



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**ESTUDIO DE UNA CONEXIÓN REMOTA TRANSPARENTE ENTRE EQUIPO
DE NAVEGACIÓN AÉREA Y SU SISTEMA DE MANTENIMIENTO Y
MONITOREO**

Werner Asdrúbal Arriola Moscoso

Asesorado por el Ing. MsEE. PhD. Enrique Edmundo Ruiz Carballo

Guatemala, julio de 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ESTUDIO DE UNA CONEXIÓN REMOTA TRANSPARENTE ENTRE EQUIPO
DE NAVEGACIÓN AÉREA Y SU SISTEMA DE MANTENIMIENTO Y
MONITOREO**

TRABAJO DE GRADUACIÓN
PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR:

WERNER ASDRÚBAL ARRIOLA MOSCOSO

ASESORADO POR EL ING. MSEE. PHD. ENRIQUE EDMUNDO RUIZ
CARBALLO

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, JULIO DE 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

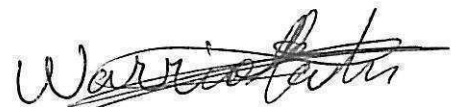
DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Guillermo Antonio Puente Romero
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADOR	Ing. Armando Alonso Rivera Carrillo
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ESTUDIO DE UNA CONEXIÓN REMOTA TRANSPARENTE ENTRE EQUIPO
DE NAVEGACIÓN AÉREA Y SU SISTEMA DE MANTENIMIENTO Y
MONITOREO,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, el 17 de septiembre de 2008.



Werner Asdrúbal Arriola Moscoso

Guatemala, 12 de mayo de 2009

Señor Ingeniero
Julio Cesar Solares Peñate
Coordinador Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Señor Coordinador:

Por este medio tengo el gusto de informar a usted, que he concluido con el asesoramiento del trabajo de graduación del estudiante **Werner Asdrúbal Arriola Moscoso**, con carné **200412932** el desarrollo de esta investigación como previo a su graduación se titula **Estudio de una conexión remota transparente entre equipo de navegación aérea y su sistema de mantenimiento y monitoreo**. Después de Revisar detenidamente su contenido final, considero su aprobación como trabajo científico de graduación.

Por lo tanto, el autor del trabajo de graduación y yo su asesor, nos hacemos responsables por el contenido y conclusiones de la misma.

Atentamente,


Ing. MsEE. PhD. (cand) Enrique Edmundo Ruiz Carballo
Colegiado No. 2225





FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
y Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

Guatemala, 18 de mayo de 2009

Señor Director
Ing. Mario Renato Escobedo Martínez
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.


Señor Director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **“ESTUDIO DE UNA CONEXIÓN REMOTA TRANSPARENTE ENTRE EQUIPO DE NAVEGACION AEREA Y SU SISTEMA DE MANTENIMIENTO Y MONITOREO”**, desarrollado por el estudiante **Werner Asdrúbal Arriola Moscoso**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS


Ing. Julio César Solares Peñate
Coordinador de Electrónica





REF. EIME 32.2009.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; Werner Asdrúbal Arriola Moscoso titulado: ESTUDIO DE UNA CONEXIÓN REMOTA TRANSPARENTE ENTRE EQUIPO DE NAVEGACIÓN AÉREA Y SU SISTEMA DE MANTENIMIENTO Y MONITOREO, procede a la autorización del mismo.

Ing. Mario Renato Escobedo Martínez

DIRECTOR



GUATEMALA, 20 DE MAYO 2,009.



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **ESTUDIO DE UNA CONEXIÓN REMOTA TRANSPARENTE ENTRE EQUIPO DE NAVEGACIÓN AÉREA Y SU SISTEMA DE MANTENIMIENTO Y MONITOREO**, presentado por el estudiante universitario **Werner Asdrúbal Arriola Moscoso**, autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olimpo Paiz Recinos
DECANO



Guatemala, julio de 2009

AGRADECIMIENTOS A:

- Dios: Por haberme ayudado y permitido la consecución de una meta más en mi vida.
- Mi familia: A cada uno de los que de alguna manera me ayudaron.
- Mi papá: Por haberme apoyado y acompañado en todo.
- El Ing. Enrique Ruiz Carballo: Por los consejos y carácter motivador que me mostró en el desarrollo de este trabajo.
- Mis amigos: Por compartir su tiempo y ser sinceros.
- La Universidad de San Carlos: Por haberme dado la oportunidad de ser parte de ella y permitirme concluir esta meta.
- Mis catedráticos: Por el empeño que dedicaron en la transmisión del conocimiento y por su paciencia.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	VII
RESUMEN	IX
OBJETIVOS	XI
INTRODUCCIÓN	XIII

1. RADIOAYUDAS

1.1 Sistemas de navegación por radio emplazados en tierra.....	1
1.1.1 VOR/DME (Radiofaro Omnidireccional de VHF) / (Equipo Medidor de Distancia).....	1
1.1.2 ILS (Sistema de Aproximación Instrumental de Precisión).....	2
1.1.3 EI NDB (Radiofaro No Direccional).....	3
1.1.4 TACAN (Tactical Air Navegation).....	3
1.2 Precisión.....	4
1.3 <i>Software</i> de mantenimiento y monitoreo ADRAC.....	4
1.3.1 Forma de comunicación con el equipo externo.....	5
1.4 Conexión actual para mantenimiento y monitoreo.....	5
1.4.1 Local.....	6
1.4.2 Por radiomodem.....	6

2. REDES

2.1 Clases de redes involucradas.....	9
2.1.1 Estructura y aparatos que conforman la red primaria.....	9
2.1.1.1 <i>Router</i> y su configuración.....	10

2.1.1.1.1 Configuración en un entorno de consola.....	10
2.1.2 Propuesta para la red secundaria.....	11
2.1.2.1 <i>Router</i> y su configuración.....	12
2.1.3 Proveedores de servicio de <i>Internet</i> (ISP).....	12
2.2 Clases de direcciones IP utilizadas.....	12
2.2.1 Direcciones IP públicas.....	13
2.2.2 Direcciones IP privadas (reservadas).....	13
2.2.3 Direcciones IP estáticas (fijas)	14
2.2.4 Direcciones IP dinámicas.....	14
2.3 Asignación de direcciones IP.....	14
2.3.1 Asignación de direcciones IP en la red primaria.....	15
2.3.2 Asignación de direcciones IP en la red secundaria.....	15
2.3.3 Solución para las direcciones IP involucradas.....	15
2.3.3.1 <i>Software</i> de publicación de dirección IP.....	16
2.3.3.1.1 No-Ip.....	16
2.3.3.2 Traductor de dirección de red y puerto (NAPT)	18
3. PUERTO SERIE	
3.1 Clases de conectores.....	22
3.1.1 Configuración y señales de los pines.....	22
3.2 <i>Handshaking</i>	24
3.3 Null-modem.....	24
3.4 Protocolo utilizado en el formato de los datos.....	25
4. LENGUAJE DE PROGRAMACIÓN VISUAL	
4.1. Entorno Integrado de Desarrollo (IDE)	27
4.2. C#.....	28

4.2.1. <i>Multithreading</i>	28
4.2.2. <i>Sockets</i>	29
4.2.2.1. Servidor.....	30
4.2.2.2. Cliente.....	31
4.2.3. Puerto serial.....	31

5. ASPECTOS ECONÓMICOS

5.1 Licencias.....	35
5.2 Desplazamiento de personal de forma periódica.....	36
5.2.1 Lugares y períodos de desplazamiento.....	36
5.3 Colocación de computadoras dedicadas.....	38
5.4 Colocación de una red.....	38

6. SOFTWARE REDIRECCIÓN DE DATOS

6.1. Código fuente.....	39
6.1.1. Funciones utilizadas.....	47
6.1.1.1. Carga del formulario.....	47
6.1.1.2. El servidor.....	48
6.1.1.3. El cliente.....	48
6.1.1.4. Abrir puerto COM.....	48
6.1.1.5. Cerrar puerto COM.....	49
6.1.1.6. Habilitar y deshabilitar opciones de configuración.....	49
6.1.2. Conexión hacia la red.....	49
6.1.3. Conexión del puerto serial.....	50
6.2. Requisitos del sistema.....	52
6.2.1 Red primaria.....	52
6.2.2 Red secundaria.....	53

6.3. Manejo de la aplicación.....	53
6.3.1. Recursos <i>hardware</i> configurables.....	53
6.3.1.1 Red primaria.....	54
6.3.1.2 Red secundaria.....	55
6.3.2. Configuración específica de parámetros.....	55
6.4. Descripción general de funcionamiento.....	56
6.4.1 Del <i>software</i> de redirección.....	57
6.4.2 Del proyecto total.....	58
6.4.2.1 En el sitio de la radioayuda.....	58
6.4.2.1 Con el monitor centralizado.....	59
7. LÍNEAS DE CONTINUACIÓN	
7.1 Capacidad de modificación de <i>software</i>	61
7.1.1 Posibilidad de agregar algoritmos de seguridad de autenticación personalizada.....	62
7.1.2 Aumento de capacidad hasta el límite requerido agregando características personalizadas.....	62
7.2 Generalización de diseño para otros usos.....	63
7.2.1 Centralización de poder computacional y ahorro de licencias.....	63
CONCLUSIONES	65
RECOMENDACIONES	67
BIBLIOGRAFÍA	69
BIBLIOGRAFÍA ELECTRÓNICA	71

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Antena DVOR 432/DME 435.....	2
2.	Equipo DME.....	2
3.	RCSU (<i>Remote Control & Status Unit</i>).....	5
4.	Conexión actual para monitoreo y mantenimiento de forma individual local.....	6
5.	Conexión actual para monitoreo y mantenimiento por radiomodem.....	6
6.	Pines del conector RJ-45 del <i>router</i>	11
7.	Opciones de creación.....	17
8.	Modos de configuración.....	19
9.	Configuración de pines en el conector DB9.....	22
10.	Configuración de pines en el conector DB25.....	23
11.	Conexión de pines en un cable <i>null-modem</i>	25
12.	Ubicación del objeto <i>SerialPort</i> en <i>Visual Studio</i>	32
13.	Gráfica cualitativa de precio de mantenimiento personal, precisión y precio de mantenimiento remoto, contra número de veces que se calibra en un período de tiempo dado.....	37
14.	Modelo de formulario para utilizar en el proyecto.....	40
15.	Parte funcional del programa que ejecuta la elección del método de ingreso de IP.....	50
16.	Parte funcional del programa en donde se selecciona el puerto a redireccionar.....	51
17.	Parte funcional del programa en donde se selecciona la configuración del puerto redireccionado	51

18.	Interconexión de puertos rs232 de una misma PC.....	54
19.	Interconexión de puertos rs232 de la PC de redirección y el puerto remoto.....	55
20.	Configuración básica de parámetros.....	56
21.	Diagrama general final de redireccionamiento sin RCSU.....	59
22.	Diagrama general final de redireccionamiento con RCSU.....	60

TABLAS

I.	Interconexión de pines.....	11
II.	Descripción de señales de las dos diferentes clases de conectores seriales	23

GLOSARIO

ACL	<i>Access control list</i>
ADRAC	<i>Automatic Digital Recording and Control</i>
DOMINIO	Es el que agrupa un conjunto de equipos y permite proveer nombres de equipo para recordarlos fácilmente en lugar de una dirección <i>IP</i> numérica.
IDE	<i>Integrated development environment</i>
ISP	<i>Internet Service Provider</i>
NAPT	<i>Network Address Port Translation</i>
RCSU	<i>Remote Control & Status Unit</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>

RESUMEN

Actualmente, COCESNA cuenta con equipo de radioayuda aérea instalada en varios puntos del país, sin embargo, el mantenimiento y calibración del equipo requiere de un considerable tiempo de movilización de personal hacia el sitio de operación de dicho equipo, haciéndose necesaria alguna forma de control remoto y lograr una centralización del personal capacitado y poder dar no sólo un mantenimiento más rápido sino más frecuente y hacerlo de una forma más económica.

