Tsubida
Unload ResultadosTs
Unload SeleccionarEnlace
Unload SeleccionarFibras
Unload SeleccionarRx
Unload SeleccionTx
End Sub

Private Sub Form_Load()
    Me.Top = 0
    Me.Left = 0
End Sub

'Private Sub Form_Resize()
' If Me.WindowState <> vbMaximized Then Me.WindowState = vbMaximized
'End Sub

Private Sub mnuBDE_Click()
    BDEmisores.Show vbModal, Me
End Sub

Private Sub mnuBDR_Click()
    BDRreceptores.Show vbModal, Me
End Sub

Private Sub mnuBDF_Click()
    BDFibras.Show vbModal, Me
End Sub

Private Sub mnuTAv_Click()
    Viabilidad.Show
End Sub

Private Sub mnuTABp_Click()
    BalancePotencia.Show
End Sub

Private Sub mnuTABt_Click()
    BalanceTiempo.Show
End Sub

Private Sub mnuTAp_Click()
    Enlace.Show
End Sub

Private Sub mnuDEeo_Click()
    ComponenteOptimo.Show
End Sub

Private Sub mnuDEco_Click()
    DiseñoEnl.Show
End Sub
```

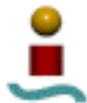
---



```
Private Sub mnuTSco_Click()  
    Tsubida.Show  
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)  
    Select Case Button.Index  
        Case 1  
            If Not Existe(App.path & Tipo0) Then MsgBox "No se ha localizado el fichero .:" &  
App.path & Tipo0, vbCritical: Exit Sub  
            'select * from Emisores order by Modelo  
            BDEmisores.Adodc1.ConnectionString = "File Name=" & App.path & Tipo0 & ";"  
            BDEmisores.Adodc1.RecordSource = "select * from Emisores order by Modelo"  
            BDEmisores.Adodc1.Refresh  
            BDEmisores.Show vbModal, Me  
        Case 2  
            If Not Existe(App.path & Tipo1) Then MsgBox "No se ha localizado el fichero .:" &  
App.path & Tipo1, vbCritical: Exit Sub  
            BDRreceptores.Adodc1.ConnectionString = "File Name=" & App.path & Tipo1 & ";"  
            BDRreceptores.Adodc1.RecordSource = "select * from receptores order by Modelo"  
            BDRreceptores.Adodc1.Refresh  
            BDRreceptores.Show vbModal, Me  
        Case 3  
            If Not Existe(App.path & Tipo2) Then MsgBox "No se ha localizado el fichero .:" &  
App.path & Tipo2, vbCritical: Exit Sub  
            BDFibras.Adodc1.ConnectionString = "File Name=" & App.path & Tipo2 & ";"  
            BDFibras.Adodc1.RecordSource = "select * from fibras order by Modelo"  
            BDFibras.Adodc1.Refresh  
            BDFibras.Show vbModal, Me  
        Case 4  
            If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\emisores.mdb", vbCritical: Exit Sub  
            If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\receptores.mdb", vbCritical: Exit Sub  
            If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\fibras.mdb", vbCritical: Exit Sub  
            Viabilidad.Data1.DatabaseName = App.path & "\emisores.mdb"  
            Viabilidad.Data2.DatabaseName = App.path & "\receptores.mdb"  
            Viabilidad.Data3.DatabaseName = App.path & "\fibras.mdb"  
            Viabilidad.Show  
            Viabilidad.Data1.Refresh  
            Viabilidad.Data2.Refresh  
            Viabilidad.Data3.Refresh  
        Case 5  
            If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\emisores.mdb", vbCritical: Exit Sub  
            If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\receptores.mdb", vbCritical: Exit Sub  
            If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" &  
App.path & "\fibras.mdb", vbCritical: Exit Sub  
            BalancePotencia.Data1.DatabaseName = App.path & "\emisores.mdb"  
            BalancePotencia.Data2.DatabaseName = App.path & "\receptores.mdb"  
            BalancePotencia.Data3.DatabaseName = App.path & "\fibras.mdb"  
            BalancePotencia.Show  
            BalancePotencia.Data1.Refresh  
            BalancePotencia.Data2.Refresh  
            BalancePotencia.Data3.Refresh
```

---



Case 6

```
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\fibras.mdb", vbCritical: Exit Sub
BalanceTiempo.Data1.DatabaseName = App.path & "\emisores.mdb"
BalanceTiempo.Data2.DatabaseName = App.path & "\receptores.mdb"
BalanceTiempo.Data3.DatabaseName = App.path & "\fibras.mdb"
BalanceTiempo.Show
BalanceTiempo.Data1.Refresh
BalanceTiempo.Data2.Refresh
BalanceTiempo.Data3.Refresh
```

Case 7

