

**Universidad Don Bosco.  
Facultad de Ingeniería.**



Proyecto de trabajo de graduación para optar por el grado de  
Ingeniero en Telecomunicaciones.

**“Sistema Semiautomático para conmutación y reproducción  
de señales de audio y video”.**

Presentado por:

Gerson Joksan Alvarado Torres.

Asesor:

Ing. Luís Alejandro Martínez Campos.

Lector:

Ing. Calixto Rodríguez.

Soyapango, 2 de Julio de 2008.

**Universidad Don Bosco.  
Facultad de Ingeniería.**



Rector:

Ing. Federico Miguel Huguet Rivera.

Vicerrector académico:

Pbro. Víctor Bermúdez Y.

Secretario general:

Ing. Xiomara Martínez.

Decano de la facultad de ingeniería:

Ing. Ernesto Godofredo Girón.

Julio 2008

El Salvador – Centroamérica.

**Universidad Don Bosco.  
Facultad de Ingeniería.**



Proyecto de trabajo de graduación para optar por el grado de  
Ingeniero en Telecomunicaciones.

**“Sistema Semiautomático para conmutación y reproducción  
de señales de audio y video”.**

Asesor:

Lector:

\_\_\_\_\_  
Ing. Luís Alejandro Martínez Campos.

\_\_\_\_\_  
Ing. Calixto Rodríguez.

Soyapango, 2 de Julio de 2008.

# ÍNDICE.

I. Descripción General del Proyecto.....	1
1.1 Análisis de las alternativas tecnológicas.....	6
1.1.1 Sistema de control remoto.....	6
1.1.2 Sistema de control central.....	9
1.1.3 Sistema de reproducción de video.....	11
1.1.4 Sistema de conmutación de video.....	12
1.1.5 Servicio de enlace de datos.....	12
1.2 Estimación de costos del proyecto.....	13
1.2.1 Sistema de control remoto.....	14
1.2.2 Sistema de control central.....	14
1.2.3 Sistema de reproducción de video.....	14
1.2.4 Sistema de conmutación de video.....	15
1.2.5 Servicio de enlace de datos.....	15
II. Justificación.....	17
III. Objetivos.....	18
IV. Alcances.....	19
V. Limitaciones.....	20
VI. Validación.....	21
VII. Descripción detallada del proyecto.....	22
7.1 Sistema de control de las estaciones remotas.....	22
7.1.1 Sistema de conmutación.....	24
7.1.1.1 Enrutador de audio/video.....	24
7.1.1.2 Circuito del controlador del Enrutador.....	28
7.1.1.3 Protocolo de comunicación entre la PC y el controlador del enrutador.....	33
7.1.1.4 Firmware del controlador del Enrutador.....	39
7.1.2 Software de control de las estaciones remotas.....	47
7.1.2.1 Librería (DLL) de control de hardware.....	49
7.1.2.2 Programa de control de la estación remota (servidor remoto de videos).....	57
7.1.2.3 Protocolo de comunicación entre cliente y servidor vía red de datos.....	63
7.1.2.4 Programa administrativo local (interfaz de usuario manual).....	74

7.2 Sistema de control de la estación local. ....	77
7.2.1 Software de control de la estación local. ....	79
7.2.1.1 Librería (DLL) de conexión para el cliente del reproductor remoto. ....	80
7.2.1.2 Software de control y administración local. ....	86
VIII. Plan de trabajo.....	89
8.1 Tiempo. ....	89
8.2 Recursos materiales. ....	89
IX. Cronograma de actividades. ....	90
X. Bibliografía y Fuentes de Consulta.....	91

## **Índice de Tablas.**

Tabla 1.1 – Alternativas de hardware para el sistema remoto. ....	6
Tabla 1.2 – Alternativas de sistema operativo para el sistema remoto. ....	7
Tabla 1.3 – Alternativas de entorno de desarrollo para el sistema remoto. ....	8
Tabla 1.4 – Alternativas de entorno de desarrollo para el sistema de control central. ....	10
Tabla 1.5 – Alternativas de hardware de reproducción de video. ....	11
Tabla 1.6 – Alternativas de servicio de enlace de datos. ....	13
Tabla 1.7 – Costos del hardware del sistema remoto. ....	14
Tabla 1.8 – Costos del hardware del sistema de control central. ....	14
Tabla 1.9 – Costos del hardware de reproducción de video. ....	14
Tabla 1.10 – Costos del hardware de conmutación de video. ....	15
Tabla 1.11– Costos detallados del servicio de enlace de datos. ....	15
Tabla 1.12– Costos totalizados del servicio de enlace de datos. ....	16
Tabla 7.1 – Tabla de verdad de selección de canal de video en el enrutador AEC- 1VAA. ....	26
Tabla 7.2 – Tabla de verdad de selección de canal de audio en el enrutador AEC- 1VAA. ....	26
Tabla 7.3 – Formato general de la trama de la PC. ....	34
Tabla 7.4 – Contenido de la cabecera enviada por la PC. ....	34
Tabla 7.5 – Formato general de la trama del controlador del enrutador. ....	35

Tabla 7.6 – Contenido de la cabecera enviada por el controlador del enrutador.....	35
Tabla 7.7 – Lista de comandos de la PC.....	36
Tabla 7.8 – Campos de bit del comando de resincronización. ....	36
Tabla 7.9 – Campos de bit del comando de selección de canal.....	37
Tabla 7.10 – Distribución de canales para el comando de selección de canal.....	37
Tabla 7.11 – Lista de respuestas del controlador del enrutador.....	38
Tabla 7.12 – Campos de bit de la respuesta de error de comunicación.....	38
Tabla 7.13 – Campos de bit de la respuesta de selección de canal.....	39
Tabla 7.14 – Funciones exportadas por la librería (DLL) de control de hardware.....	50
Tabla 7.15 – Formato general del comando TCP/IP. ....	64
Tabla 7.16 – Contenido del comando de identificación. ....	65
Tabla 7.17 – Contenido del comando de sincronización. ....	66
Tabla 7.18 – Contenido del comando de envío de un bloque de videos.....	67
Tabla 7.19 – Contenido del descriptor de video. ....	68
Tabla 7.20 – Contenido del comando de notificación de eventos. ....	69
Tabla 7.21 – Códigos de notificación del sistema remoto.....	69
Tabla 7.22 – Estructura de la notificación de la DLL de conexión a servidores remotos.....	82
Tabla 7.23 - Códigos de notificación de la DLL de conexión a servidores remotos.....	83
Tabla 7.24 – Códigos de retorno de las funciones de la DLL de conexión a servidores remotos.....	86

## **Índice de Figuras.**

Figura 1.1 – Diagrama en bloques del sistema completo. ....	2
Figura 1.2 – Distribución Geográfica del Sistema de Transmisión.....	4
Figura 7.1 – Diagrama de conexión entre los equipos de la estación remota.....	22
Figura 7.2 – Distribución de los pines del puerto de control en el AEC-1VAA. ....	25
Figura 7.3 – Distribución de los pines del puerto de control en el VS-611.....	27
Figura 7.4 – Esquema de conexión de control sugerido por Kramer Electronics.....	27
Figura 7.5 – Diagrama electrónico del controlador del enrutador de video. ....	30

Figura 7.6 – Diagrama de flujo principal del programa del controlador del enrutador. ....	40
Figura 7.7 – Diagrama de flujo de la rutina de control de tiempo del enrutador. ....	41
Figura 7.8 – Diagrama de flujo de la rutina de recepción de datos del controlador. ....	42
Figura 7.9 – Diagramas de flujo de las rutinas de manejo de hardware del controlador. ....	43
Figura 7.10 – Diagramas de flujo de las rutinas de manejo de datos del controlador. ....	45
Figura 7.11 – Jerarquía del software y hardware de control remoto. ....	47
Figura 7.12 – Diagrama de flujo de las rutinas de inicialización en la DLL. ....	51
Figura 7.13 – Diagrama de flujo de las rutinas de finalización, lectura de tiempo y adición de videos en cola en la DLL. ....	52
Figura 7.14 – Diagrama de flujo de las rutinas de inicio y paro de la reproducción de videos en la DLL. ....	54
Figura 7.15 – Diagramas de flujo de las rutinas de eliminación de todos los videos en cola y cambio de volumen del decodificador en la DLL. ....	55
Figura 7.16 – Diagrama de flujo de las rutinas de cambio de canal del enrutador y comunicación con el controlador del enrutador en la DLL. ....	56
Figura 7.17 – Ventana principal del software de la estación remota. ....	58
Figura 7.18 – Cuadro de diálogo de opciones de conexión. ....	60
Figura 7.19 – Cuadro de diálogo de selección de directorios. ....	60
Figura 7.20 – Cuadro de diálogo de temporización del sistema. ....	61
Figura 7.21 – Cuadro de diálogo de opciones de audio. ....	62
Figura 7.22 – Algoritmo de Cristian. ....	71
Figura 7.23 – Descripción de los controles de programa administrativo local. ....	74
Figura 7.24 – Cuadro de mensaje que recuerda al usuario que no se ha agregado la señal piloto. ....	75
Figura 7.25 – Ejemplo de la ventana del programa de control local cuando está listo para reproducir. ....	76
Figura 7.26 – Cuadro de diálogo de opciones del programa administrativo local. ....	77
Figura 7.27 – Diagrama en bloques del sistema de control local. ....	77
Figura 7.28 – Jerarquía del software y hardware de la estación de control. ....	80
Figura 7.29 – Interfase del programa administrativo del sitio de control. ....	87

## **I. Descripción General del Proyecto.**

El presente proyecto consiste en el diseño e implementación de un sistema de conmutación de señales de televisión controlado a distancia, el cual permitirá al Canal 12 de televisión nacional transmitir diferentes contenidos de programación en diversos puntos del país.

La intención básica del proyecto consiste en transmitir programación variada y de carácter específico, adaptada a la población de las regiones que se desea abarcar. Esta transmisión de la programación debe ser simultánea y en tiempo real, lo cual quiere decir que el sistema debe permitir que diferente contenido de programación de televisión sea transmitido en diferentes regiones a la vez.

La separación de las zonas o regiones será a nivel de estaciones repetidoras, mismas que ya radican dentro de la infraestructura del canal. El diagrama en bloques de la siguiente página muestra a grandes rasgos los bloques principales que componen el sistema. Por motivos de espacio se muestran los elementos que estarían presentes en sólo una estación repetidora, entendiéndose que para las otras repetidoras se reutilizará esta misma estructura. Todo el sistema estará comunicado entre sí a través de una red de datos, la cual será provista por terceros. Las estaciones se encuentran en regiones altas como cerros y volcanes por motivos de lograr la mejor transmisión, sin embargo, estas zonas son de difícil acceso, por lo que el servicio de enlace de datos llegaría a las repetidoras a través de enlaces satelitales, o bien por otros medios que pudieran estar disponibles.