El equipo de radioayuda aérea es fundamental para los vuelos de aeronaves en el territorio y de esta forma un mantenimiento y calibración más frecuente hace más confiable a dicho equipo.

Una conexión transparente remota de datos permite que el personal técnico con conocimiento específico en equipo especializado no se preocupe de configuraciones remotas y pueda actuar como si se tratara de una configuración presencial.

La centralización del *software* de mantenimiento implica una reducción de licencias de instalación y soporte de parte de la empresa que proporciona el *software* debido a que es necesario sólo una computadora que posea todo el poder computacional.

En el desarrollo de proyectos tales como el de redirección de puertos, resulta conveniente el uso de un IDE visual, proporcionando las características necesarias que hay que configurar y relacionar de forma correcta.

OBJETIVOS

- **General:**

Dar a conocer el proceso para lograr una comunicación transparente con redirección de datos, entre personal capacitado para el monitoreo y mantenimiento y el equipo de navegación aérea y también el ahorro en licencias, tiempo y recursos que éste implicaría.

- **Específicos:**

1. Crear un *software* adecuado para redireccionar un puerto RS232 hacia el puerto de red y viceversa con las conexiones apropiadas.
2. Analizar las configuraciones de los distintos equipos de la red para lograr una redirección transparente de datos entre computadoras dedicadas.
3. Estudiar las distintas clases de dirección IP utilizadas o propuestas, su repercusión en la interconexión y soluciones posibles para cada una de ellas.
4. Estudiar los aspectos económicos básicos de la implementación del proyecto.

INTRODUCCIÓN

Las formas de comunicación remota han ido evolucionando desde la transmisión de ideas a puntos lejanos por medios escritos, hasta la actualidad, con la aparición de redes de comunicación que permiten la transmisión de cualquier tipo de datos.

Una de las mayores ventajas de muchos equipos es que el mantenimiento y monitoreo es por medio de enlaces de datos en forma de paquetes, permitiendo así un procesamiento de dichos paquetes y la transmisión hacia cualquier punto lejano.

Las comunicaciones remotas y en especial la redirección de datos técnicos, tienen una implicación muy grande en la actualidad, debido a que se logra una menor movilización física de personal y una mayor rapidez y frecuencia para la realización de mantenimiento y monitoreo.

Una implementación remota de forma transparente entre equipo de suma importancia aérea y el sistema de mantenimiento y monitoreo, implica redirección de datos por medio de configuraciones de los diferentes caminos por donde se enviarán y mayor capacidad de actuación técnica y reducción de costos.

1. RADIOAYUDAS

Las Radioayudas Aeronáuticas son diferentes instalaciones equipadas con equipos electrónicos de radiofrecuencia que proporcionan a las aeronaves diferentes servicios de ayuda a la navegación aérea.

1.1 Sistemas de navegación por radio emplazados en tierra

Su uso se inició en el año 1930 y en la actualidad es el sistema de navegación más empleado. Basan su funcionamiento en estaciones emplazadas en tierra que emiten señales de radiofrecuencia, las cuales son captadas por sensores a bordo de las aeronaves, permitiendo que cada aeronave calcule su posición.

1.1.1 VOR/DME (Radiofaro Omnidireccional de VHF) / (Equipo Medidor de Distancia)

El VOR son los radiofaros que proporcionan el rumbo de la aeronave en la ruta para orientar su dirección; a su vez, el equipo DME asociado ofrece la información continua de la distancia respecto a la estación; en la figura 1 se puede observar el sitio de las antenas y en la figura 2 el equipo dentro de las instalaciones.

Figura 1 Antena DVOR 432/DME 435



(COCESNA, **Boletín informativo año 6, No. 002/2005**, pág. 6)

Figura 2 Equipo DME



(COCESNA, **Boletín informativo año 6, No. 002/2005**, pág. 7)

1.1.2 ILS (Sistema de Aproximación Instrumental de Precisión)

Sistema de antenas colocado en un extremo de la pista del aeródromo que suministra al avión la información sobre la trayectoria correcta al inicio de ésta y del ángulo adecuado en el descenso.

1.1.3 El NDB (Radiofaro No Direccional)

Da una posición angular de la aeronave respecto al norte magnético. Es el sistema de navegación más antiguo en la aviación, pero aún hoy en día es uno de los más necesarios.

Opera en uno rango de frecuencia de 190 – 400 Hhz. Y su señal es transmitida en todas las direcciones. El equipo receptor a bordo se llama ADF (*Automatic Direction Finder*) o “Localizador automático de dirección”.

Así como la brújula señala siempre el norte magnético, la aguja del ADF señalará siempre la dirección de la estación NDB que se haya sintonizado. En este caso se debe girar el avión hasta ubicar esta aguja hacia delante.

1.1.4 TACAN (Tactical Air Navegation)

Fue básicamente desarrollado para satisfacer necesidades militares, único usuario que en la actualidad emplea todas las funciones del sistema.

La aviación civil hace uso únicamente de la función distancia del tacan, con el nombre de equipo radiotelemétrico de uhf (dme).

La información que proporciona al piloto es la de azimut y la distancia con respecto a la instalación de tierra, dando así, en cada instante, la posición del avión.

La identificación de las estaciones TACAN es auditiva, en código MORSE, y está compuesta por tres letras que se repiten una vez cada 30 segundos.

1.2 Precisión

La precisión de estos sistemas depende de la distancia a la que se encuentre el avión respecto a las radioayudas. El VOR y el NDB ofrecen una precisión de $\pm 5^\circ$ y $\pm 10^\circ$, respectivamente. Para el VOR, si el avión está a 30 km, debe estar máximo ± 2624 metros a la derecha o a la izquierda de la línea recta formada por el avión y el VOR y si está a 5 km debe estar máximo a ± 437 metros a la derecha o a la izquierda de la línea recta. Ocurre lo mismo con el NDB para estas distancias con la salvedad que este permite máximo 10° . Sin embargo, los fabricantes de radioayudas ofrecen márgenes más seguros.

1.3 Software de mantenimiento y monitoreo ADRAC

Es el que provee el fabricante de las radioayudas y es utilizado para monitorear el funcionamiento de cada una de ellas, se puede modificar parámetros de las mismas, para lograr una adecuada calibración.

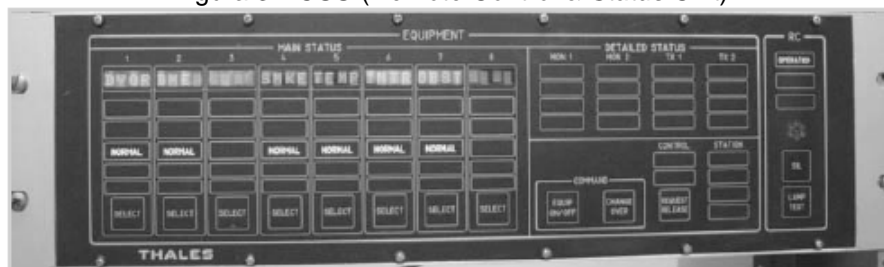
1.3.1 Forma de comunicación con el equipo externo

La comunicación del *software* con el exterior se realiza por medio de comunicación rs232, y es esta clase de datos en donde se hará el redireccionamiento.

1.4 Conexión actual para mantenimiento y monitoreo

El mantenimiento del equipo se hace por medio de la conexión serial entre un computador con el *software* ADRAC, *Automatic digital recording and control* (Control y registro digital automático) y el monitor centralizado RCSU (Remote Control & Status Unit, figura 3) instalado en la sala de equipos del Edificio Regional (a través de un radioenlace permite el monitoreo control y mantenimiento de las radiayudas a una distancia relativamente corta), o entre este mismo computador y la radioayuda específica a la que se le quiere dar mantenimiento.

Figura 3 RCSU (*Remote Control & Status Unit*)

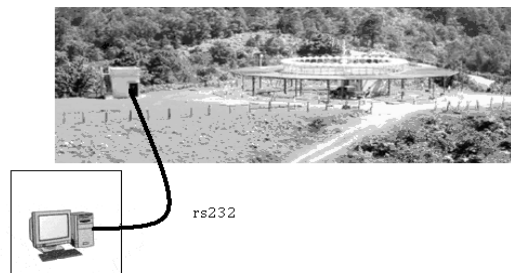


(COCESNA, Boletín informativo año 6, No. 002/2005, pág. 6)

1.4.1 Local

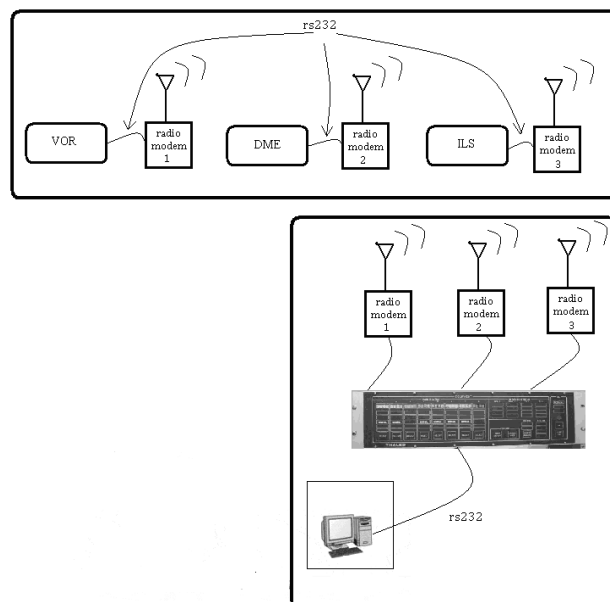
Esta es la forma más básica de dar mantenimiento a cada radioayuda y consiste en conectar la computadora con el *software* ADRAC directamente a cada equipo de la radioayuda, esta conexión se ilustra en la figura 4.

Figura 4 Conexión actual para monitoreo y mantenimiento de forma individual local



1.4.2 Por radiomodem

Figura 5 Conexión actual para monitoreo y mantenimiento por radiomodem.



La comunicación se centraliza en el RCSU y se conecta a este equipo la computadora con el *software* ADRAC para dar mantenimiento, como se puede observar en la figura 5.

2. REDES

Una red es una interconexión de dispositivos para compartir información, recursos y servicios. Esta interconexión puede ser a través de un enlace físico o inalámbrico.

2.1. Clases de redes involucradas

En el proyecto se considerará la red instalada en las oficinas centrales a la cual se le denominará primaria, y la red que se propone instalar en el sitio de las radioayudas, la cual se denominará red secundaria, dichas redes estarán en contacto por medio de *INTERNET*.

2.1.1. Estructura y aparatos que conforman la red primaria

La estructura de esta red está basada en un router y varios *switches*, de tal manera que se logra un control de direcciones IP internas por medio del router y se distribuyen con los *switches* de los cuales se tomará un puerto para la computadora dedicada.

2.1.1.1. Router y su configuración

El router utilizado en la red primaria es un CISCO1700, este es el principal aparato que necesita de configuración para el redireccionamiento de datos, debido a que es aquí donde se concentran los paquetes y se decide hacia donde se envían.