```
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\fibras.mdb", vbCritical: Exit Sub
Enlace.Data1.DatabaseName = App.path & "\emisores.mdb"
Enlace.Data2.DatabaseName = App.path & "\receptores.mdb"
Enlace.Data3.DatabaseName = App.path & "\fibras.mdb"
Enlace.Show
Enlace.Data1.Refresh
Enlace.Data2.Refresh
Enlace.Data3.Refresh
```

Case 9

```
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\fibras.mdb", vbCritical: Exit Sub
DiseñoEnl.Data1.DatabaseName = App.path & "\emisores.mdb"
DiseñoEnl.Data2.DatabaseName = App.path & "\receptores.mdb"
DiseñoEnl.Data3.DatabaseName = App.path & "\fibras.mdb"
DiseñoEnl.Show
DiseñoEnl.Data1.Refresh
DiseñoEnl.Data2.Refresh
DiseñoEnl.Data3.Refresh
```

Case 10

```
If Not Existe(App.path & "\emisores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\emisores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\receptores.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\receptores.mdb", vbCritical: Exit Sub
If Not Existe(App.path & "\fibras.mdb") Then MsgBox "No se ha localizado el fichero .:" & App.path & "\fibras.mdb", vbCritical: Exit Sub
ComponenteOptimo.Data1.DatabaseName = App.path & "\emisores.mdb"
ComponenteOptimo.Data2.DatabaseName = App.path & "\receptores.mdb"
ComponenteOptimo.Data3.DatabaseName = App.path & "\fibras.mdb"
ComponenteOptimo.Show
ComponenteOptimo.Data1.Refresh
ComponenteOptimo.Data2.Refresh
ComponenteOptimo.Data3.Refresh
```



```
Case 11
  Tsubida.Show
  End Select
End Sub
```

```
Private Sub TreeView1_NodeClick(ByVal Node As Node)
Text1.Text = resul(Node.Index - 1)
End Sub
```

## Compatibilidad.frm

Option Explicit

```
Public Function viab(Ctipo, Ctx, Crx, Cfo, _
  LOMatx, LOMarx, LOMafo, LOMitx, LOMirx, LOMifo, LOenl) As Integer
Dim patron1 As Boolean, patron2 As Boolean, patron3 As Boolean
Dim fo1 As Boolean, fo2 As Boolean, fo3 As Boolean, fo As Boolean
Dim txfo1 As Boolean, txfo2 As Boolean, txfo3 As Boolean, txfo As Boolean
Dim rxfo1 As Boolean, rxfo2 As Boolean, rxfo3 As Boolean, rxfo As Boolean
Dim ltx As Boolean, lrx As Boolean, lfo As Boolean
Dim vbl As Integer
If Ctipo Like "**ST*" = True Then
  patron1 = True
Else
  patron1 = False
End If
If Ctipo Like "**SC*" = True Then
  patron2 = True
Else
  patron2 = False
End If
If Ctipo Like "**FC*" = True Then
  patron3 = True
Else
  patron3 = False
End If
If patron1 = True Then
  fo1 = Cfo Like "**ST*"
Else
  fo1 = False
End If
If patron2 = True Then
  fo2 = Cfo Like "**SC*"
Else
  fo2 = False
End If
If patron3 = True Then
  fo3 = Cfo Like "**FC*"
Else
  fo3 = False
End If
```

|

---



```
f patron1 = True Then
txfo1 = Ctx Like "**ST*"
Else
txfo1 = False
End If
If patron2 = True Then
txfo2 = Ctx Like "**SC*"
Else
txfo2 = False
End If
If patron3 = True Then
txfo3 = Ctx Like "**FC*"
Else
txfo3 = False
End If
If patron1 = True Then
rxfo1 = Crx Like "**ST*"
Else
rxfo1 = False
End If
If patron2 = True Then
rxfo2 = Crx Like "**SC*"
Else
rxfo2 = False
End If
If patron3 = True Then
rxfo3 = Crx Like "**FC*"
Else
rxfo3 = False
End If
If fo1 = True Or fo2 = True Or fo3 = True Then
fo = True
Else
fo = False
End If
If txfo1 = True Or txfo2 = True Or txfo3 = True Then
txfo = True
Else
txfo = False
End If
If rxfo1 = True Or rxfo2 = True Or rxfo3 = True Then
rxfo = True
Else
rxfo = False
End If
If fo = False Then
'Print "Enlace no viable, por incompatibilidad de la Fibra"
vbl = 1
End If
If txfo = False Then
'Print "Enlace no viable, por incompatibilidad del Transmisor"
vbl = 2
End If
```