# Sistema de conmutación de televisión

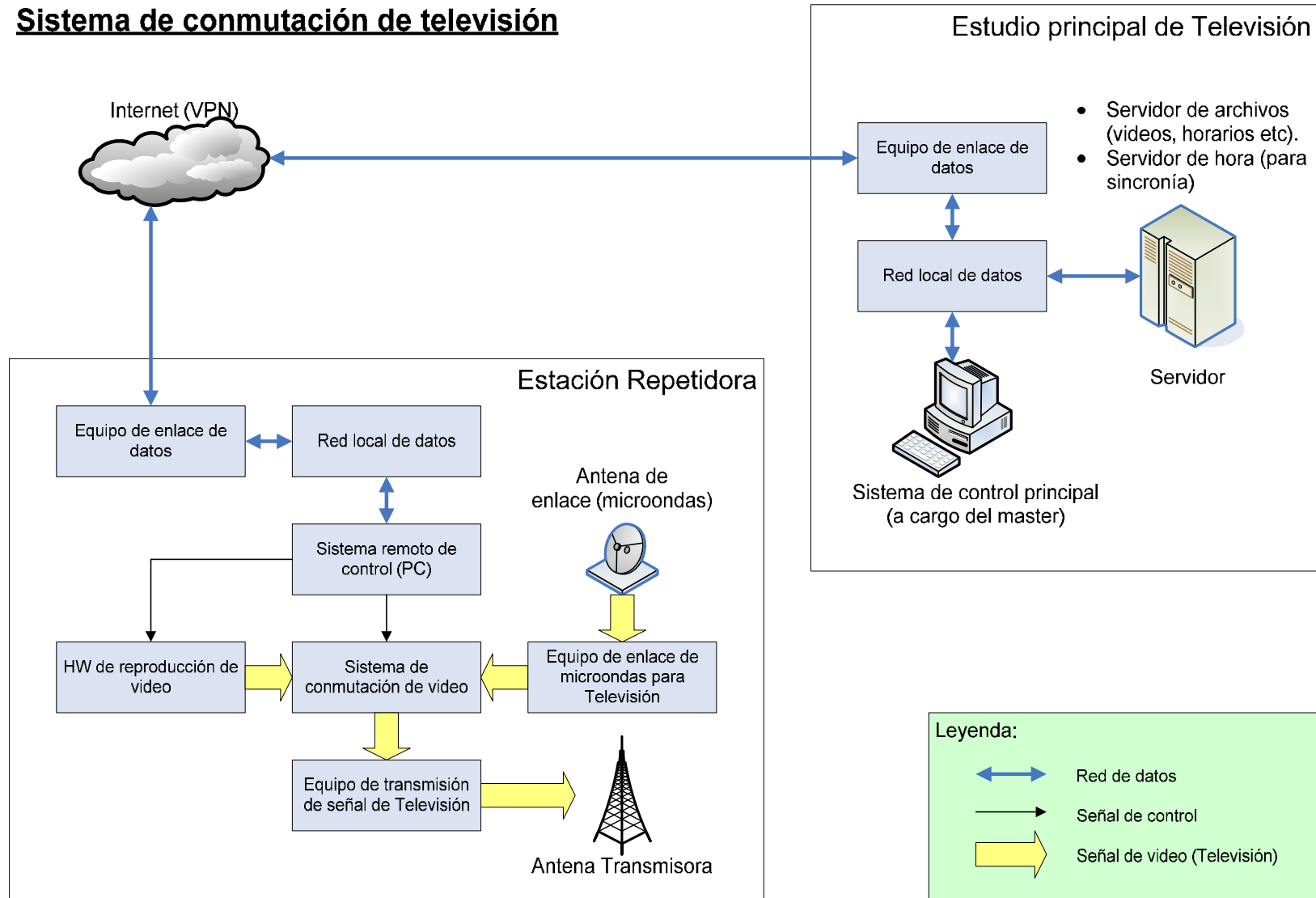


Figura 1.1 – Diagrama en bloques del sistema completo.

La infraestructura actual del canal comprende lo que es equipos de enlace de microondas para la transferencia de la señal de televisión a las repetidoras. Dichos enlaces son de carácter analógico, y actualmente manejan una señal única que el canal transmite a nivel nacional. El contenido de dicha señal proviene directamente de la cabina de control o master<sup>1</sup>, que está a cargo del operador del master. Esta señal piloto es de carácter fundamental, pues la mayor parte del contenido de la programación provendrá directamente de la misma una vez se implemente el proyecto, siendo bloques menores de programación televisiva (como anuncios comerciales) los que serán transmitidos de forma diferenciada a cada región.

Actualmente el Canal 12 posee tres estaciones de transmisión, una principal y dos repetidoras. El transmisor principal está ubicado en el boquerón, y provee cobertura a la capital y la zona central de país. Una de las repetidoras está ubicada en el cerro Cachío, ubicado entre Sonsonate y Ahuachapán, su cobertura abarca la zona occidental, mientras que la otra repetidora se encuentra en el cerro El Pacayal, en San Miguel, proveyendo cobertura a la zona Oriental del país.

La señal piloto del canal, la cual se genera en el estudio principal en Antigua Guatemala se envía vía microondas al transmisor principal en el Boquerón. La repetidora en El Pacayal recibe su señal piloto desde un transmisor de microonda ubicado en el Boquerón, mientras que la repetidora en cerro Cachío obtiene su señal piloto por microonda desde una estación de enlace ubicada en Sonsonate, la cual a su vez obtiene la señal de televisión de la señal al aire, proveniente del transmisor principal en el Boquerón. En la siguiente página se incluye el mapa con la distribución geográfica de todos los elementos del sistema de transmisión.

Naturalmente la infraestructura actual no es suficiente para manejar contenido diferenciado de programación, por lo que la propuesta de proyecto consiste en hacer una ampliación al sistema actual, aprovechando los recursos existentes para cumplir dicho cometido.

---

<sup>1</sup> Como master se define a la ubicación donde convergen todas las fuentes de video con que cuenta el canal, de las cuales se elige la señal a transmitir al aire.

# Distribución Geográfica Del Sistema De Transmisión



Figura 1.2 – Distribución Geográfica del Sistema de Transmisión.

Luego, la mayor parte de la ampliación será al nivel de las repetidoras, las cuales serán dotadas de nuevo equipo. Obviamente se harían algunos cambios en el estudio principal, pero estos consisten mayormente en la adición de un sistema de control y monitoreo remoto, por medio del cual el operador del master podrá dirigir de manera semiautomática la conmutación de las señales y la reproducción de videos. La señal con la programación diferenciada que será transmitida por las repetidoras se generará localmente, por medio de equipos de reproducción de video digital que albergarían las computadoras locales, los cuales podrán pasar grabaciones con los contenidos locales de forma automatizada.

El sistema será gestionado tal como se describe a continuación:

- El contenido de programación local para cada región será generado, grabado o adquirido por el personal del canal en el estudio principal. Los videos resultantes serán trasladados a las repetidoras por medios digitales, usando para ello la red de datos. Los videos serán transferidos a las computadoras en las repetidoras con anticipación, pudiendo usar para ello las horas menos congestionadas de tráfico en Internet.
- Asimismo, tras la adición de los videos al sistema de almacenamiento de las estaciones remotas, se elaborará una pauta<sup>2</sup>, esto es, la planificación de los momentos en que se espera que transmita ya sea la señal piloto o el material grabado. El sistema contará con un programa administrativo para dicho fin.
- Cada repetidora pasará de manera predeterminada el contenido de la señal piloto. En los instantes en que se deba pasar un video local, cada repetidora hará el cambio a la señal grabada en forma semiautomática. El sistema de conmutación remoto estará a cargo del operador del master, quien indicará los momentos exactos en que se deba transmitir ya sea la señal piloto o bien los videos grabados.

---

<sup>2</sup> Al proceso de elaboración de pautas televisivas se le conoce informalmente como “pautaje”, y consiste en hacer una tabla detallada con la secuencia y hora a la que se transmiten los diferentes segmentos televisivos (programas, comerciales, etc.).

## **1.1 Análisis de las alternativas tecnológicas.**

Para el análisis de estas alternativas, se consideran actualmente los siguientes componentes del sistema:

- Sistema de control remoto (PC).
- Sistema de control central (PC Servidor).
- Sistema de reproducción de video (tarjetas decodificadoras de video).
- Sistema de conmutación de video.
- Servicio de enlace de datos.

### **1.1.1 Sistema de control remoto.**

Este componente se encargará del control en las estaciones repetidoras. Esta función será desarrollada principalmente por computadoras, que podrán operar en forma remota por medio de la utilización de software de control. Las alternativas consideradas para el proyecto se destacan a continuación:

#### **Alternativas de Hardware.**

<b>Arquitectura de hardware.</b>	<b>Ventajas.</b>	<b>Desventajas</b>	<b>Uso factible</b>
Plataforma Intel Core Duo de 32 bits.	Sistema relativamente barato y ampliamente utilizado. Desempeño moderado.	Ninguna.	Si.
Plataforma Intel Core Duo de 64 bits.	Sistema caracterizado por su alta velocidad de procesamiento para datos. Sin embargo su rendimiento supera grandemente las necesidades del proyecto.	Precio moderadamente alto.	Si
Plataforma AMD de 64 bits.	Sistema caracterizado por su alta eficiencia en el manejo de recursos. Ideal para aplicaciones de control simultaneo de varios procesos. Su rendimiento también supera las necesidades del proyecto.	Precio elevado.	Si

**Tabla 1.1 – Alternativas de hardware para el sistema remoto.**

Otros requisitos mínimos adicionales e independientes de la plataforma:

- Memoria principal (RAM): 512MB.
- Disco duro: Arriba de 300GB (para almacenamiento de videos).
- Tarjeta de video: 64MB de VRAM y soporte de DirectX 9.0.

La solución más recomendada, ya que cumple con los requerimientos tecnológicos del proyecto es la plataforma Intel Core Duo de 32 bits, pues tiene un rendimiento adecuado.