Existen diferentes formas de configurar el router CISCO1700, una es por medio de consola (el más específico) y otra es por medio de un entorno visual intuitivo.

2.1.1.1.1. Configuración en un entorno de consola

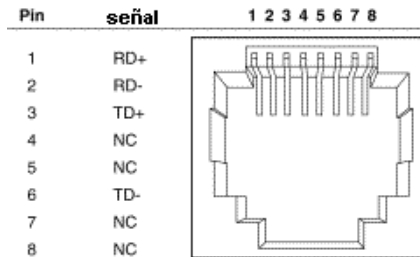
Se utiliza un cable roll-over para conectar un pc hacia el router de tal manera que se pueda acceder este por medio de consola y lograr así configurarlo manualmente.

El cable utilizado en la configuración denominado roll-over se utiliza entre un router (puerto de consola) y una PC (puerto serial) con un emulador de Terminal y con envío de datos ASCII (*HyperTerminal*), en la tabla I se observa la interconexión de pines de los conectores a utilizar y en la figura 6 se observa un conector RJ-45 del router, el conector DB9 se discutirá en otro capítulo.

Tabla I Interconexión de pines

Puerto de Consola (DTE), (pc)	RJ-45-to-RJ-45 Cable Rollover		RJ-45-a-DB9(female) Adaptador de terminal	Dispositivo de Consola (DCE), (router)
Señal	RJ-45 Pin	RJ-45 Pin	DB-9 Pin	Señal
RTS	1	8	8	CTS
No conectado	2	7	6	DSR
TxD	3	6	2	RxD
GND	4	5	5	GND
GND	5	4	5	GND
RxD	6	3	3	TxD
No conectado	7	2	4	DTR
CTS	8	1	7	RTS

Figura 6 Pines del conector RJ-45 del *router*



2.1.2. Propuesta para la red secundaria

Debido a que actualmente en el sitio de instalación de las radioayudas no se cuenta con una red, se aprovechará que en este lugar se posee una línea telefónica que permite crear un acceso a *Internet*, necesitándose por consiguiente tan solo una computadora dedicada que se conecte desde el sitio hasta las oficinas centrales por medio del *software* de redireccionamiento, que se creará para dicho propósito.

2.1.2.1. Router y su configuración

Actualmente no se cuenta con un router específico, la configuración será de una forma parecida a la del router de la red primaria o en dado caso se podría hacer de manera visual de forma sencilla, configurando lo mismo que se configure en el router de la red primaria anteriormente mencionado.

2.1.3. Proveedores de servicio de *Internet* (ISP)

Para la red primaria el ISP se encuentra fuera del país y provee de *Internet* a las instalaciones centrales por medio de un enlace satelital punto a punto.

Para la red secundaria se propone que el ISP sea capaz de dar servicio a través de la línea telefónica, de tal manera que se aproveche la infraestructura actual y se permita utilizar el *Internet* no solo para el mantenimiento remoto de radioayudas, sino también para otros servicios como *Volp*, etc.

2.2. Clases de direcciones IP utilizadas

La dirección IP es el identificador de cada host dentro de su red. Cada host conectado a una red tiene una dirección IP asignada, la cual debe ser distinta a todas las demás direcciones que estén vigentes en ese momento en el conjunto de redes visibles por el host.

En el caso de *Internet*, no puede haber dos ordenadores con 2 direcciones IP (públicas) iguales. Pero sí se podría tener dos ordenadores con la misma dirección IP, siempre y cuando pertenezcan a redes independientes entre sí (sin ningún camino posible que las comunique).

2.2.1. Direcciones IP públicas

Son visibles en todo *Internet*. Un ordenador con una IP pública es accesible (visible) desde cualquier otro ordenador conectado a *Internet*. Para conectarse a *Internet* es necesario tener una dirección IP pública.

2.2.2. Direcciones IP privadas (reservadas)

Son visibles únicamente por otros hosts de su propia red o de otras redes privadas interconectadas por routers. Se utiliza en la empresa para los puestos de trabajo. Los ordenadores con direcciones IP privadas pueden salir a *Internet* por medio de un router que tiene una IP pública. Sin embargo, desde *Internet* no se puede acceder a ordenadores con direcciones IP privadas a menos que se haga un redireccionamiento, lo cual se efectúa en el proyecto.

2.2.3. Direcciones IP estáticas (fijas)

Un host que se conecte a la red con dirección IP estática siempre lo hará con una misma IP. Las direcciones IP públicas estáticas son las que utilizan los servidores de *Internet* con objeto de que estén siempre localizables por los usuarios de *Internet*. Estas direcciones hay que contratarlas.

2.2.4. Direcciones IP dinámicas

Un host que se conecte a la red mediante dirección IP dinámica, cada vez lo hará con una dirección IP distinta. Las direcciones IP públicas dinámicas son las que se utilizan en las conexiones a *Internet* mediante un módem. Los proveedores de *Internet* utilizan direcciones IP dinámicas debido a que tienen más clientes que direcciones IP (es muy improbable que todos se conecten a la vez).

2.3. Asignación de direcciones IP

Las direcciones asignadas en las redes involucradas son de diferentes clases, debido a esto se dividen las redes como se hizo anteriormente en primaria y secundaria.

2.3.1. Asignación de direcciones IP en la red primaria

Las direcciones utilizadas se dividen en internas y externas, siendo las primeras estáticas y asignadas por medio del router, las segundas son dinámicas y asignadas por el ISP, creando así la necesidad de un mapeo de tráfico de direcciones externas a internas y viceversa.

2.3.2. Asignación de direcciones IP en la red secundaria

Debido a que actualmente no se cuenta con ISP la asignación aún no está definida pero siguiendo la estructura propuesta anteriormente, esta red poseería un MODEM, el cual le pasaría a un router una dirección externa debido a que se utilizará directamente la línea telefónica; dicha dirección externa se propone que sea dinámica, ya que ésta es más económica e igualmente útil para los propósitos generales, tal como el caso del *software* de redirección de datos.

2.3.3. Solución para las direcciones IP involucradas

Dado que las direcciones IP públicas para conexión a *Internet* son dinámicas se requiere que el *software* de redirección de datos sea actualizado continuamente sobre la dirección actual de cada una de las computadoras involucradas, esto se puede lograr al publicar periódicamente la dirección actual en un servidor, por medio de algún *software* de publicación de dirección IP y localizar la IP por medio de un nombre invariante.

También tomando en cuenta que la red primaria y la secundaria son cada una redes intranet, con direcciones IP internas estáticas, necesitarán ser mapeadas hacia una dirección IP pública que proporciona el ISP, por medio de NAT.

2.3.3.1. Software de publicación de dirección IP

Existen diferentes clases de *software* que publican alguna dirección actual ip en un servidor gratuito, permitiendo así conocer cuál es la dirección por medio de un nombre (dominio) tal como nombre.extension.xx.xx.

2.3.3.1.1. No-Ip

Para crear una DNS gratuita se puede crear una cuenta en No-Ip.com, para esto se necesita ir a la dirección <http://www.no-ip.com>.

Después de seguir todos los pasos para crear la cuenta es necesario autenticarse y así tener servicios de DNS gratuitos, limitados pero suficientes para los propósitos del proyecto que se estudia.

Después de la autenticación, crear un host (figura 7), aquí solicitan un nombre de host (*hostname*) el cual se elige con algún nombre aleatorio (nombrealeatorio.no-ip.biz) y dejando las otras opciones como aparecen, por último dar un *click* en crear.

Figura 7 Opciones de creación

Own a domain name?
Use your own domain name with our DNS system. Add your domain name now or read more for pricing and features.

Hostname Information

Hostname: nombreakaleatorio no-ip.biz

Host Type: DNS Host (A) DNS Host (Round Robin) DNS Alias (CNAME)
 Port 80 Redirect Web Redirect

IP Address: 201.216.146.61

Assign to Group: - No Group - [Configure Groups](#)

Enable Wildcard: Wildcards are a Plus / Enhanced feature. [Upgrade Now!](#)

Ya creado el host se procede a descargar de la página el *software* que se instalará en la computadora de IP dinámica, el cual actualizará la dirección IP cada cierto tiempo en el servidor gratuito.

El último paso es instalar y configurar adecuadamente el *software* de publicación de dirección IP, para lo cual solo se necesita darle la dirección de correo electrónico y la clave para autenticarse en la página de no-ip y luego indicarle el host que se creó unos pasos atrás (nombreakaleatorio.no-ip.biz).

Por último se comprueba haciendo un ping en ventana de consola DOS hacia el nombre del servidor y comprobar que todo funciona bien(>ping nombreakaleatorio.no-ip.biz), si esto sucede ya se puede utilizar el nombre de dominio en vez de la dirección IP pública es decir en vez de escribir por ejemplo 200.251.16.110 se escribe nombreakaleatorio.no-ip.biz.

2.3.3.2. Traductor de dirección de red y puerto (NAPT)

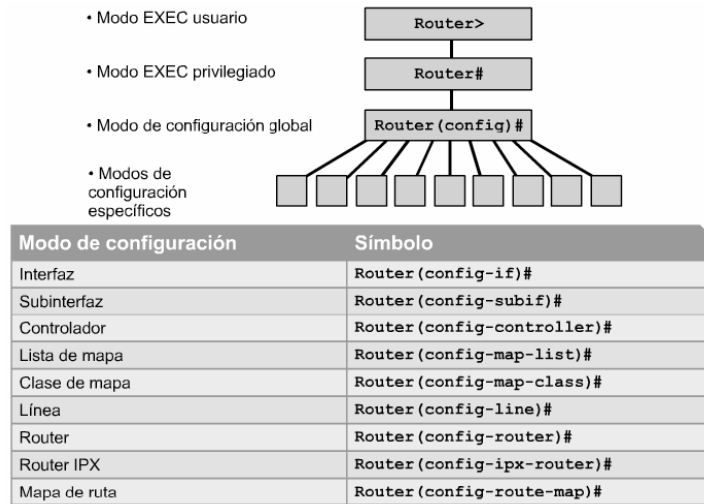
En NAPT se utilizan, aparte de las direcciones IP, los identificadores de puerto ampliando así un nivel más el concepto de traducción/asociación. Al utilizar los identificadores de puerto (TCP/UDP) en las traducciones, se consigue traducir/asociar varias direcciones IP privadas a una única dirección global IP, ya que se utiliza multiplexación de puertos utilizando una única dirección global IP. NAPT permite conectar varios host de una *inside network*, al compartir una sola dirección de la *outside network*.

Se utilizará NAPT para traducir la dirección interna proporcionada a la computadora dedicada para el proyecto y el puerto asociado del *software* de redireccionamiento, con el puerto y dirección pública.

La configuración de un router se hace a través del cable conectado en su puerto de consola por medio de un PC, el cual tiene en un puerto serial conectado el otro extremo del cable como se mencionó en párrafos anteriores, ahora se necesita llegar al modo de configuración específica como se muestra en la figura 8.

El modo específico utilizado para la configuración de NAPT es el de interfaz.

Figura 8 Modos de configuración



(Cisco, **CCNA 1 and 2, Versión 3.1**, pág. 273)

Ahora los comandos específicos serán:

R>

R>enable

R>Password:

R#

R#configure terminal

R(config)#

Se crea una ACL (*access control list*) que identifique a los hosts para la traslación

La ACL da un informe sobre los permisos o derechos de acceso que tiene cada usuario sobre un objetivo determinado (como un directorio o un archivo).