```
If rxfo = False Then
'Print "Enlace no viable, por incompatibilidad del Receptor"
vbl = 3
End If
'Viabilidad referente a la longitud de onda
If LOMatx = LOenl Or LOMitx = LOenl Then
Itx = True
'Print "Enlace Viable"
Else
Itx = False
'Print "Enlace no viable, por lotx"
vbl = 4
End If
If LOMarx = LOenl Or LOMirx = LOenl Then
Irx = True
'Print "Enlace Viable"
Else
Irx = False
'Print "Enlace no viable, por lorx"
vbl = 5
End If
If LOMafo = LOenl Or LOMifo = LOenl Then
Ifo = True
'Print "Enlace Viable"
Else
Ifo = False
'Print "Enlace no viable, por lofo"
vbl = 6
End If
If fo = True And txfo = True And rxfo = True And Itx = True And Irx = True And Ifo = True Then
'Print "Enlace Viable"
vbl = 0
End If
viab = vbl
End Function
```

## BalPotencia.bas

Option Explicit

```
Public Function balpot(ptx As Single, prx As Single, afib As Single, _
    acon As Single, MS As Single, lenl As Single, NReg As Single) As Boolean
    Dim lmax As Single
    lmax = (ptx - (prx + 2 * acon + MS)) / afib
    If lmax > (lenl / (1 + NReg)) Then
        balpot = True
    Else
        balpot = False
    End If
End Function
```



## BalTiempos.bas

```
Public Function baltime(dis As Single, lLOnd As Single, lenl As Single, _
    ttx As Single, trx As Single, vel As Single, cod As Boolean, _
    NReg As Single) As Boolean
    Dim Tres As Single, TresCuad As Double, TmatCuad As Double, c As Single
    TmatCuad = (disp * lLOnd * (lenl / (1 + NReg))) ^ 2
    TresCuad = ((ttx ^ 2) + (trx ^ 2) + TmatCuad)
    Tres = Sqr(TresCuad)
    If cod = False Then
        c = 0.7
    Else
        c = 0.35
    End If
    If (Tres * vel * 0.001) < c Then
        baltime = True
    Else
        baltime = False
    End If
End Function
```

## Presupuesto.bas

```
Public Function coste(Ctx As Single, cadc As Single, Crx As Single, _
    campf As Single, ccon As Single, lenl As Single, cfib As Single, _
    NReg As Single) As Single
    coste = ((NReg + 1) * (2 * ccon + Ctx + cadc + Crx + campf)) + (lenl * cfib)
End Function
```

## Redondear.bas

```
Public Function Redondeo(ByVal Num As String) As Double
    Dim ParteEntera, ParteDecimal As Double
    Dim Intermedia As String
    Dim Temporal As Double
    If Not IsNumeric(Num) Then Num = Val(Num)
    Temporal = Num
    ParteEntera = Fix(Temporal)
    ParteDecimal = CCur(Temporal) - ParteEntera
    If Len(CStr(ParteDecimal)) <= 4 Then Redondeo = ParteEntera + ParteDecimal: Exit
Function
    Intermedia = Left(CStr(ParteDecimal), 5)
    ParteDecimal = Left(Intermedia, 4)
    If Val(Right(Intermedia, 1)) > 4 Then
        ParteDecimal = (ParteDecimal * 100) + 1
        ParteDecimal = ParteDecimal / 100
    End If
    Redondeo = ParteEntera + ParteDecimal
End Function
```



## Resultado.bas

```
Public resul(1000) As String
Public resulR(1000) As String
Public resulEnl(1000) As String
Public resulEnlR(1000) As String
Public resulBP(1000) As String
Public resulBPR(1000) As String
Public resulBT(1000) As String
Public resulBTR(1000) As String
Public resulV(1000) As String
Public resulVR(1000) As String
Public resulTs(1000) As String
Public resulTsR(1000) As String
Public resulcomp(1000) As String
Public resulcompR(1000) As String
```

## GestionUdl.bas

```
Public Const Tipo0 As String = "\emisores.udl"
Public Const Tipo1 As String = "\Receptores.udl"
Public Const Tipo2 As String = "\fibras.udl"
Public Function Existe(ByVal Fichero As String) As Boolean
    'nos permite saber si unfichero existe
    Existe = True
    On Error Resume Next
    If Fichero <> "" Then Existe = (Dir(Fichero) <> "") Else Existe = False
    Existe = (Err = 0) And Existe
    On Error GoTo 0
End Function
```



## AGRADECIMIENTOS

A mi tutor, Juan García Ortega, por su guía y consejos en la elaboración del Proyecto.

A mi compañero y amigo Gaba, magnífico programador y profundo conocedor de la Programación Orientada a Objetos, que ha sido un balón de oxígeno en los momentos críticos.

A todos mis amigos, que con su apoyo me han animado en la elaboración de este arduo trabajo, y muy especialmente a Paco y a Leti.