### Alternativas de software.

El sistema operativo no solo gestiona los recursos del sistema, sino además determina el rendimiento global del mismo por la manera en que está diseñado. A continuación se detallan las alternativas consideradas bajo esta categoría:

Sistema operativo	Ventajas	Desventajas	Uso factible
Plataforma Microsoft Windows XP (32 o 64 bits).	Ampliamente conocido y usado. Compatibilidad casi asegurada con gran cantidad de tecnologías de hardware y software. Amplia disponibilidad de herramientas de desarrollo.	Conocido por su relativa inestabilidad cuando se configura inadecuadamente.	Si.
Plataforma Microsoft Windows Vista (32 o 64 bits).	El más moderno sistema de Microsoft. Incorpora las tecnologías multimedia más recientes.	Requerimientos elevados de hardware, ineficiente uso de los recursos de Hardware. Su relativa novedad lo hace incompatible con algunas tecnologías de hardware y software anteriores.	Si.
Plataforma Linux (32 o 64 bits).	Plataforma eficiente, estable y muy robusta. Ideal para casi todo tipo de aplicaciones profesionales por su elevada eficacia. Precio gratuito en la mayoría de distribuciones.	Es incompatible con varios modelos de hardware de reproducción de video, debido a que varios no cuentan con controladores para estos sistemas. El tiempo de desarrollo de aplicaciones es prolongado debido a su complejidad.	No.

**Tabla 1.2 – Alternativas de sistema operativo para el sistema remoto.**

El sistema operativo recomendado es Windows XP, pues asegura la compatibilidad de todos los demás elementos de hardware y software a utilizar.

El entorno de desarrollo comprende los lenguajes de programación a utilizar así como las librerías y herramientas disponibles para elaborar las aplicaciones a utilizar en el proyecto. El desarrollo de software aplicativo es un punto crucial para el desarrollo del proyecto, y consiguientemente se elegirán las herramientas con cuidado. Bajo este ámbito se consideran los siguientes entornos:

<b>Entorno de desarrollo</b>	<b>Ventajas</b>	<b>Desventajas</b>	<b>Uso factible</b>
Visual C++ 6.0.	Plataforma para desarrollo de aplicativos usada por mucho tiempo y aun muy eficiente. Permite acceso a todos los recursos del sistema sin restricciones.	Relativamente complejo para algunas tareas que a menudo son sólo posibles con él, lo cual a veces prolonga el tiempo de desarrollo. Ya esta desfasado, mas no obsoleto.	Si.
Visual C++ .NET	Una de las más potentes herramientas de desarrollo para la plataforma Microsoft. Provee todas las funciones de su predecesor más muchas nuevas características.	Su complejidad es aun elevada.	Si.
Visual Basic 6.0	Un entorno amigable y ampliamente conocido. Tiempo de desarrollo sorprendentemente corto.	Su alto nivel no solo dificulta, sino a menudo imposibilita la implementación de aplicaciones relacionadas al sistema.	Si.
Java (JDK)	Provee mecanismos para programación independiente de la plataforma así como acceso innato a las tecnologías de Internet. Ideal para manejar bases de datos y conectividad remota.	Desempeño ligeramente afectado por la maquina virtual en que se ejecuta.	Si.
Visual C# .NET	Reduce ampliamente el tiempo de desarrollo a comparación de su antecesor C++. Prácticamente provee la misma funcionalidad necesaria para el desarrollo del proyecto que Visual C++. Similar a Java en varios aspectos.	Todos los aplicativos generados con este lenguaje hacen uso de la infraestructura .NET, la cual presenta una ligera desventaja de rendimiento.	Si.

**Tabla 1.3 – Alternativas de entorno de desarrollo para el sistema remoto.**

En este caso se propone una combinación de dos plataformas de desarrollo. Visual C++ permite la programación del sistema a un nivel relativamente bajo, permitiendo el manejo directo del hardware y acceso a los recursos del sistema operativo, por lo que se recomienda para este tipo de tareas. Visual Basic permite el desarrollo rápido y eficiente de interfases de usuario, por lo que el diseño de los aplicativos administrativos puede resolverse con relativa facilidad y se recomienda también.

### **1.1.2 Sistema de control central.**

Esta parte del sistema proveerá una función de organización dentro del sistema. Las funciones que deberá efectuar son las siguientes:

- Proveer una interfase mediante la cual un operador podrá gestionar el sistema remoto en las estaciones repetidoras.
- Albergar una base de datos con la pauta de la programación televisiva.
- Albergar una base de datos con la información de los archivos de video.

### **Alternativas de Hardware.**

Las alternativas de este sistema son las mismas que la del sistema de control remoto, y los requisitos mínimos son también muy similares, por lo que el sistema recomendado estará basado también en la plataforma Intel Dual Core de 32 bits. Este sistema contendrá una base de datos con la información de los archivos de video, mas no los videos en sí, por lo que sus requerimientos de disco duro podrán ser menores.

### **Alternativas de Software.**

Sistema operativo: Las alternativas son también las mismas que para el sistema de control remoto. El sistema operativo recomendado es también Windows XP por razones similares de compatibilidad.

Las alternativas para el software de desarrollo se destacan a continuación:

Entorno de desarrollo	Ventajas	Desventajas	Uso factible
Visual C++ 6.0.	Plataforma para desarrollo de aplicativos usada por mucho tiempo y aun muy eficiente. Permite acceso a todos los recursos del sistema sin restricciones.	Relativamente complejo para algunas tareas que a menudo son sólo posibles con él, lo cual a veces prolonga el tiempo de desarrollo. Ya esta desfasado, mas no obsoleto.	Si.
Visual C++ .NET	Una de las más potentes herramientas de desarrollo para la plataforma Microsoft. Provee todas las funciones de su predecesor más muchas nuevas características.	Su complejidad es aun elevada.	Si.
Visual Basic 6.0	Un entorno amigable y ampliamente conocido. Tiempo de desarrollo sorprendentemente corto.	Su alto nivel no solo dificulta, sino a menudo imposibilita la implementación de aplicaciones relacionadas al sistema.	Si.
Java (JDK)	Provee mecanismos para programación independiente de la plataforma así como acceso innato a las tecnologías de Internet. Ideal para manejar bases de datos y conectividad remota.	Desempeño ligeramente afectado por la maquina virtual en que se ejecuta.	Si.
Visual C# .NET	Reduce ampliamente el tiempo de desarrollo a comparación de su antecesor C++. Prácticamente provee la misma funcionalidad necesaria para el desarrollo del proyecto que Visual C++. Similar a Java en varios aspectos.	Todos los aplicativos generados con este lenguaje hacen uso de la infraestructura .NET, la cual presenta una ligera desventaja de rendimiento.	Si.
Microsoft Access	Desarrollo rápido de las bases de datos.	Bajo rendimiento para bases de datos grandes.	No.
MySQL	Brinda mayor seguridad, Permite conexiones directas en red. Alto desempeño con bases de datos grandes.	Ninguna.	Si.

**Tabla 1.4 – Alternativas de entorno de desarrollo para el sistema de control central.**

Para este caso se propone una combinación de dos alternativas: Por un lado Visual Basic facilita el acceso a bases de datos y el desarrollo de interfases gráficas, por lo que es ideal para el sistema de control. Por otra parte Access es un gestor de bases de datos de rápida

instalación y compatible con gran cantidad de plataformas, por lo que se recomienda como servidor de bases de datos para la administración del sistema de almacenamiento de videos y pautas televisivas.

### **1.1.3 Sistema de reproducción de video.**

En este rubro se identifican dos categorías principales de alternativas: las tarjetas de video aceleradas, las cuales deben poseer entrada de sincronía (Genlock<sup>3</sup>) y las decodificadoras de video especializadas. La tabla a continuación detalla las alternativas disponibles:

<b>Tipo de dispositivo de reproducción.</b>	<b>Interfase de software a usar.</b>	<b>Ventajas</b>	<b>Desventajas</b>	<b>Uso factible</b>
Tarjeta de video nVidia, con entrada de señal para Genlock.	Direct Show y filtros de codec.	Tecnología usada ampliamente en varias aplicaciones multimedia.	Costo elevado debido a que requiere hardware adicional. Requiere un CPU ligeramente más potente al usar codecs por software.	Si.
Decodificadora especializada para video MPEG2.	SDK <sup>4</sup> propietario provisto por el fabricante.	Uso completo del hardware en cuestión. Eficiencia garantizada y corto tiempo de desarrollo.	Precio mas elevado en caso que el SDK se venda por separado.	Si.
	Direct Show (para administrarla solamente).	Es más barato al no tener que adquirir el SDK en caso que se venda separadamente.	Se desaprovechan algunas características del hardware al tener que hacer uso de una API generalizada.	Si.

**Tabla 1.5 – Alternativas de hardware de reproducción de video.**

<sup>3</sup> “Genlock” es un término técnico que indica que el equipo tiene la capacidad de sincronizar la señal de video que maneja cuadro por cuadro. En el caso del equipo de reproducción de video, significa que la señal se genera en concordancia a una señal de patrón de entrada. Esto es necesario cuando se interconectan múltiples equipos de video en un mismo sistema para asegurar transiciones adecuadas durante la conmutación.

<sup>4</sup> SDK va por “Source Development Kit” y se refiere al software de desarrollo de aplicaciones para una determinada plataforma.

Por motivos de eficiencia, el hardware más adecuado para la reproducción de videos es una tarjeta decodificadora especializada. En concordancia con lo anterior, el SDK provisto por el fabricante es naturalmente la interfase de software más idónea. La recomendación en cuanto a que interfase de software se usará se hará en base a los costos y alternativas disponibles, las cuales serán descritas posteriormente.

#### **1.1.4 Sistema de conmutación de video.**

En este caso, la única tecnología disponible son los conmutadores o enrutadores de audio/video profesionales. La capacidad que deben proveer estos equipos son las siguientes:

- Debe contar con al menos 3 canales de entrada para audio y video.
- La conmutación de las señales de video debe tener sincronía por cuadros (Genlock)
- Debe permitir la interconexión con una PC para permitir el manejo desde la misma.

La recomendación para este tipo de equipos se hará en cuanto a disponibilidad y costos.

#### **1.1.5 Servicio de enlace de datos.**

La interconexión de las computadoras entre los sitios remotos y el estudio de televisión se hará por medio de una tecnología de enlace de datos, pudiendo ser a través del servicio de acceso a Internet o del servicio de un enlace dedicado de datos. El aspecto más importante a destacar es la tecnología de acceso (última milla), pues ésta determina mayormente la velocidad y calidad del servicio, así como su disponibilidad y costo. Entre las alternativas de tecnologías de acceso tenemos las siguientes:

Tecnología de acceso		Ventajas	Desventajas	Uso factible.
Enlace de datos terreno.	Fibra óptica.	Alta velocidad, baja latencia.	Lamentablemente la lejanía entre las estaciones repetidoras y cualquier zona poblada hacen que estos servicios no estén disponibles en Cerro Cachío y El Pacayal.	No.
	Cable (ISDN).	Velocidad moderada, latencia aceptable. Bajo costo.		
	Línea telefónica (DSL).	Velocidad intermedia a alta, latencia relativamente baja. Bajo costo.		
Enlace de datos satelital.		Disponible en casi cualquier parte del mundo.	Costo elevado. A menudo la velocidad es baja, y la calidad del servicio depende mucho de factores climáticos y atmosféricos.	Si.