R(config)#access-list 1 permit 172.16.1.0 0.0.0.255

Configurar NAPT basado en la dirección origen

```
R(config)#ip nat inside source list 1 interface s0 overload
```

Especificar la interfaz interna

```
R(config)# interface e0  
R(config-if)# ip nat inside  
R(config-if)# exit
```

Especificar la interfaz externa

```
R(config)# interface s0  
R(config-if)# ip nat outside  
R(config-if)# exit  
R(config)#exit
```

3. PUERTO SERIE

Los puertos seriales (también llamados RS-232, por el nombre del estándar al que hacen referencia) fueron las primeras interfaces digitales que permitieron que los equipos intercambien información con el "mundo exterior". El término serial se refiere a los datos enviados mediante un solo hilo: los bits se envían uno detrás del otro.

La comunicación serial se lleva a cabo asincrónicamente, es decir que no es necesaria una señal (o reloj) de sincronización: los datos pueden enviarse en intervalos aleatorios, sin embargo se pueden controlar estos intervalos por medio de otras líneas dedicadas para *handshaking*.

Cada paquete de datos puede ser de cinco a ocho bits de datos y se encuentra precedido por un bit de ARRANQUE y seguido por uno, uno y medio o dos bits de PARADA. Estos bits de control, necesarios para la transmisión serial, desperdician un cierto porcentaje del ancho de banda (por ejemplo cada 10 bits enviados, 8 se utilizan para cifrar el carácter y 2 para la recepción, desperdiciando un 20%) sin embargo para conexiones en las que la velocidad no es un factor tan importante este desperdicio de ancho de banda se desprecia, como es el caso que se considera en las radioayudas.

3.1. Clases de conectores

Existen dos clases de conectores estándar y dos formas para cada una de ellas.

Una clase es el conector DB9 que puede ser hembra o macho, la otra clase es el conector DB25 que puede ser también hembra o macho, no obstante las radioayudas poseen en sus terminales de comunicación, terminales DB9 macho.

3.1.1. Configuración y señales de los pines

La configuración de pines en el conector DB9 se puede observar en la figura 9, la del conector DB25 en la figura 10 y finalmente las señales según el número de pin se puede observar en la tabla II.

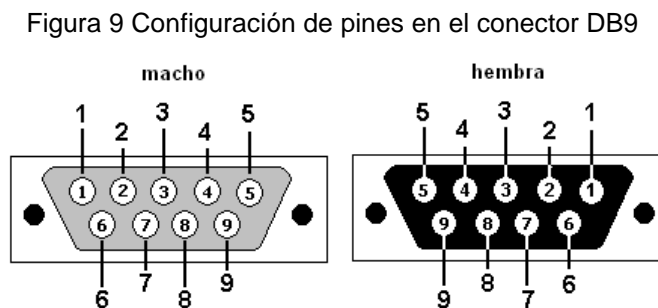


Figura 10 Configuración de pines en el conector DB25

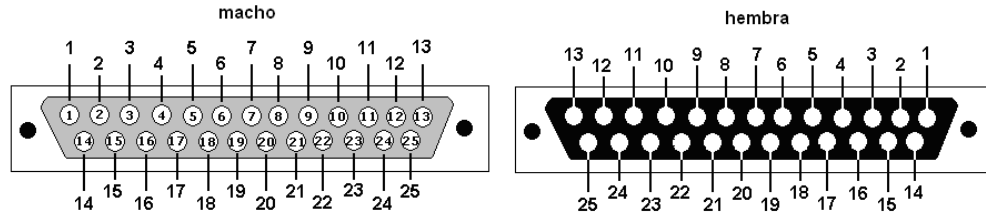


Tabla II Descripción de señales de las dos diferentes clases de conectores seriales.

No. DB25	No. DB9	Acroni-mo	Dirección	Significado en inglés	Significado en español
1		-	-	Protective/shielded ground	Tierra de protección
2	3	TxD	OUT	Transmit Data (a.k.a TxD, Tx) (ASYNC)	Emisión de datos
3	2	RxD	IN	Receive Data (a.k.a RxD, Rx) (ASYNC)	Recepción de datos
4	7	RTS	OUT	Request To Send (ASYNC)	Petición de emisión
5	8	CTS	IN	Clear To Send (ASYNC)	Preparado para transmitir
6	6	DSR	IN	Data Set Ready (ASYNC)	Equipo de com. preparado
7	5	SGND	-	Signal Ground	Tierra de señal
8	1	CD	IN	Carrier Detect (a.k.a DCD).	Detección de portadora
9		-	-	Reserved for data set testing.	Reservada
10		-	-	Reserved for data set testing.	Reservada
11		-	-	Unassigned	No asignada
12		SDCD	IN	Secondary Carrier Detect. Only needed if second channel being used.	Detección de portadora sec.
13		SCTS	IN	Secondary Clear to send. Only needed if second channel being used.	Preparado para transmitir sec.
14		STD	OUT	Secondary Transmit Data. Only needed if second channel being used.	Emisión de datos secundario
15		DB	OUT	Transmit Clock (a.k.a TCLK, TxCLK). Synchronous use only.	Reloj de transmisión
16		SRD	IN	Secondary Receive Data. Only needed if second channel being used.	Recepción de datos secundario
17		DD	IN	Receive Clock (a.k.a RCLK). Synchronous use only.	Reloj de recepción
18		LL	-	Local Loopback	Bucle local
19		SRTS	OUT	Secondary Request to Send. Only needed if second channel being used.	Petición de emisión Secundario

20	4	DTR	OUT	Data Terminal Ready. (ASYNC)	Terminal de datos preparado
21		RL/SQ	-	Signal Quality Detector/Remote loopback	Detector de señal de calidad
22	9	RI	IN	Ring Indicator. DCE (Modem) raises when incoming call detected used for auto answer applications.	Indicador de llamada
23		CH/CI	OUT	Signal Rate selector.	Selección de velocidad
24		DA	-	Auxiliary Clock (a.k.a. ACLK). Secondary Channel only.	Reloj auxiliar de transmisión externo
25		-	-	Unassigned	No asignada

3.2. Handshaking

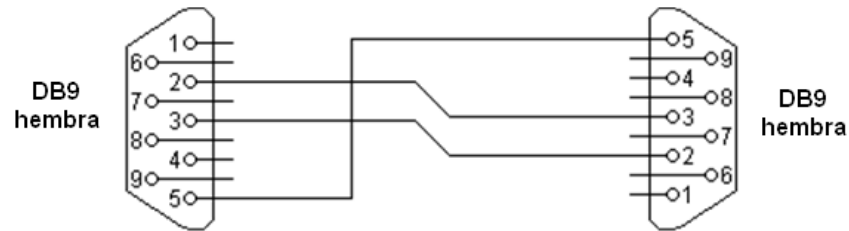
Es la comunicación de apretón de manos y significa que cada terminal serial le avisa a la otra si está lista para recibir datos o si posee datos para ser enviados.

En el caso de la comunicación con las radioayudas no se utiliza el *handshaking*.

3.3. Null-modem

Se refiere a la comunicación en la que solo se utilizan tres líneas para la comunicación (Rx, Tx, y GND), y se asume que siempre están listas las terminales para enviarles datos, un cable típico de conexión se observa en la figura 11.

Figura 11 Conexión de pines en un cable *null-modem*.



Esta es la forma en que se comunican las radioayudas con los equipos externos.

3.4. Protocolo utilizado en el formato de los datos

Se refiere a la forma en que se comunican las terminales rs232, es decir si utilizan una codificación específica ya sea ASCII, decimal o simplemente paquetes de datos sin significado estándar, para transmitir determinados comandos.

Las radioayudas utilizan simplemente paquetes de datos sin codificación estándar, lo cual no tiene mucha importancia debido a que al plantear el proyecto en base al estudio de las mejores opciones, el redireccionamiento se hace considerando a los datos como byte y no como mensajes específicos; la configuración es 8 bits de datos, a una velocidad de 9600 baudios, un bit de stop y sin paridad.

4. LENGUAJE DE PROGRAMACIÓN VISUAL

La programación visual surge con la necesidad de colocar elementos con propiedades determinadas de una manera rápida y efectiva de tal manera que se tengan ya modelos preestablecidos con características definidas.

En el proyecto que se plantea se utiliza VPL (lenguaje de programación visual) debido a que los elementos necesarios para ejecutar redireccionamiento se obtienen de forma bastante simplificada al tomar elementos visuales.

4.1. Entorno Integrado de Desarrollo (IDE)

Se trabajará en Visual Studio debido a que es un entorno bastante completo y confiable además de tener todas las herramientas para desarrollar el proyecto.

Visual Studio es utilizado para programar varias clases de lenguajes entre los cuales se encuentra C#.

4.2. C#

C# es un moderno lenguaje muy poderoso el cual ha sido hecho teniendo en mente la experiencia de desarrolladores durante muchos años y sobre sus necesidades. Uno de los mas grandes objetivos en el comercio es hacer programas re-usables, mantenibles, escalables, portables y fáciles de depurar. C# es adecuado para las anteriores necesidades. Y es por todo lo anterior el lenguaje que se selecciona para el proyecto.

4.2.1. *Multithreading*

Multithreading es una característica que provee el sistema operativo, que habilita a alguna aplicación a tener al “mismo” tiempo más de una vía de ejecución.

Multithreading y multitarea son dos conceptos relacionados y es tan común en la actualidad que es difícil encontrar aplicaciones Windows que no usen esto, por ejemplo se puede ejecutar un programa de reproducción de música mientras se edita un texto en otro programa y se baja algún archivo de *Internet* también con un programa distinto a los anteriores, siendo esto multitarea pero cuando *un mismo programa* ejecuta varias funciones como por ejemplo reproducir sonido y video al “mismo” tiempo se le llama *multithreading*.

La forma de hacer esto es por medio de la división del tiempo de ejecución del procesador.

Para implementar en C# *multithreading* se requiere agregar el encabezado `using System.Threading` y utilizar un delegado, como se muestra en el siguiente ejemplo:

```
Thread ThreadUno = new Thread(new ThreadStart(Fun1));
```

Aquí `ThreadUno` es el delegado, el cual ejecutará `Fun1` (alguna función definida por el programador) en un thread distinto al que se ejecute todo lo demás del programa con tan solo colocar el siguiente código:

```
ThreadUno.Start();
```

4.2.2. Sockets

Un socket es una descripción abstracta del significado de enviar y recibir información entre una computadora/aplicación en una red. Este describe una conexión entre dos puntos de comunicación.

Los *sockets* constan de tres piezas importantes:

- El protocolo (como TCP/IP o UDP/IP),
- La dirección IP y
- El número de puerto.

Se necesita agregar el encabezado `using System.Net.Sockets`, para poder usar *sockets*, teniendo esto se puede crear un servidor y el cliente.

4.2.2.1. Servidor

Es el encargado de aceptar las conexiones entrantes y que se utilizarán en el intercambio de datos.

Para crear un servidor con *sockets* se procede de la siguiente manera:

- Crear una variable

```
TcpListener server;
```

- crear el servidor para que escuche en el puerto 5525 (puede ser cualquier puerto que no este en uso por otra aplicación):

```
server = new TcpListener(5525);
```

- Se pone a escuchar en espera de algún cliente

```
server.Start();
```

- Para aceptar el cliente se necesita lo siguiente:

```
client = server.AcceptTcpClient();
```

4.2.2.2. Cliente

Se utiliza un cliente para conectarse con el servidor y establecer una conexión para intercambio de datos.