**Tabla 1.6 – Alternativas de servicio de enlace de datos.**

En este caso se recomienda el enlace de datos satelital al ser la única tecnología disponible en los sitios remotos.

## **1.2 Estimación de costos del proyecto.**

Los costos del proyecto se estimarán también bajo los mismos rubros que el apartado anterior, y son los siguientes:

- Sistema de control remoto (PC).
- Sistema de control central (PC Servidor).
- Sistema de reproducción de video (tarjetas decodificadoras de video).
- Sistema de conmutación de video.
- Servicio de enlace de datos.

### **1.2.1 Sistema de control remoto.**

<b>Sistema de cómputo.</b>	<b>Precio</b>	<b>Comentarios</b>
Computadora basada en Intel Core Duo.	\$2,000.00 (aproximado)	512MB de RAM, Disco duro SATA de 1TB.

**Tabla 1.7 – Costos del hardware del sistema remoto.**

### **1.2.2 Sistema de control central.**

<b>Sistema de cómputo.</b>	<b>Precio</b>	<b>Comentarios</b>
Computadora basada en Intel Core Duo.	\$1,500.00 (aproximado)	512MB de RAM, Disco duro SATA 80GB.

**Tabla 1.8 – Costos del hardware del sistema de control central.**

### **1.2.3 Sistema de reproducción de video.**

<b>Marca y modelo de tarjeta decodificadora</b>	<b>Precio del hardware.</b>	<b>Precio del software de desarrollo (SDK).</b>	<b>Comentarios.</b>
Focus Enhancements Harmony 2SE	\$800.00	Viene incluido con la compra del producto.	Posee dos canales de salida. No ha sido posible a la fecha contactar al distribuidor para los detalles.
Stradis SDM250	No disponible.	\$2,495.00	Ya se cuenta con la tarjeta, mas no se tiene el SDK. Puede prescindirse del SDK usando Direct Show. Producto descontinuado.
Stradis SDM260e	\$1,495.00	\$2,495.00	Modelo más moderno que el SDM250.

**Tabla 1.9 – Costos del hardware de reproducción de video.**

En este caso se recomienda usar la tarjeta decodificadora Stradis modelo SDM260e, ya que es de una marca que la empresa considera confiable y tiene un rendimiento y calidad comprobados.

### **1.2.4 Sistema de conmutación de video.**

<b>Marca y modelo del enrutador de video.</b>	<b>Costo del equipo.</b>	<b>Comentarios.</b>
Adrienne Electronics Corporation AEC-1VAA	\$1145.00	Posee 10 canales de entrada para audio/video. La sincronía de señal se obtiene a partir de la señal de entrada solamente. El modelo está actualmente en vías de ser discontinuado.
Kramer Electronics VS-611	\$1000.00	Posee 6 canales de entrada para audio/video. La sincronía puede ser por una referencia externa u obtenida a partir de una señal de entrada.

**Tabla 1.10 – Costos del hardware de conmutación de video.**

En este caso se recomienda el modelo VS-611 de Kramer Electronics, se trata de un sistema moderno, con más prestaciones y más cómodo.

### **1.2.5 Servicio de enlace de datos.**

<b>Proveedor del servicio de enlace de datos satelital.</b>	<b>Tipo de servicio y velocidad.</b>	<b>Costo de instalación de un enlace.</b>	<b>Costo mensual de un enlace.</b>	<b>Costo del equipo para un enlace (MODEM satelital)</b>	<b>Tiempo promedio para transferir un video<sup>5</sup>.</b>
SIPROSAT	Enlace de Internet 512Kbps garantizado <sup>6</sup> .	\$375.00	\$1990.00	\$1000.00	26 minutos.
SIPROSAT	Enlace de Internet de 128Kbps.	\$1200.00	\$500.00	Incluido con la instalación.	1 hora con 44 minutos.

**Tabla 1.11– Costos detallados del servicio de enlace de datos.**

<sup>5</sup> Estimación aproximada asumiendo un peso promedio de video de 80MB (para anuncios comerciales y promocionales solamente), y un aprovechamiento del 80% del ancho de banda contratado (para representar el peso del protocolo y pérdidas aleatorias con un 20%).

<sup>6</sup> El término “enlace de velocidad garantizada” significa que el servicio se presta por un canal dedicado o reservado sólo para el cliente (y no a través de un canal compartido).

Es de hacer notar que el costo es para cada enlace. Se requieren dos enlaces para los dos sitios remotos (Cerro Cachío y Cerro El Pacayal). Por lo que los costos totalizados son los siguientes:

<b>Tipo de servicio y velocidad.</b>	<b>Costo de instalación de los dos enlaces.</b>	<b>Costo mensual de los dos enlaces</b>	<b>Costo del equipo para dos enlaces (MODEM).</b>
Enlace de Internet 512Kbps garantizado.	\$750.00	\$3980.00	\$2000.00
Enlace de Internet de 128Kbps.	\$2400.00	\$1000.00	Incluido con la instalación.

**Tabla 1.12– Costos totalizados del servicio de enlace de datos.**

El enlace de 128Kbps resulta ser el más barato, sin embargo el tiempo que toma transferir cada video puede llegar a ser muy prolongado y eso podrá afectar seriamente la productividad del sistema al tener que transferirse varios videos. Por tanto, se recomienda el enlace de 512Kbps el cual permitirá una mejor productividad.

## **II. Justificación.**

Los medios televisivos presentan un carácter altamente dinámico, este es un ámbito donde los medios necesitan evolucionar y renovar su imagen continuamente para mantenerse competitivos. En este sentido el proyecto propuesto viene a formar parte de dicha evolución, pretendiendo ampliar la infraestructura actual con la que cuenta el Canal 12 de televisión para que éste preste un servicio todavía más adecuado a la población de las diversas zonas del país.

La infraestructura actual del canal permite la transmisión de una única señal piloto a nivel nacional. Esta forma de operar ha sido suficiente en fechas anteriores, pero hoy por hoy el país ha cambiado y crecido, y hace falta cada vez más mejorar la influencia que el contenido de la programación ejerce sobre los diversos sectores.

Por tanto, el presente proyecto plantea una propuesta de ampliación a la infraestructura del canal, incluyendo elementos de hardware y software para su cometido. Todo ello para ayudar a mejorar los servicios que ya presta. De esta forma el proyecto provee una función de beneficio social, ayudando al sector de la empresa privada.

### **III. Objetivos.**

- Objetivo general.  
Diseñar y crear un sistema remoto de reproducción y conmutación de señales de video.
  
- Objetivos específicos.
  1. Hacer uso de la infraestructura existente en el canal para la implementación del proyecto y ampliar dicha infraestructura.
  2. Elaborar un software capaz de manejar en forma conjunta y eficiente los sistemas de video remotos desde una sola localidad central.
  3. Elegir e implementar los componentes de hardware computacional y electrónicos necesarios para el desarrollo de una solución adecuada a las necesidades del canal.
  4. Elegir y utilizar las herramientas de software adecuadas para el desarrollo de las aplicaciones que se van a necesitar.

## **IV. Alcances.**

- Elaborar un sistema que pueda trabajar en tiempo real con varios contenidos de programación televisiva simultáneos en diferentes puntos.
- El sistema deberá integrar varios elementos de hardware y software a fin de automatizar lo más posible la tarea de conmutación de las señales.
- Asimismo, el sistema deberá tener mecanismos para verificar que opera correctamente en los puntos remotos.
- Debido a la naturaleza con que se realiza el manejo del contenido de programación en el canal, el sistema deberá también ser capaz de eliminar o bien insertar contenido de manera no planeada. Esto debe ser posible incluso minutos antes de que la programación deba ser transmitida por la repetidora.

## **V. Limitaciones.**

- El sistema será semiautomático, lo que significa que estará a cargo de una persona en particular. Bajo la logística actual de operación del canal no es posible implementar un sistema completamente automático al 100%, pues el contenido televisivo es a menudo generado instantes antes de ser transmitido, especialmente con los programas transmitidos en vivo. En este sentido se automatizará el sistema lo más posible.
- En caso de que la red de datos falle, el sistema dejaría de ser operado en tiempo real. Se esperaría que el sistema pase de manera completamente automática a transmitir la señal piloto al dejar de recibir instrucciones desde la cabina principal del master.
- El funcionamiento del sistema de conmutación se verá desligado de los posibles problemas de interferencias mutuas que pudieran darse entre repetidoras (pues transmiten contenido diferente a la misma frecuencia), ya que estos, de existir, deberán ser resueltos por la sección de ingeniería de transmisión de la empresa.
- Asimismo el proyecto, al proveer un nuevo mecanismo que permite generar nuevos productos (siendo el producto en sí la programación televisiva), tiene un impacto significativo a nivel administrativo, pues será necesario implementar nuevos procesos de gestión y actualizar algunos procesos ya existentes. Sin embargo en este sentido se planteará una limitante importante, pues la propuesta actual del proyecto no pretende abarcar los aspectos administrativos con profundidad. En este sentido será el personal administrativo de Canal 12 quienes se encarguen de gestionar dichos cambios.
- El proyecto se llevará a cabo bajo la supervisión directa del personal técnico y administrativo del canal, y por tanto estará sometido a sus decisiones. Bajo esta perspectiva puede haber limitantes adicionales que ellos dispongan y que no se contemplan en este documento.

## **VI. Validación.**

El proyecto proveerá una solución eficiente a la necesidad de transmitir contenido diversificado que tiene canal, pues involucrará la creación de todo un sistema que poseerá un control centralizado y que operará en tiempo real. La intención es que la conmutación pueda ser realizada remotamente a partir de las órdenes dadas por el master, encargado de disponer el contenido de la programación en el momento adecuado. Todo esto es con el fin de generar un sistema dinámico que se acople al existente y que permita agilizar el recurso tiempo, que es de carácter fundamental para la empresa beneficiada.

En cuanto a la demostración del funcionamiento, se realizarán demostraciones de la operación del sistema primeramente a nivel de prototipo, demostrando el funcionamiento de los bloques que van siendo implementados según las fechas a acordar. Tentativamente, podrían hacerse pruebas piloto que no influyan en la operación normal del canal a fines de demostrar el funcionamiento del prototipo terminado de una manera más concisa.

Para el caso de las primeras demostraciones del sistema, se propone hacerlas en redes locales, para fines de que todo el sistema resida en un mismo sitio y pueda ser evaluado como un todo. Posteriormente podrían realizarse las demostraciones en los sitios remotos, según se cuente con la disponibilidad para ello.

## VII. Descripción detallada del proyecto.

### 7.1 Sistema de control de las estaciones remotas.

El sistema de control remoto lo comprenden todos los equipos ubicados en las estaciones repetidoras, así como el software que sus diversas partes ejecutan. A continuación se muestra un diagrama en bloques que muestra las interconexiones entre los distintos componentes de este sistema:

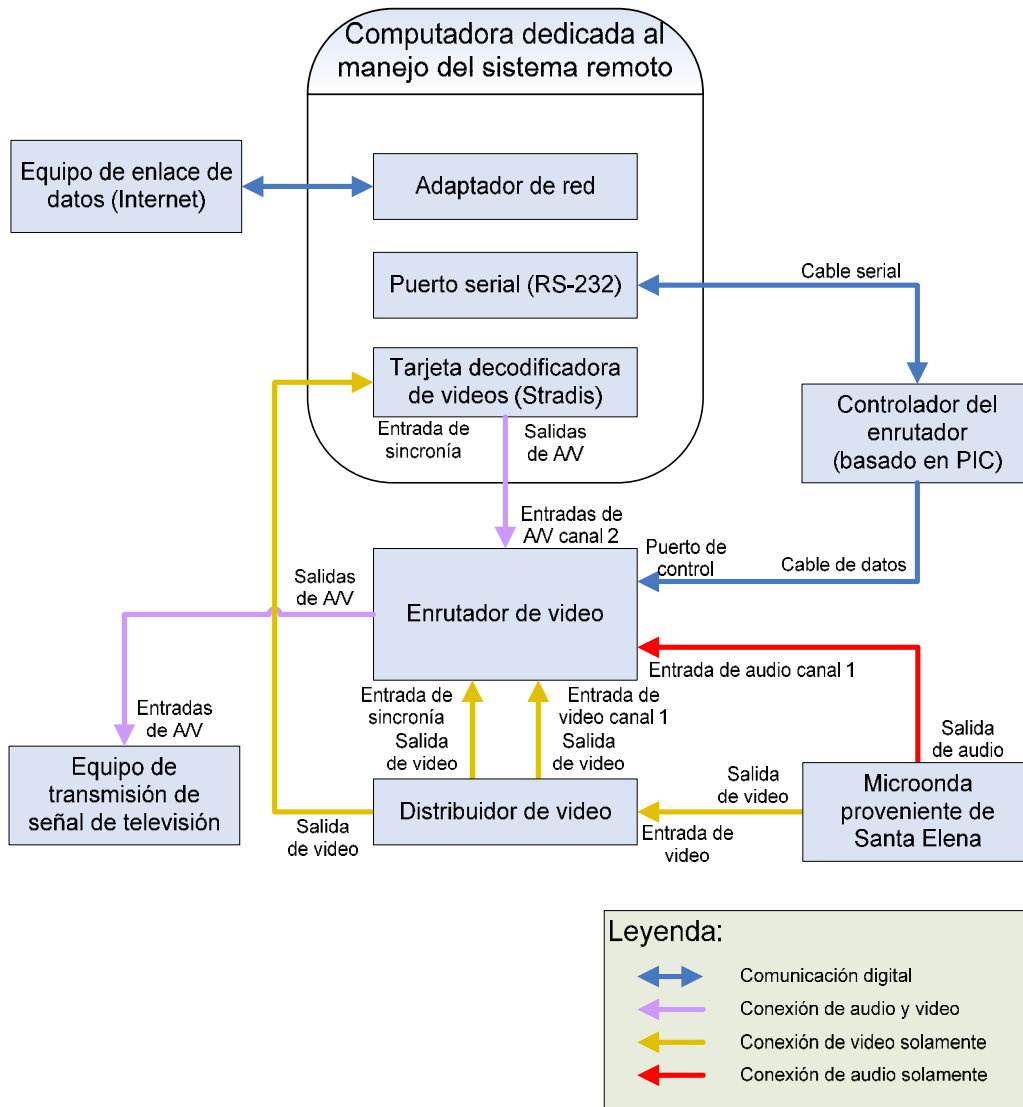


Figura 7.1 – Diagrama de conexión entre los equipos de la estación remota.

En el diagrama anterior fueron omitidas todas las conexiones de alimentación y periféricos tanto de la PC como de los demás equipos por motivos de claridad, mostrándose únicamente las conexiones de datos, audio y video que son relevantes.

Las funciones principales que puede realizar este sistema son:

- Permitir el paso de la señal piloto proveniente de Santa Elena desde la microonda al transmisor (Para transmitir el mismo contenido televisivo que la zona central).
- Interrumpir la señal piloto e insertar contenido grabado, el cual se transmitirá exclusivamente a la zona a la que este dedicada el equipo (zona oriental u occidental). Dicho contenido grabado será reproducido a través de la tarjeta decodificadora de video en la PC.
- Interrumpir la señal piloto y permitir que una tercera (o cuarta) fuente de señal pase al transmisor (pensado para posibles ampliaciones futuras).

Este sistema es capaz de seleccionar, generar y enviar diferentes señales de video al equipo de transmisión, por lo que se hace necesario sincronizar todos aquellos equipos que generen o manipulen las señales<sup>7</sup>. Dicha sincronía se obtiene a partir de la señal de video piloto (proveniente de la cabina del master en Santa Elena), la cual los equipos obtienen indirectamente a partir de la señal televisiva proveniente del equipo de microondas. Esta señal se reparte por medio de un distribuidor de video<sup>8</sup>, el cual posee salidas independientes con impedancias acopladas, de tal forma que se distribuya la señal a todos los equipos sin generar pérdidas.

El método de sincronía utilizado tanto por el decodificador de videos y el enrutador consiste en aplicar la señal de video proveniente del estudio como referencia (y no una

---

<sup>7</sup> Es práctica común sincronizar todos los equipos de televisión en los estudios, para garantizar transiciones suaves entre las imágenes y evitar saltos. Este sistema no está exento de esta necesidad.

<sup>8</sup> También conocido en el ambiente técnico como “Video Splitter”.

señal de sincronía especial dedicada para tal fin<sup>9</sup>), pues estos equipos son capaces de extraer la temporización de dicha señal aunque posea contenido dinámico.

### **7.1.1 Sistema de conmutación.**

En el sistema de conmutación participan diferentes componentes tanto de hardware como software para la selección de las señales de video a enviar al equipo de transmisión. Este apartado detallara cada uno a continuación:

#### **7.1.1.1 Enrutador de audio/video.**

El enrutador de audio/video es un equipo especializado cuya principal función es la de tomar una señal de video de una de sus entradas y dirigirla a su salida. En este sentido pues, puede considerársele como un multiplexor de video. Los modelos considerados para este rubro son dos: El enrutador AEC-1VAA de Adrienne Electronics Corporation y el VS-611 de Kramer Electronics.

Dada la relativa simplicidad operacional de estos enrutadores no se harán explicaciones más profundas de su teoría de operación, dado que es muy simple. Solo cabría agregar que el cambio de canal (señal de origen) lo realizan justo cuando se completa el cuadro de video próximo al comando de cambio de canal (para hacer una transición suave en un proceso totalmente automático), y que para ello necesitan una señal de video externa de la cual toman referencia para determinar el momento justo de la conmutación.

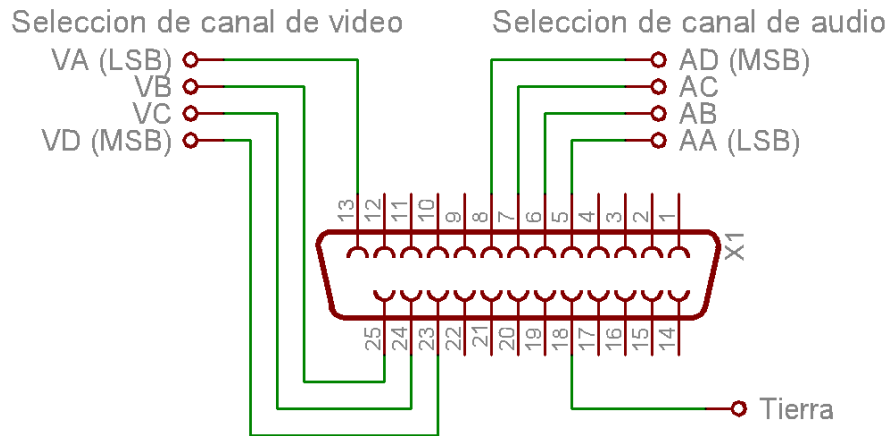
A continuación se hará una descripción de la interfase de control digital que exponen los enrutadores, para explicar posteriormente la forma en que se ha llevado a cabo su control a través de un microcontrolador.

---

<sup>9</sup> Una señal de sincronía utilizada comúnmente es la denominada “black burst” o “pantalla negra”.

## Adrienne Electronics AEC-1VAA.

En la parte posterior de este modelo se encuentra el puerto de control, el cual puede ser usado para seleccionar el canal de video directamente por medio de introducir una señal digital. La distribución de pines del puerto se muestra a continuación:



**Figura 7.2 – Distribución de los pines del puerto de control en el AEC-1VAA.**

Cabe aclarar que el puerto provee otras interfases que no están mostradas por simplicidad, entre ellas una interfase serial RS-232, pero por motivos de flexibilidad y seguridad, se opta por usar el control digital directo, cuyos pines son los que están mostrados. Nótese además que este modelo de enrutador permite seleccionar los canales de audio en forma independiente de los canales de video.

Un aspecto del el AEC-1VAA que vale recalcar, es que el control manual esta implementado internamente de forma tal que el usuario puede elegir el canal a través del teclado frontal, sin embargo para la función de control remoto, es necesario remover un par de circuitos integrados que se localizan en su interior <sup>[1]</sup>, así los circuitos de selección de canal se desconectan del teclado y las líneas digitales de control quedan libres para ser manejadas externamente.

Los pines de control son entradas digitales, compatibles con lógica TTL/CMOS de 5V. A continuación se muestra la tabla de verdad para la selección de canales de audio y video:

Entradas lógicas				Canal de video Seleccionado
VD	VC	VB	VA	
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10

**Tabla 7.1 – Tabla de verdad de selección de canal de video en el enrutador AEC-1VAA.**

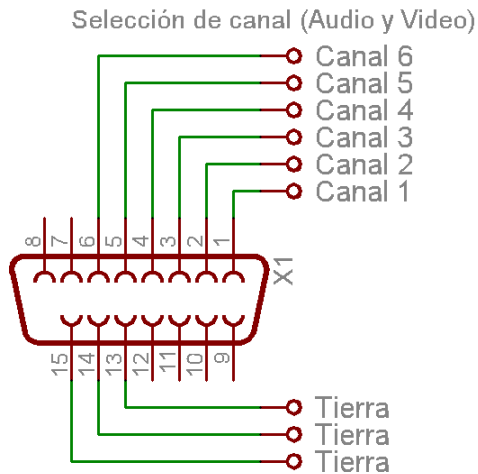
Entradas lógicas				Canal de audio Seleccionado
AD	AC	AB	AA	
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10

**Tabla 7.2 – Tabla de verdad de selección de canal de audio en el enrutador AEC-1VAA.**

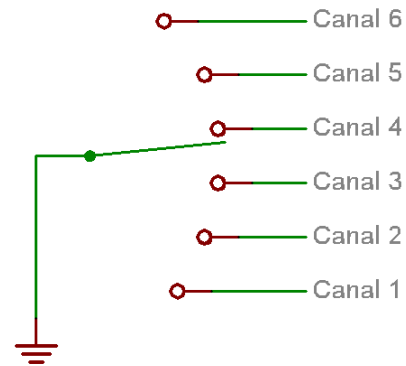
Como puede verse, el proceso de selección de canal se simplifica bastante con este enrutador de video, pues bastará para ello conectar las salidas digitales de los puertos en un microcontrolador hacia estas entradas de control para poder elegir externamente el canal. El enrutador se encargará automáticamente de hacer el cambio real de la fuente de video en el próximo intervalo de sincronía vertical de la señal de video.