Para crear un cliente con *sockets* se procede de la siguiente manera:

- Crear una variable adecuada:

```
TcpClient client;
```

- Crear el cliente que establecerá una conexión:

```
client = new TcpClient(201.216.146.61, 5526);
```

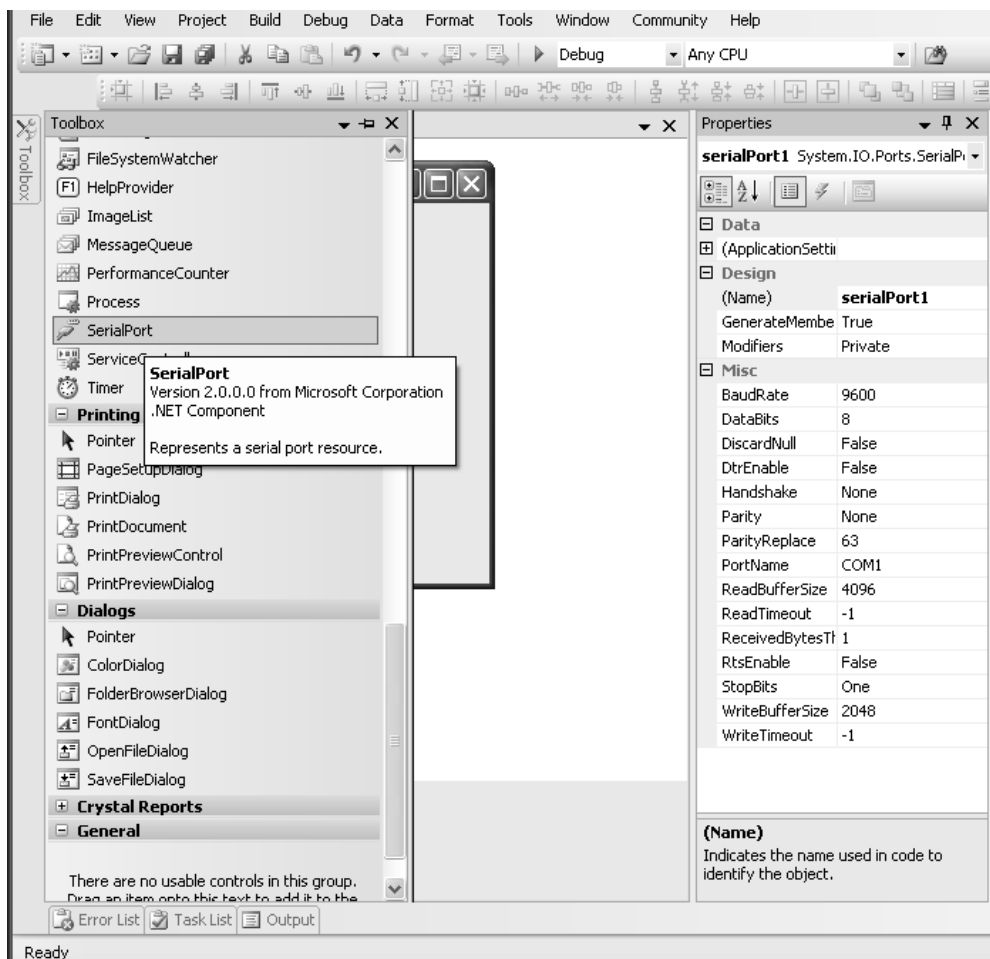
En donde 201.216.146.61 es la dirección pública y 5526 el puerto de la PC a la cual se conectará el cliente.

4.2.3. Puerto serial

Como se describe en el capítulo tres del presente trabajo el puerto serial consta de varias posibilidades de configuración.

En C# es sencillo el manejo del puerto serial, al tener un control ya especializado en el que solo es necesario configurar las propiedades; en la figura 12 se observa en la parte izquierda la localización del control y en la parte derecha se detalla la parte de configuración, sin embargo esta última también se puede hacer dentro del código que el programador escriba, tratando como objeto al puerto serial.

Figura 12 Ubicación del objeto *SerialPort* en *Visual Studio*



Ya que el puerto serial se trata como un objeto, es posible configurarlo tan solo nombrando sus propiedades, además este objeto también posee *eventos* tales como realización de alguna acción al recibir determinada cantidad de datos. La utilización específica se mostrará explicando el código fuente del *software* de redirección.

5. ASPECTOS ECONÓMICOS

Para el análisis económico se considerarán solo los puntos más fundamentales y objetivos de tal manera que se muestre las principales ventajas de implementar el proyecto.

5.1. Licencias

Dado que todo el *software* utilizado en el manejo de las radioayudas posee licencias y por consiguiente al utilizar varias radioayudas se requiere comprar varias licencias para cada computadora que posea *software* de manejo, es una alternativa de ahorro lograr reducir el número de estas licencias.

Utilizando el presente proyecto se podría centralizar el uso de una sola computadora con una licencia de *software* y utilizarla para el manejo de diversas radioayudas.

5.2. Desplazamiento de personal de forma periódica

La movilización de recurso humano desde las instalaciones centrales hasta los puntos en donde se localizan los sitios de las radioayudas, significa un consumo de recurso periódico, a la vez que estos traslados no se pueden realizar tan frecuentemente debido precisamente a que se busca reducir el gasto de traslado.

Un mantenimiento más periódico traería consigo una precisión mayor y un control mas acertado, sin embargo requeriría un gasto mayor por el traslado de personas, no obstante al implementar el presente proyecto, se lograría un control de las radioayudas casi continuo y se ahorrarían recursos al no trasladar personas directamente al sitio sino simplemente los datos necesarios de configuración y mantenimiento.

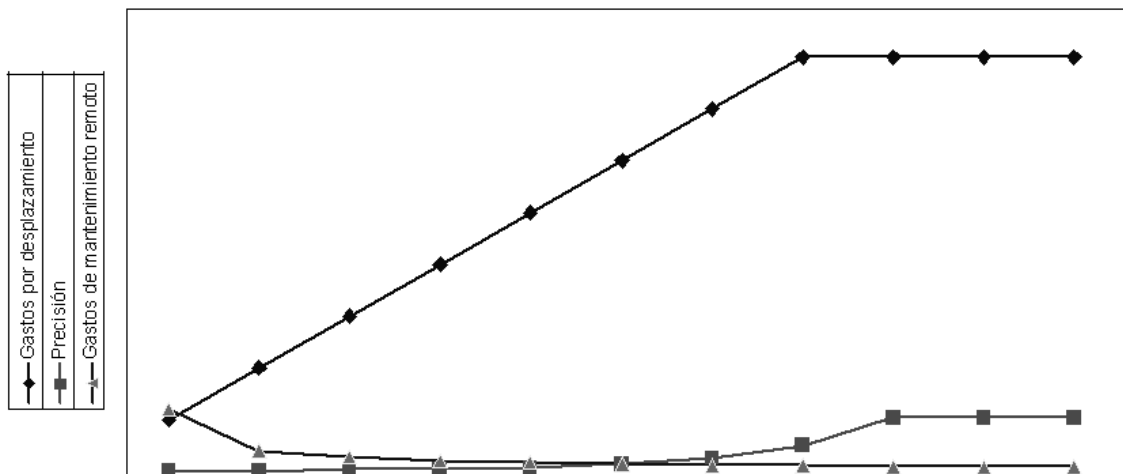
5.2.1. Lugares y períodos de desplazamiento

Actualmente la radioayuda más lejana está ubicada en Puerto Barrios y es la indicada para la implementación del proyecto debido a que por la distancia la forma más económica y rápida de calibración es por medio de mantenimiento remoto, permitiendo así un mantenimiento tan periódico como se necesite, eliminando de esta forma los períodos muy distanciados.

Al hacer un estudio cualitativo de los gastos se observa que se vence la dificultad entre precio-precisión que es realmente el objetivo del proyecto, esto se puede ver en la figura 13, haciéndose notar que conforme se da un

mantenimiento mas continuo, la precisión aumenta y el gasto por calibración, en el caso de desplazamiento personal aumenta y en el caso remoto disminuye.

Figura 13 Gráfica cualitativa de precio de mantenimiento personal, precisión y precio de mantenimiento remoto, contra número de veces que se calibra en un período de tiempo dado.



La explicación de la gráfica de la figura 13 es que los gastos de mantenimiento personal es decir debidos a desplazamiento físico, aumenta entre más veces se desplaza, llegando a estabilizarse cuando se contrata a una persona fija en el sitio, la precisión al dar un mantenimiento más continuo aumenta proporcionalmente al número de veces por período de tiempo no obstante llega un punto en el que la precisión llega al “máximo” (siempre hay probabilidad de error) posible y por último, el gasto del mantenimiento remoto es fijo se use o no, sin embargo entre más se use el precio fijo se distribuye entre el número de veces utilizado, notándose que entre más se use mas barato resulta y más preciso es el sistema.

5.3. Colocación de computadoras dedicadas

Dado que en la actualidad ya se posee una computadora dedicada con el *software* de mantenimiento y monitoreo, solo es necesaria una computadora más en el sitio de las radioayudas, la computadora necesaria solamente tendría que cumplir con los requisitos mínimos que se darán en el siguiente capítulo.

El gasto de la colocación de las computadoras y su mantenimiento se justifica al utilizarlo de forma continua y durante un período dado de tiempo, ya que presenta las ventajas de rapidez y el ahorro continuo de los gastos de traslado de personal, logrando finalmente una absorción de gastos y posteriormente un ahorro continuo.

5.4. Colocación de una red

Como se consideró anteriormente, la red primaria ya está totalmente instalada y solo requiere configuración; la red secundaria consistiría en el contrato de un servicio de *Internet* por medio de vía telefónica con la velocidad que se propone en el apartado de requerimientos, del presente proyecto en el siguiente capítulo, sin embargo esta red se podría utilizar para otros servicios, dando así una distribución de costos entre los diferentes servicios que se utilicen haciendo de esta forma aún mas económico el mantenimiento del equipo.

6. SOFTWARE REDIRECCIÓN DE DATOS

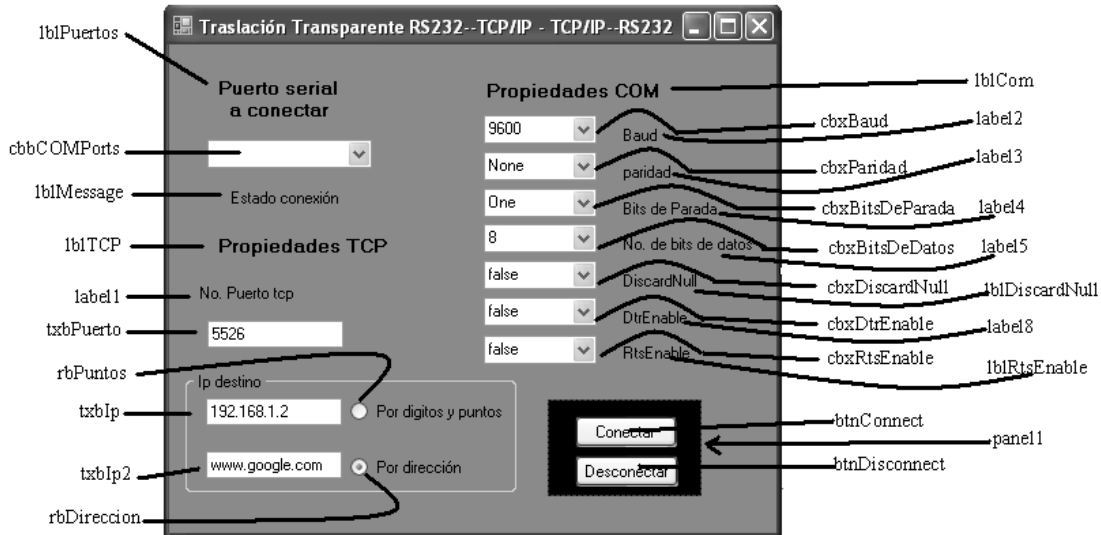
El diseño del *software* de redirección de datos se basó en las necesidades fundamentales de la empresa de redireccionar los datos técnicos y en base a la arquitectura de las redes en donde se planea que funcione.