### **Kramer Electronics VS-611.**

De manera similar al modelo anterior este enrutador posee un puerto de control en la parte posterior, aunque para este caso en particular se opera de una manera sustancialmente distinta. A continuación se muestra el esquema de conexión del puerto junto con la conexión sugerida por el fabricante <sup>[2]</sup>:



**Figura 7.3 – Distribución de los pines del puerto de control en el VS-611.**



**Figura 7.4 – Esquema de conexión de control sugerido por Kramer Electronics.**

Como puede apreciarse, la forma en que el fabricante sugiere implementar el control remoto es mediante un switch de un polo por varias posiciones (selector de perilla). Mientras que esto es muy práctico para aplicaciones sencillas, supone cierto problema desde el punto de vista de automatización del sistema, puesto que la propuesta del fabricante es un control completamente manual que no toma en cuenta otras alternativas de comando por PC o por microcontrolador.

Para solventar esta situación, se optó por simular electrónicamente el switch selector. La solución empleada se verá en el siguiente apartado.

Cabe aclarar que el tipo de switch a usar debe ser del tipo que no cortocircuita los contactos durante la transición<sup>10</sup>, y por esa misma razón se encontró durante el desarrollo de este proyecto que la conmutación instantánea del selector (por medios electrónicos) produce un funcionamiento errático del enrutador de video (el equipo a menudo no cambia de canal cuando se le indica), posiblemente debido a que dentro de él haya circuitos que compensan el rebote del selector y que fueran diseñados tomando esos factores en cuenta. De ahí que el firmware del controlador del conmutador produce un breve retardo para abrir totalmente el selector antes de cerrar el nuevo contacto.

<sup>10</sup> Conocido en inglés como “Break Before Make”, que significa que al mover el selector, todos los contactos quedan totalmente desconectados antes de unirse para formar una nueva conexión.

### 7.1.1.2 Circuito del controlador del Enrutador.

Este pequeño circuito basado en microcontrolador se encarga principalmente de controlar el enrutador de video. La necesidad de implementarlo nace principalmente de la necesidad de que el sistema sea confiable; De no existir tal necesidad, el sistema podría prescindir completamente de él y la labor de control del enrutador de video pudiera pasar a ser función exclusiva de la PC en el sitio remoto.

Sin embargo, el depender totalmente de la PC al para controlar todo el sistema presenta un peligro potencial, puesto que los sistemas operativos usados no son del todo estables a largo plazo (es decir, por varios días de operación continua). Si bien el diseño de este sistema procura garantizar la máxima estabilidad, sabemos que esto podría no cumplirse del todo, además existen otros factores entre los que podemos incluir: fallas en el suministro eléctrico, fallas de las piezas de hardware por exceso de horas de operación continua (falla de los discos duros, por ejemplo), pérdida del enlace de Internet, etc. Así que en vista a estos posibles percances, se ha optado por agregar un sistema externo confiable pero sobretodo muy estable, que permita al menos que la señal de video piloto proveniente de la microonda llegue al transmisor en caso que la PC deje de responder adecuadamente por cualquiera de las causas antes citadas.

Téngase en cuenta que si ocurre una eventualidad que impida el funcionamiento correcto de la PC (como por ejemplo que se reinicie), la misma no podría indicar automáticamente al enrutador de video que cambie la señal a transmitir. Asimismo, el enrutador de video no posee la “inteligencia” suficiente como para cambiar de canal automáticamente bajo una posible eventualidad, por lo que es necesario un agente externo “inteligente” capaz de detectar una falla potencial en la PC y actuar acordeamente controlando el enrutador.

En caso que la PC no pueda mantener una comunicación con el agente diseñado para tal fin, se puede asumir con seguridad que la misma ha dejado de operar correctamente (o que por una u otra razón el programa administrativo diseñado para tal fin dejó de ejecutarse), lo que también llevaría a pensar consecuentemente que la PC ya no está en capacidad de

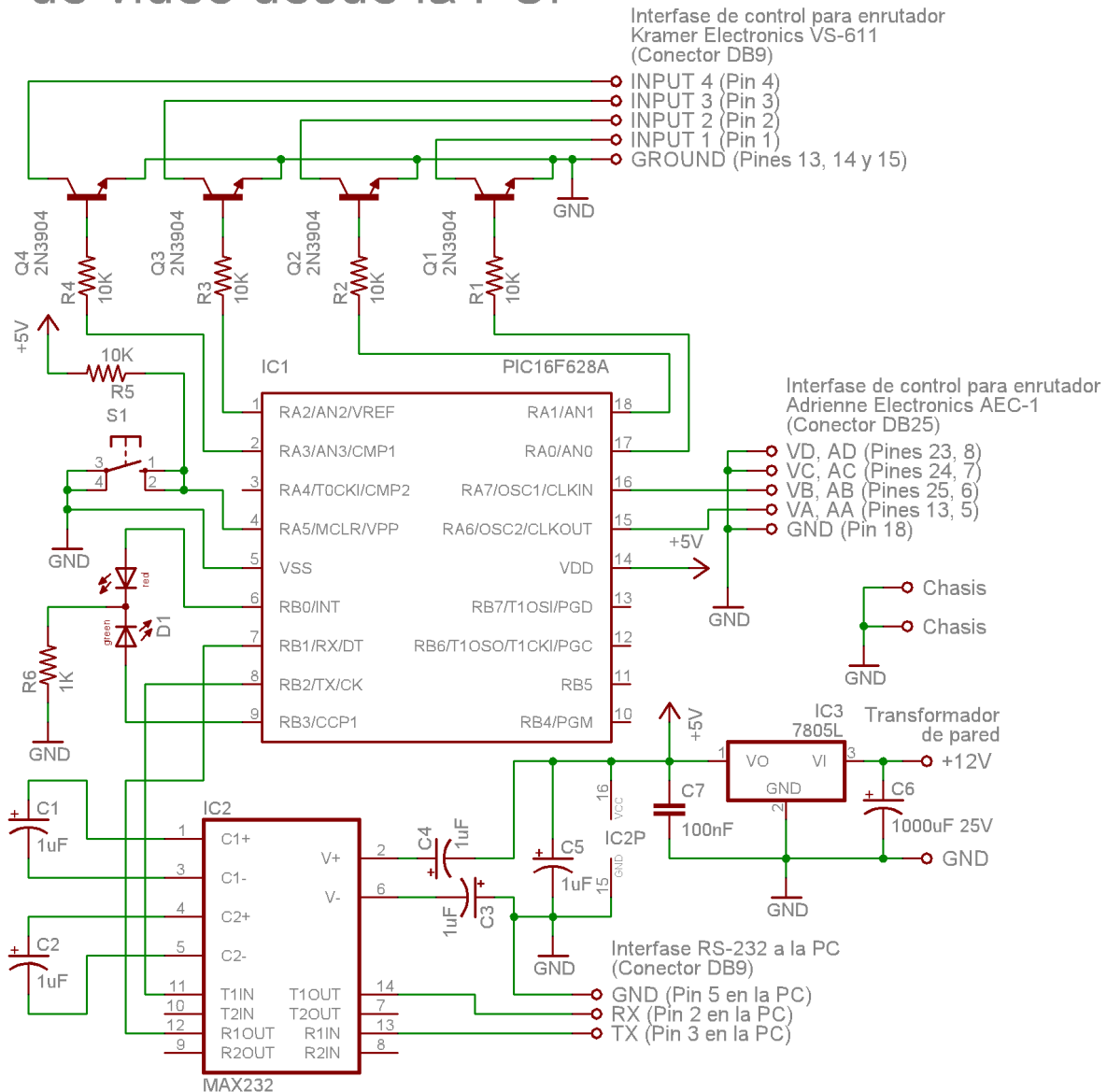
seguir reproduciendo videos (mediante la tarjeta decodificadora que tiene instalada), de tal forma que al no haber garantía de que lo que se está transmitiendo al aire sea un video reproducido adecuadamente, se opta por transmitir la señal piloto para no interrumpir el servicio.

Así pues, el circuito de control del conmutador mantiene una comunicación activa con la PC durante todo momento, de tal forma que si la PC por alguna u otra razón deja de responder por un período razonablemente prolongado, éste podrá indicar al enrutador de video en forma automática que seleccione el canal de entrada por defecto (canal 1), para que la señal de microonda proveniente del estudio pase al transmisor y no se interrumpa la transmisión televisiva.

El microcontrolador utilizado se trata del PIC16F628, o bien su revisión más reciente: el PIC16F628A, ambos desarrollados por la empresa Microchip. La elección de este microcontrolador sobre otros se basa principalmente en su bajo costo y fácil adquisición en el mercado local de componentes electrónicos, aunado al hecho de que posee ya la cantidad justa de periféricos necesarios en la misma unidad para su función particular en el sistema. Asimismo, la elección de un microcontrolador por sobre otras tecnologías (como puede ser la lógica digital discreta) se basa en la reducción significativa de componentes que el uso de un microcontrolador implica, la cual junto al bajo costo difícilmente puede ser superada por otras tecnologías.

En este sistema se hace uso de las características del MCU tales como “Watch Dog Timer” y “Brown Out Reset” <sup>[3]</sup>. La primera garantiza que el MCU siempre responda aunque ocurra una mala operación del mismo a nivel de ejecución de firmware, mientras que la segunda garantiza que el CPU responda siempre a pesar de las disminuciones de tensión en el suministro eléctrico. Estas dos características combinadas permiten que el sistema sea muy robusto y confiable a pesar de las condiciones adversas del entorno que pudieran provocar el mal funcionamiento del firmware.

# Sistema de control para el enrutador de video desde la PC.



**Figura 7.5 – Diagrama electrónico del controlador del enrutador de video.**

En el esquema anterior puede verse que las conexiones hechas al enrutador AEC-1 son conexiones digitales directas usando un puerto del microcontrolador, aprovechando para ello que las entradas de control para el enrutador son TTL/CMOS de 5V. Para manipularlas se utiliza únicamente los dos LSB del puerto de control, por lo que el límite establecido para manejo de canales es de 4. Esto permitirá seleccionar entre la señal piloto, la salida de video de la tarjeta decodificadora de videos y dos canales adicionales que están reservados

para futuras ampliaciones. Cabe aclarar que como parámetro de diseño general, el límite de canales manejados será siempre de 4, tanto para hardware como para software.

La interfase de control para el enrutador VS-611 se implementa a través de los transistores Q1 a Q4 que trabajan en modo de conmutación con una configuración de emisor común, simulando electrónicamente un switch de un polo por 4 posiciones, tal como el puerto de control del enrutador lo requiere. La activación de los transistores será de uno a la vez como máximo en todo momento, apagando temporalmente todos los transistores en el momento de la transición (para simular la característica de apertura antes de cierre del switch).

Lamentablemente la documentación del VS-611 no incluye diagramas electrónicos internos, por lo que se tuvo que recurrir a mediciones externas a fin de estimar cálculos para los transistores que manejan el puerto de control. Así pues, se pudo apreciar que todos los pines del puerto de control presentan un voltaje de +5V cuando se encuentran abiertos, de manera que se concluye que lo que existe en las entradas de control es un circuito con entradas digitales y resistencias de pull-up hacia la fuente (para predeterminedar un estado alto). Esto nos lleva a estimar el valor de la resistencia de polarización de base de los transistores, de tal forma que el transistor se sature manejando únicamente la resistencia de pull-up interna como carga en el colector.

El valor de las resistencias de base (R1, R2, R3 y R4) se eligió arbitrariamente de 10KΩ, por lo que la resistencia mínima de colector que se puede manejar es:

$$\begin{aligned}
 I_B &= \frac{5 - 0.7}{10K} & I_C &= \beta \cdot I_B = 150 \times 430 \mu A & I_{CSAT} &= \frac{5V - V_{CESAT}}{R_C} = \frac{5V - 0}{R_C} \\
 I_B &= 430 \mu A & I_C &= 64.5 mA & R_C &= \frac{5V}{I_{CSAT}} = \frac{5V}{64.5 mA} = 77.52 \Omega
 \end{aligned}$$

La ganancia típica ( $h_{fe}$  ó  $\beta$ ) del 2N3904 es de  $150^{11}$  [5], lo que implica que puede manejar una carga de hasta  $77.52\Omega$  como mínimo manteniéndose todavía saturado. Si se considera que las resistencias de pull-up típicas oscilan entre  $1K\Omega$  y  $47K\Omega$ , se puede estar seguro de que el transistor se saturará fuertemente para cualquier valor de resistencia que el enrutador pudiera contener en su interior.

El cálculo de la resistencia limitadora del LED de estado es muy simple, requiriéndose entre 2 y 5mA para que su grado de luminosidad sea adecuado y sin dañarlo (nótese que a pesar de ser un LED bicolor, sólo un color se enciende a la vez):

$$I_{LED} = \frac{5V - V_{LED}}{1K\Omega} = \frac{5V - 1.5V}{1K\Omega} = \frac{3.5V}{1K\Omega} = 3.5mA$$

Naturalmente esta corriente se encuentra dentro del límite de manejo de corriente de los puertos del PIC16F628A (25mA) [3].

Por otra parte, la determinación del valor de los capacitores que acompañan al MAX232 (C1, C2, C3, C4 y C5) se hace por medio de la hoja técnica [4], ya que éstos vienen predeterminados por el mismo fabricante.

Entre otros parámetros de diseño se incluye el consumo total del circuito para determinar la corriente máxima que manejarán tanto la fuente como el regulador de voltaje (78L05). La carga más importante del circuito es la del microcontrolador, que a 4MHz consume poco menos de 30mA, mientras que la segunda carga más considerable es el LED indicador de estado (aproximadamente 3.5mA). El consumo del MAX232 es básicamente despreciable y no se toma en cuenta (típicamente  $15\mu A$ ) [4], mientras que los transistores drenan su corriente del colector directamente del enrutador (no generan carga mas allá de la corriente de base de  $430\mu A$ ), por lo que la capacidad total del regulador de voltaje, que es de 100mA [6] resulta más que suficiente para esta aplicación.

---

<sup>11</sup> Dato aproximado obtenido a partir de la gráfica de ganancia versus corriente de colector para un valor de corriente de colector (estimada inicialmente) de 10mA.

### 7.1.1.3 Protocolo de comunicación entre la PC y el controlador del enrutador.

Bajo la perspectiva de seguridad del sistema de control del enrutador, se ha diseñado un protocolo de comunicación compacto y simple, que permite la recuperación de datos y la resincronización entre los equipos por medio de comandos simples. La forma en que el controlador determina que la PC ha parado de funcionar, consiste en llevar la cuenta del tiempo que ha transcurrido desde que la PC envió el último comando. Si este tiempo supera cierto límite, el controlador dará por sentado que la PC ha parado de funcionar e inmediatamente dará la orden al enrutador de video para que seleccione el canal 1 automáticamente.

La comunicación realizada se hace a través del estándar RS-232 usado ampliamente en gran cantidad de sistemas, lo que le da una gran compatibilidad. Por motivos de diseño, se establecen los siguientes parámetros para la comunicación:

- Velocidad de comunicación de 9600 baudios.
- Tamaño de palabra de 8 bits.
- 1 bit de parada.
- Sin paridad (reemplazada por el uso de checksum<sup>12</sup>).

La velocidad se elige de 9600 baudios por ser lo suficientemente baja como para asegurar la integridad de los datos, pero al mismo tiempo es lo suficientemente alta como para permitir una comunicación casi instantánea entre los sistemas, gracias a que el tráfico de datos es relativamente reducido.

El tamaño de palabra se elige de 8 bits en forma natural, pues es manejado en forma nativa tanto por el PIC como por la PC, mientras que el bit de parada sencillo se usa así pues es la única forma soportada por el periférico serial dentro del PIC.

---

<sup>12</sup> El término en inglés “checksum” denota el concepto de “valor de comprobación”. Es decir, se trata de un número generado a partir de los datos que el emisor envía en la trama, calculado de tal forma que el receptor puede saber si los datos llegaron correctamente haciendo un proceso de cálculo inverso. Entre más bits son usados para el proceso, más confiable es el resultado del proceso de comprobación.

Finalmente se usa un método de verificación de datos mucho más robusto que el simple bit de paridad, que aparte de no ser soportado por el hardware del PIC (a menos que se use emulación por firmware), no provee la protección necesaria por su elevada probabilidad de confirmar datos erróneos como correctos.

A continuación se detallarán las características de las tramas para dicho protocolo:

### Trama de la PC.

La trama proveniente de la PC contiene únicamente comandos, los cuales el controlador del enrutador deberá analizar y ejecutar. Esta trama consiste en 3 bytes, los cuales se detallan a continuación:

<b>Campo</b>	Cabecera	Datos	Checksum
<b>Tamaño</b>	1 byte	1 byte	1 byte

**Tabla 7.3 – Formato general de la trama de la PC.**

➤ Cabecera:

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
1	0	1	0	1	0	1	0

**Tabla 7.4 – Contenido de la cabecera enviada por la PC.**

Para el caso de la computadora, esta siempre envía 0xAA al inicio de cada trama.

➤ Datos:

En este caso, los campos de bits dependerán del comando. La lista de comandos se detallará más adelante.

➤ Checksum:

Este es un número casi único que acompaña a cada trama, el cual sirve para determinar a través de un método matemático simple la integridad de la trama de datos.

Generación:  $\text{Checksum} = 0x00 - \text{Cabecera} - \text{Datos}$   
(Solo se toman los 8 LSB del resultado)

Comprobación:  $\text{Cabecera} + \text{Datos} + \text{Checksum} = 0x00$   
(Los 8 LSB del resultado deben de dar cero)

### Trama del controlador del enrutador.

El controlador se dedica únicamente a enviar respuestas a los comandos de la PC, siempre que sea pertinente (hay casos en los que no debe haber respuesta del hardware). Para todo caso, la respuesta del controlador deberá ser siempre predecible, por lo que si la PC recibe una respuesta no esperada, podrá tomar las medidas adecuadas para restablecer la comunicación. La trama del controlador, al igual que la de la PC, posee 3 bytes:

Campo	Cabecera	Datos	Checksum
Tamaño	1 byte	1 Byte	1 byte

Tabla 7.5 – Formato general de la trama del controlador del enrutador.

➤ Cabecera:

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	1	0	1

Tabla 7.6 – Contenido de la cabecera enviada por el controlador del enrutador.

Para el caso del controlador, este siempre envía 0x55 al inicio de cada trama.

➤ Datos:

En este caso, los campos de bits dependerán de la respuesta. La lista de respuestas se detallará más adelante.

➤ Checksum:

El algoritmo de generación y comprobación del checksum es idéntico al de la PC, el cual ya fue descrito anteriormente.

## Comandos de la PC.

Para este diseño, el único comando disponible es el de establecer el canal del enrutador de video. El sistema actual soporta un máximo de 4 canales, por lo que los mismos se codifican en 2 bits dentro de su comando correspondiente. Adicionalmente, se incluye un comando auxiliar de resincronización.

D7	D6	D5	D4	D3	D2	D1	D0	Comando
0	0	0	0	0	0	0	0	Resincronizar comunicación
0	0	0	0	0	1	C1	C0	Establecer canal

Tabla 7.7 – Lista de comandos de la PC.

➤ Comando “Resincronizar comunicación”:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0

Tabla 7.8 – Campos de bit del comando de resincronización.

Este es un comando auxiliar cuya función es la de restablecer la sincronía de la comunicación (es decir, la alineación de las tramas en ambos extremos) por medio de evacuar el búfer<sup>13</sup> de recepción del microcontrolador y forzar en el mismo la condición de escucha. Nótese que el comando de resincronización carece de cabecera y checksum, por lo que se envía directamente y no posee comprobación de errores alguna por tratarse de un comando especial.

Tras recibir exitosamente este comando, el controlador no enviará respuesta alguna a la PC.