6.1. Código fuente

Teniendo en cuenta que el IDE utilizado en este proyecto fue Visual Studio, se necesita crear un proyecto nuevo, en este caso se le llamo `pueto_serial_tcp` y en el formulario vacío que se despliega agregar algunos elementos para después introducir el código fuente que se detallará mas adelante.

Se tienen que introducir los elementos necesarios (botones, etiquetas, etc.) para después agregar simplemente el código fuente. Estos elementos tienen que nombrarse tal como aparecen señalados el la figura 14.

Figura 14 Modelo de formulario para utilizar en el proyecto.



Teniendo ya listo el formulario se necesita introducir el siguiente código fuente (se documentó con comentarios para un mayor entendimiento):

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.IO.Ports;
using System.Net;
using System.Net.Sockets;
using System.Threading;

namespace puerto_seiral_tcp
{
    public partial class Form1 : Form
    {
        //se crean variables adecuadas
        TcpListener server;
        TcpClient client;
        IPAddress ipAddress;
        byte[] size = new byte[4];
    }
}
```

```

int recv, packsize;

byte[] data;

public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    //-Muestra todos los puertos seriales disponibles en la pc
    string[] portNames =
System.IO.Ports.SerialPort.GetPortNames();
    for (int i = 0; i <= portNames.Length - 1; i++)
    {
        cbbCOMPorts.Items.Add(portNames[i]);
    }
    btnDisconnect.Enabled = false;
}

//Se activa un servidor que siempre va a recibir datos

private void serividor()
{
    server = new TcpListener(Convert.ToInt32(txbPuerto.Text));

    while ((lblMessage.Text != serialPort.PortName + "
desconectado.") && (lblMessage.Text != "Estado"))
    {
        try
        {
            server.Start();
            client = server.AcceptTcpClient();
            NetworkStream ns = client.GetStream();

//se recibe el tamaño en size en formato byte[]
//se recibe aqui primero y se mira cuantos bytes se reciben
            recv = ns.Read(size, 0, 4);

//se pasa a entero el numero debytes q se van a recibir
            packsize = BitConverter.ToInt32(size, 0);

//se crea la variable del tamaño justo
            data = new byte[packsize];

//se recibe la informacion
            recv = ns.Read(data, 0, packsize);

//la variable ms es para pasar al textbox

```



```

        using (MemoryStream ms = new MemoryStream(data, 0,
packsize))
        {
            ms.Write(data, 0, packsize);
//Se escribe en el puerto si esta abierto (no se ha desconectado)
            if (serialPort.IsOpen)
            {
//se escribe la info
                serialPort.Write(data, 0, packsize);
            }
        }
        ns.Flush();
        ns.Close();
        client.Close();
        server.Stop();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}

//Se activa un cliente que siempre va a estar enviando datos

private void cliente(byte[] caracter)
{
    TcpClient client;
    if (rbPuntos.Checked == true)
    {
//Convierte el texto a formato IP
        ipAddress = IPAddress.Parse(txbIp.Text);
    }
    else
    {
//Resuelve la dirección ip del dominio
        IPEndPoint ipHostInfo = Dns.Resolve(txbIp2.Text);
        ipAddress = ipHostInfo.AddressList[0];
    }

//Establece quien es el cliente al que se le manda
    client = new TcpClient(ipAddress.ToString(),
Convert.ToInt32(txbPuerto.Text.ToString()));
    NetworkStream ns = client.GetStream();

//se adquiere el tamaño de la información
    int size = caracter.Length;

//se crea una variable adecuada
    byte[] packsize = new byte[4];

```

```

//se graba el tamaño de entero a formato byte[]
        packsize = BitConverter.GetBytes(size);//se adquiere el
numero de bytes a transmitir

//se envia cuantos bytes de información se van a transmitir
        ns.Write(packsize, 0, 4);

//se escribe la información
        ns.Write(caracter, 0, size);

//se limpia el flujo y se cierra
        ns.Flush();
        ns.Close();
        client.Close();
    }

    private void abrirCOM(string NoPuerto)
    {

//cierra el puerto serial si esta abierto
        if (serialPort.IsOpen)
        {
            serialPort.Close();
        }
        try
        {
//se configuran varios parametros del puerto
            serialPort.PortName = NoPuerto;

            serialPort.BaudRate =
Convert.ToInt32(cbxBaud.SelectedItem.ToString());

            switch (cbxParidad.SelectedItem.ToString())
            {

                case "Even":
                    serialPort.Parity = Parity.Even;
                    break;
                case "Mark":
                    serialPort.Parity = Parity.Mark;
                    break;
                case "None":
                    serialPort.Parity = Parity.None;
                    break;
                case "Odd":
                    serialPort.Parity = Parity.Odd;
                    break;
                case "Space":
                    serialPort.Parity = Parity.Space;
                    break;
                default:
                    serialPort.Parity = Parity.None;
            }
        }
    }
}

```

```

        break;
    }

    switch (cbxBitsDeParada.SelectedItem.ToString())
    {
        case "None":
            serialPort.StopBits = StopBits.None;
            break;
        case "One":
            serialPort.StopBits = StopBits.One;
            break;
        case "OnePointFive":
            serialPort.StopBits = StopBits.OnePointFive;
            break;
        case "Two":
            serialPort.StopBits = StopBits.Two;
            break;
    }

    serialPort.DataBits =
    Convert.ToInt32(cbxBitsDeDatos.SelectedItem.ToString());

    serialPort.DiscardNull =
    Convert.ToBoolean(cbxDiscardNull.SelectedItem.ToString());

    serialPort.DtrEnable =
    Convert.ToBoolean(cbxDtrEnable.SelectedItem.ToString());

    serialPort.RtsEnable =
    Convert.ToBoolean(cbxRtsEnable.SelectedItem.ToString());

    try
    {
        //se abre el puerto serial
        serialPort.Open();
    }
    catch (IOException ex)
    {
        serialPort.Close();
        MessageBox.Show(ex.Message, "IO Exception:
Reconfigurar propiedades del puerto COM (no se abrio el puerto)");
    }

    //Se actualiza el estado del puerto
    //habilita/deshabilita los bonotnes
    lblMessage.Text = cbbCOMPorts.Text + " conectado.";

    //se desconectan los combobox y el boton de conectar
    btnConnect.Enabled = false;
    cbxDesconectar(); //esta es una funcion definida mas
abajo

    //se habilita el boton desconectar

```

```

        btnDisconnect.Enabled = true;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}

private void cbxDesconectar()
{
    cbbCOMPorts.Enabled = false;
    cbxBaud.Enabled = false;
    cbxBitsDeDatos.Enabled = false;
    cbxBitsDeParada.Enabled = false;
    cbxParidad.Enabled = false;
    txbIp.Enabled = false;
    txbIp2.Enabled = false;
    rbDireccion.Enabled = false;
    rbPuntos.Enabled = false;
    txbPuerto.Enabled = false;
    cbxDiscardNull.Enabled = false;
    cbxDtrEnable.Enabled = false;
    cbxRtsEnable.Enabled = false;
}

private void cbxConectar()
{
    cbbCOMPorts.Enabled = true;
    cbxBaud.Enabled = true;
    cbxBitsDeDatos.Enabled = true;
    cbxBitsDeParada.Enabled = true;
    cbxParidad.Enabled = true;
    txbPuerto.Enabled = true;
    txbIp.Enabled = true;
    txbIp2.Enabled = true;
    rbDireccion.Enabled = true;
    rbPuntos.Enabled = true;
    cbxDiscardNull.Enabled = true;
    cbxDtrEnable.Enabled = true;
    cbxRtsEnable.Enabled = true;
}

void cerrarCOM()
{
    try
    {
        //se cierra el puerto
        serialPort.Close();

//Se actualiza el estado del puerto
//habilita/deshabilita los bonotnes

```

```

        lblMessage.Text = serialPort.PortName + "
desconectado.";
//se habilitan el boton de conectar y los combobox
        btnConnect.Enabled = true;
        cbxConectar();
//se deshabilita el boton desconectar
        btnDisconnect.Enabled = false;
//se crea un cliente para conectarse y asi poder cerrar la aplicacion
correctamente
        TcpClient client;
//Convierte el texto a formato IP

//Resuelve la dirección ip de la misma pc donde se ejecuta
        IPEndPoint ipHostInfo =
        Dns.Resolve(Dns.GetHostName());
        ipAddress = ipHostInfo.AddressList[0];

//Establece quien es el cliente que se conectará
//a la misma pc para desactivar la escucha continua
        client = new TcpClient(ipAddress.ToString(),
Convert.ToInt32(txbPuerto.Text.ToString()));
        client.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}

private void btnConnect_Click_1(object sender, EventArgs e)
{
    abrirCOM(cbbCOMPorts.SelectedItem.ToString());
    Thread primerthread = new Thread(new
ThreadStart(servidor));
    primerthread.Start();
}

private void btnDisconnect_Click(object sender, EventArgs e)
{
    cerrarCOM();
}

private void serialPort_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
//se crea la variable del tamaño necesario
    byte[] datos = new byte[serialPort.BytesToRead];
//se leen los datos
    serialPort.Read(datos, 0, serialPort.BytesToRead);
//se envian los datos

```

```

        cliente(datos);
    }

    private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
    {
        cerrarCOM();
    }
}
}
}

```

6.1.1. Funciones utilizadas

Se utilizaron varias funciones para realizar el proyecto que se describen de una forma general para entender de una manera mas profunda el funcionamiento del código fuente anteriormente expuesto.

6.1.1.1. Carga del formulario

Cuando se carga el formulario con sus componentes el programa ejecuta al inicio la función *private void Form1_Load(object sender, EventArgs e){}*, es aquí donde se aprovecha a colocar el código para encontrar los puertos COM disponibles y colocarlos en un combobox llamado *cbbCOMPorts*, con el código que aparece internamente en esta función y el cual se ejecuta al inicio.

6.1.1.2. El servidor

Es la función que aparece en el código como *private void serividor(){}*, es aquí en donde se crea un servidor, el cual se ejecuta en un *thread* distinto al del formulario y permite estar escuchando sin bloquear la aplicación, no obstante la función consiste en esperar a que se presente un cliente, aceptarlo, leer la información que este le envíe y por último escribir esta información en el puerto serial.

6.1.1.3. El cliente

Es la función *private void cliente(byte[] caracter){}*, la cual crea un cliente cuando se recibe información del puerto COM, logrando así conectarse hacia el servidor y enviarle la información recibida. Recibe un argumento, el cual es la información anteriormente dicha que le envía el puerto serial.

6.1.1.4. Abrir puerto COM

Es con la función *private void abrirCOM(string NoPuerto)* con la que se abre el puerto serial y se le configuran las distintas propiedades, recibe de argumento el nombre del puerto que se manejará, el cual se obtiene de la selección que el usuario escoja.

6.1.1.5. Cerrar puerto COM

La función *private void abrirCOM(string NoPuerto)* ejecuta el cierre y liberación del puerto cuyo nombre se le pasa en la variable *NoPuerto* y el cual se ha redireccionado.

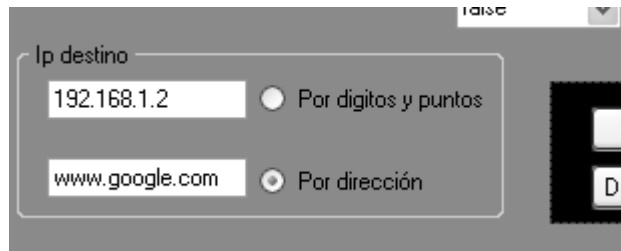
6.1.1.6. Habilitar y deshabilitar opciones de configuración

Las funciones *private void cbxDesconectar()* y *private void cbxConectar()* son las que habilitan y deshabilitan las opciones de configuración al presionar el botón desconectar y conectar respectivamente.

6.1.2. Conexión hacia la red

La conexión con la red se efectúa por medio de la elección de la forma de búsqueda de la PC remota, dado que las dos posibilidades son por medio de una dirección IP en forma decimal o por medio de un dominio que representa la dirección, como se observa en la figura 15, haciendo la selección por medio del *check_box*.

Figura 15 Parte funcional del programa que ejecuta la elección del método de ingreso de IP.



La resolución de la dirección IP por medio del dominio se logra con el bloque de código siguiente:

```
//Resuelve la dirección ip del dominio
IPHostEntry ipHostInfo = Dns.Resolve(txbIp2.Text);
ipAddress = ipHostInfo.AddressList[0];
```

Aquí `txbIp2.Text` es el texto que se encuentra en el textbox dedicado para la escritura del dominio del que se desea averiguar la dirección IP y la variable `ipAddress`, contiene la dirección IP ya resuelta.

6.1.3. Conexión del puerto serial

La elección del puerto que se redireccionará se elige en el elemento funcional que se observa en la figura 16 y la configuración del puerto se realiza en el bloque que se muestra en la figura 17.

Figura 16 Parte funcional del programa en donde se selecciona el puerto a redireccionar.

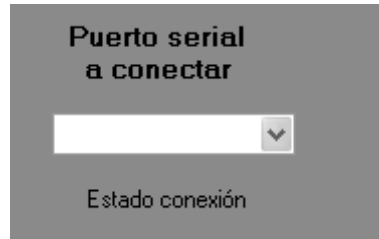


Figura 17 Parte funcional del programa en donde se selecciona la configuración del puerto redireccionado.



La observación importante que es fundamental mencionar es que un puerto se redirecciona de tal manera que no puede ser utilizado por un *software* interno, porque el redireccionamiento es de datos externos y de datos que han entrado por el puerto de red y dirigidos hacia dicho puerto.

6.2. Requisitos del sistema

Se describirá a continuación los requisitos fundamentales que se han de tener en ambas redes para el correcto funcionamiento del proyecto, tomando como anteriormente se hizo a la red primaria la que se encuentra en las oficinas centrales, constando también de la computadora que tiene el *software* de mantenimiento usual, llamado ADRAC y la red secundaria, la que se propone instalar en los sitios de las radioayudas.

6.2.1 Red primaria

Debido a que ya se cuenta con una red interconectada a *Internet* solo es necesario especificar que la velocidad mínima de transmisión es de 56kbps, la cual es superada actualmente.

La computadora que posee el *software* ADRAC deberá además de los requisitos propios de este programa tener las siguientes características para poder utilizar el *software* de redirección:

- a. Cumplir con los requisitos mínimos de Windows XP.
- b. Tener instalado el ServicePack2 de Windows.
- c. Tener instalado el Framework 2.0 o una versión más actual, la cual se puede descargar gratuitamente en la página de Microsoft.

- d. Dos puertos rs232 disponibles para manejo.

6.2.2 Red secundaria

Los requisitos de la PC redireccionadora son casi los mismos que en la de la red secundaria a diferenciándose en que solo es necesario un puerto serial y la velocidad de *Internet* puede ser de 56Kbps o mayor, sin embargo ya que algunos proveedores proporcionan algunas veces servicio de *Internet* que se corta en ocasiones se diseñó el proyecto pensando en que cada dato enviado rectifica la conexión actualizándola en cada uso y salvando esta dificultad.

6.3. Manejo de la aplicación

El manejo implica la correcta configuración de los parámetros y de las acciones siguientes que determinan la conexión adecuada de la capa física.

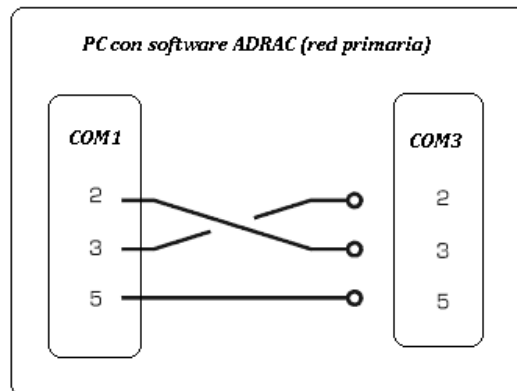
6.3.1. Recursos *hardware* configurables

Dado que las redes de *Internet* brindan un puerto de conexión RJ45 se considerará únicamente lo relacionado a la configuración del cableado hasta llegar a la conexión de *Internet*.

6.3.1.1 Red primaria

En esta red es necesario hacer una interconexión de puertos seriales tal como se observa en la figura 18, esta configuración es necesaria dado que el *software* ADRAC maneja al puerto COM1 y el *software* de redireccionamiento maneja el COM3, esto es así debido a que todo lo que envíe el COM1 lo transmite hacia el COM3 y estos datos entrantes se transmiten hacia la red, de igual forma los datos que provengan de la red se redireccionan hacia el COM3 y este a su vez los transmite hacia el COM1 logrando así una comunicación transparente entre el *software* ADRAC que mira a COM3 como si fuera un puerto serial de alguna radioayuda.

Figura 18 Interconexión de puertos rs232 de una misma PC.

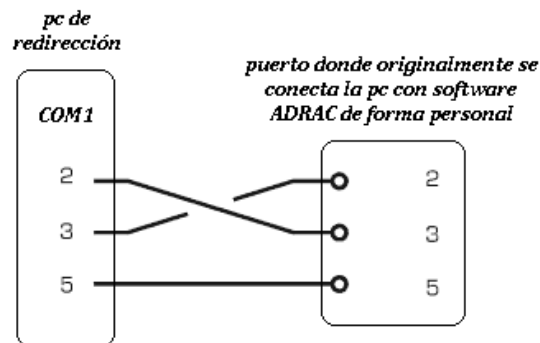


Después de hacer la conexión de puertos como se indicó anteriormente es necesario conectar la PC hacia la red y configurar posteriormente los parámetros del *software* de redirección.

6.31.2 Red secundaria

En esta red lo que sucede es que cuando se recibe algún dato de la red se transmite hacia el puerto RS232 de la PC que puede ser el COM1 u otro y este puerto a la vez lo transmite hacia la radioayuda o al monitor centralizado de radioayudas, y de la misma forma lo que envía la radioayuda lo capta en este caso COM1 y lo transmite hacia la red, la configuración física se puede observar en la figura 19.

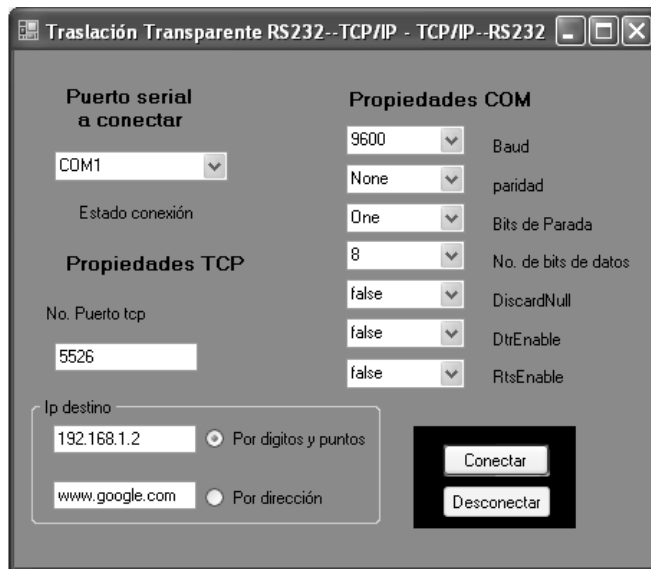
Figura 19 Interconexión de puertos rs232 de la PC de redirección y el puerto remoto.



6.3.2. Configuración específica de parámetros

La configuración se ejecuta de forma fácil al elegir los parámetros correctos en cada opción presentada. Los parámetros se han colocado variables para volver más aplicable el *software* a otras especificaciones si se requieren, la figura 20 muestra la configuración correcta.

Figura 20 Configuración básica de parámetros



Después de establecer todos los parámetros de configuración se presiona el botón de conectar deshabilitando posibles modificaciones y estableciendo el puente de comunicación, si ya no se requiere el uso del *software* solo se requiere presionar el botón desconectar, cerrando las conexiones de red y liberando el puerto RS232 utilizado.

6.4. Descripción general de funcionamiento

El funcionamiento del proyecto se puede enfocar desde el punto de vista interno que abarcaría al *software* de redirección y el externo que abarcaría al conjunto, es decir la interconexión total.

6.4.1 Del *software* de redirección

Se planeó el funcionamiento, debido a las conexiones inestables de *Internet* que se podría cortar la comunicación, que el *software* se conectara cada vez que se establecía un envío de datos logrando así reconexión automática y poder detectar fácilmente el momento de pérdida de conexión si fuera fundamental una transferencia.

Para el manejo de conexiones se utilizan *sockets* y de esta forma se crean los clientes y servidores los cuales coexisten en cada pc, esto ya que cuando se envían datos es un cliente el que los envía y un servidor el que los recibe y viceversa. Dado que en los dos puntos se tiene un cliente y un servidor, se utiliza el mismo *software* de redirección en cada lado.

Un punto importante a considerar del funcionamiento es que cuando se presiona el botón conectar se abre el puerto RS232 elegido y no permitiendo que ninguna otra aplicación lo utilice, reservándose el derecho de manejo de forma individual.

Puede redireccionar cualquier puerto serial con cualquier clase de datos que se envíen en los formatos soportados.

El *software* se realizó de tal manera que si le llegan datos de alguna manera en el formato soportado por sus opciones, sea cual sea el equipo que le envíe los datos, el *software* los transmitirá de forma transparente lográndose así una expansión de funcionamiento no orientada totalmente hacia radioayudas sino a equipos RS232.

Por último al presionar el botón conectar el mismo *software* bloquea las opciones de configuración para indicar cuales son las actuales y no permitir modificaciones “en la marcha” que distorsionen el funcionamiento.

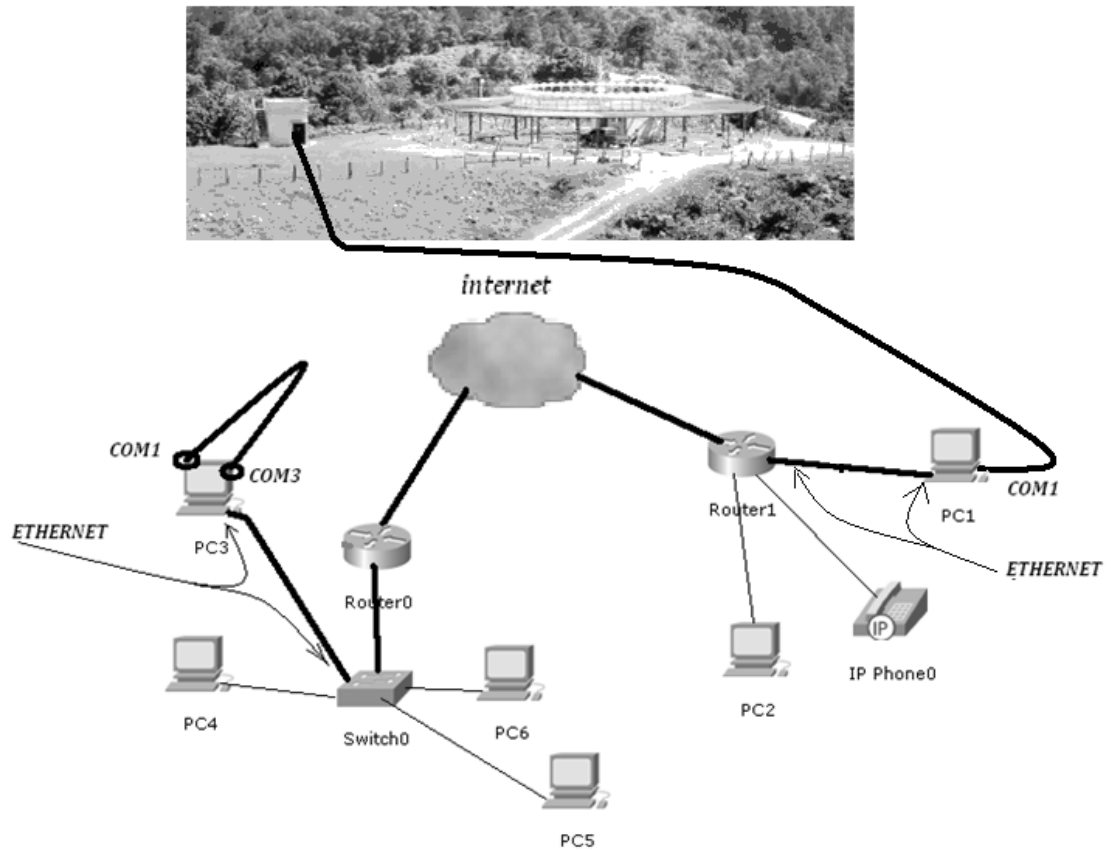
6.4.2 Del proyecto total

Como se mencionó en el capítulo uno, a las radioayudas se les puede dar mantenimiento de dos formas, las cuales de igual manera se pueden hacer de forma remota.

6.4.2.1 En el sitio de la radioayuda

Tomando la configuración inicial del capítulo uno, se logra separar a la computadora con el *software* ADRAC al ejecutar las conexiones adecuadas que se han mencionado anteriormente logrando un diagrama general de interconexión como se muestra en la figura 21, aquí se puede observar que el puerto COM1 y COM3 se interconectan para emular el puerto de alguna radioayuda y hacer transparente el funcionamiento de la red.

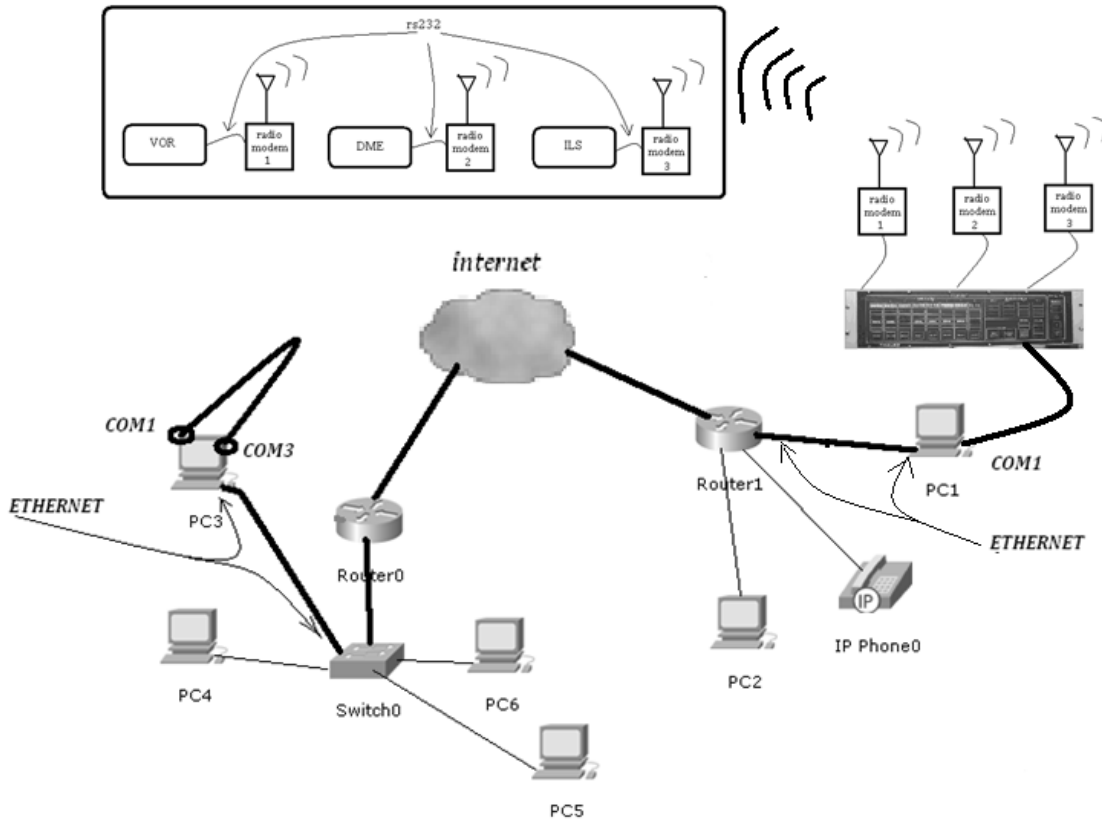
Figura 21 Diagrama general final de redireccionamiento sin RCSU.



6.4.2.1 Con el monitor centralizado

En esta configuración el único cambio es que se centralizan las radioayudas en un monitor y de esta manera se puede dar mantenimiento desde un solo punto pero igualmente de forma remota, en la figura 22 se observa la interconexión.

Figura 22 Diagrama general final de redireccionamiento con RCSU.



7 LÍNEAS DE CONTINUACIÓN

El proyecto se basa fundamentalmente en la redirección y transporte de datos hacia *Internet* como medio de transmisión y dado que la construcción del *software* se hizo de forma modular, se puede ampliar su funcionamiento de diversas formas, permitiendo así un seguimiento hacia la mejora y adaptación a determinados objetivos.

7.1 Capacidad de modificación de *software*

Como se mencionó antes la capacidad de modificación de *software* es posible debido a que se orientó el proyecto a módulos funcionales capaces de modificarse o hacerse compatibles con funciones extra.

Algunos ejemplos:

- Conexión de red con fuentes diversas de datos, permitiendo así no solo redireccionar datos del puerto serial sino datos generados por otros programas y enviarlos por red, lo único que se tendría que hacer es una función que recoja los datos de dichos programas y se comunique con el bloque de conexión de red y así poder transmitirlos de forma transparente a través de *Internet*.

- Capacidad de redirección de datos de puerto serial hacia algún otro puerto o incluso hacia otro puerto serial permitiendo así monitorear el comportamiento de un puerto de forma transparente.

7.1.1 Posibilidad de agregar algoritmos de seguridad de autenticación personalizada

Dado que se utiliza el protocolo orientado a conexión TCP, cada vez que se establece dicha conexión se puede solicitar una autenticación que podría ir variando en función de algún parámetro también variable.

Otra forma de ampliar la seguridad de autenticación es utilizar un algoritmo de codificación y agregarlo a la autenticación de la conexión y lograr así un control de los clientes que se conectan con la aplicación.

7.1.2 Aumento de capacidad hasta el límite requerido agregando características personalizadas

Las características se pueden ampliar al agregar funciones compatibles con las actuales permitiendo así incluso conectar diversas computadoras y distribuir aplicaciones en ellas y enlazarlas en red controlando lo que hace cada computadora.

Una posibilidad de aumentar la capacidad es conectar diversos sensores de forma serial en computadoras alejadas geográficamente y centralizar toda la información en determinado punto al efectuar el redireccionamiento.

7.2 Generalización de diseño para otros usos

El diseño del *software* es bastante general permitiendo no solo conectar radioayudas, sino diversos aparatos que se comuniquen en forma serial para separar controles de instrumentos tanto como el enlace de *Internet* lo permita.

7.2.1 Centralización de poder computacional y ahorro de licencias

Si determinado *software* controlador de instrumentos se vendiera por cada computadora utilizada como control de mando, habría un ahorro en licencias al poder utilizar el *software* centralizado en un punto y de forma remota controlar los diversos instrumentos con una sola licencia y la posibilidad de aumentar la capacidad computacional de la PC que posee el *software* licenciado y las otras computadoras conectadas de forma remota a los instrumentos con tan solo una capacidad de poder computacional pequeña relativamente.

CONCLUSIONES

1. La capacidad de redirección de datos permite evitar desplazamientos físicos periódicos de personal hacia el lugar que necesite mantenimiento y monitoreo.
2. Debido a que las direcciones IP públicas más económicas son dinámicas, se requiere de un dominio que permita conocer la dirección dinámica en cualquier tiempo, por medio de una relación periódica entre dicha dirección y dominio.
3. El uso de *Internet* como medio de transporte resulta económico al aprovecharlo para diferentes usos tales como voz IP.
4. El envío de datos de una red a otra por medio de *Internet* requiere la configuración de los diferentes dispositivos que se conectan a la red.
5. La programación visual permite obtener *software* confiable y complejo de forma fácil.

RECOMENDACIONES

1. La aplicación del proyecto a radioayudas permite un manejo remoto y más periódico, haciéndolas más precisas y confiables, a la vez que reduce costos.
2. El uso de *Internet* para varias funciones en la red secundaria de las radioayudas, permite distribuir costos haciendo aún más económica la aplicación del proyecto.

BIBLIOGRAFÍA

1. Adsuar Mazón, Joaquín C. **Navegación Aérea**. 2ª edición; España: Editorial Paraninfo, 2002.
2. Lammle, Todd **CCNA™ Cisco® Certified Network Associate Study Guide**. 2ª edición; United States of America: Editorial SYBEX Inc., 2000.
3. Lee, Wei-Meng. **Practical .NET 2.0 Networking Projects**. United States of America: Editorial: Springer-Verlag New York, Inc., 2007.

BIBLIOGRAFÍA ELECTRÓNICA

1. Documentación de redes

www.cisco.com

28 de noviembre de 2008

2. NAT

http://es.wikipedia.org/wiki/Network_Address_Translation

13 de noviembre de 2008

3. NAT

<http://www.info-ab.uclm.es/asignaturas/42550/PDFs/PRACTICA5.pdf>

15 de noviembre de 2008

4. NAT

<http://www.monografias.com/trabajos20/traductor-nat/traductor-nat.shtml>

12 de noviembre de 2008

5. Publicar un dominio

www.no-ip.com

28 de noviembre de 2008

6. Sistemas de navegación aérea

http://www.corpac.gob.pe/publica/gnss/links/capitulo_ii.htm

27 de diciembre de 2008