### Técnica de resincronización.

La PC es responsable en todo momento de asegurarse que la comunicación se mantenga en sincronía. En el momento que la PC perciba un posible error de sincronía dará comienzo a

---

<sup>13</sup> Búfer es un término con el que se designa a una zona de memoria específicamente dedicada a almacenar los datos pendientes de algún proceso, ya sea lectura, escritura, o cálculos hechos por un CPU. Bajo otros contextos, la palabra búfer puede tener otros significados.

la rutina de resincronización, la cual consiste en enviar el comando de Resincronización 3 veces, es decir, en enviar 3 bytes consecutivos con valor de cero. Ante esto, el controlador terminará de recibir cualquier comando incompleto con los últimos valores en cero, a lo que muy probablemente contestará con error, pues no habrá un checksum correcto o el comando será inválido; A continuación la PC vaciará su búfer de recepción e ignorará cualquier respuesta del controlador. De esta forma el controlador se quedará nuevamente en condición de escucha, y en caso de que tras completar el comando se recibieran ceros extra, estos serán simplemente recibidos como comandos de resincronización y no se generará respuesta. A partir de este momento ambos extremos quedan sincronizados y puede reanudarse la operación normal del sistema.

➤ Comando “Establecer canal”:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	C1	C0

**Tabla 7.9 – Campos de bit del comando de selección de canal.**

Este comando establece el canal actual del enrutador de video, seleccionando así la fuente de video que se enviará al transmisor. Los canales disponibles se detallan a continuación:

C1	C0	Canal	Función del canal
0	0	Canal 1	Señal piloto (canal por defecto)
0	1	Canal 2	Decoder (salida de video de la PC)
1	0	Canal 3	Reservado para futura ampliación
1	1	Canal 4	Reservado para futura ampliación

**Tabla 7.10 – Distribución de canales para el comando de selección de canal.**

Tras la recepción exitosa de este comando, el controlador del enrutador cambiará el canal actual a la nueva selección de forma inmediata. Luego el controlador responderá con "Canal seleccionado OK" (ver mas adelante los detalles de dicha respuesta).

## Respuestas del controlador del enrutador.

Bajo el diseño actual del sistema existen dos posibles respuestas por parte del controlador. Estas son la de selección exitosa de canal y la de error de comunicación. Recuérdese que el comando de resincronización de la PC no posee respuesta, por lo que no aparece tabulada una respuesta para el mismo.

D7	D6	D5	D4	D3	D2	D1	D0	Respuesta
0	0	0	0	0	0	0	0	Error de comunicación
0	0	0	0	0	1	C1	C0	Canal seleccionado OK

**Tabla 7.11 – Lista de respuestas del controlador del enrutador.**

➤ Respuesta “Error de comunicación”:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0

**Tabla 7.12 – Campos de bit de la respuesta de error de comunicación.**

Siempre que el controlador determine que ha ocurrido un error en la comunicación enviará esta respuesta. Ha de aclararse que esta respuesta, a diferencia del comando de resincronización, se envía dentro de una trama completa y no constituye una excepción a pesar de que solamente se envían ceros.

Si el controlador determina que la cabecera, checksum o comando son inválidos en la trama, enviará esta respuesta tras completar la recepción de la misma (es decir, esperara a que se termine de recibir la trama antes de responder con error). A continuación la PC podrá tomar una acción adecuada como el reenvío de la trama o bien la resincronización de la comunicación.

En caso de que el controlador detecte un error de trama o bien un sobre flujo de datos en su buffer de entrada, también emitirá esta respuesta tras terminar de recibir el comando de la PC.

➤ Respuesta “Canal seleccionado OK”:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	C1	C0

Tabla 7.13 – Campos de bit de la respuesta de selección de canal.

Esta respuesta se envía inmediatamente tras la recepción exitosa del comando "Establecer canal", y sirve para confirmar el canal seleccionado actualmente.

La distribución de los canales es idéntica a la del comando asociado (Tabla 7.10).

#### 7.1.1.4 Firmware del controlador del Enrutador.

El firmware<sup>14</sup> se almacena dentro de la memoria FLASH contenida en el microcontrolador PIC. Dicho firmware es un programa que ejecuta y coordina todas las actividades del hardware del controlador, participando del tráfico de datos con la PC y controlando el enrutador de video a cargo de él.

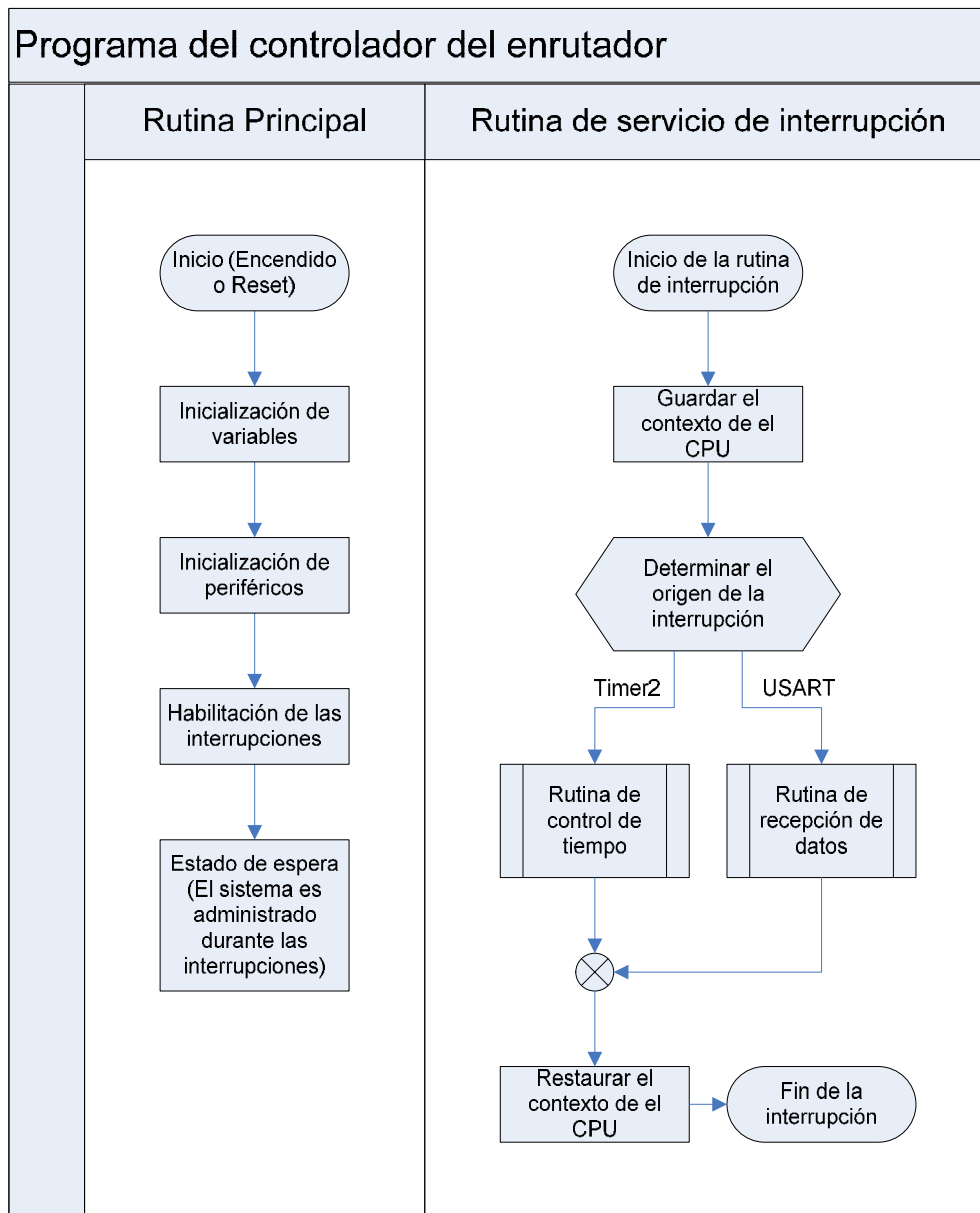
La técnica de programación usada es la de detectar eventos por medio de interrupciones. Así pues, el programa inicializa el sistema durante el arranque, para luego quedarse en un bucle de espera sin realizar ninguna actividad hasta que se genere una interrupción, bien sea del módulo temporizador (Timer2) o del módulo de comunicaciones seriales (USART<sup>15</sup>).

El diagrama de flujo a continuación muestra la rutina principal (arranque) y la rutina de servicio de interrupción general del MCU.

---

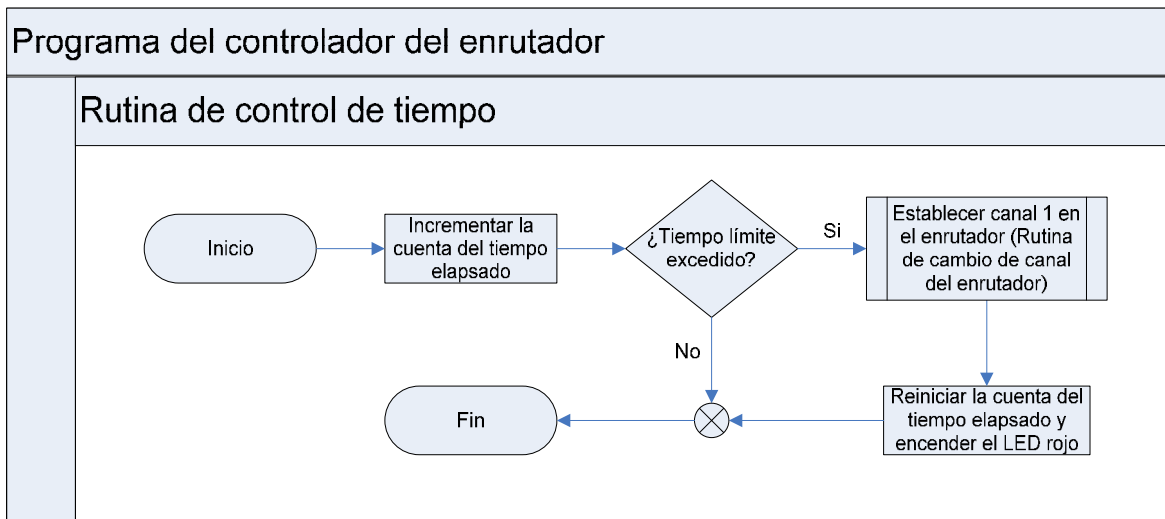
<sup>14</sup> Término en inglés con el que se denota a un programa o conjunto de datos grabado permanentemente ya sea en una memoria, un microcontrolador, etc., utilizando tecnologías de memoria no volátil.

<sup>15</sup> USART es el nombre dado al módulo de hardware que en los microcontroladores PIC que realiza la función de enviar y recibir datos a través de un puerto serie. Sus siglas en inglés significan “Universal Synchronous Asynchronous Receiver Transmitter”<sup>[3]</sup>.



**Figura 7.6 – Diagrama de flujo principal del programa del controlador del enrutador.**

La rutina de control de tiempo sirve para llevar la cuenta del tiempo sin actividad que pudiera mostrar la PC. La idea básica es que si se da un intervalo de tiempo razonablemente prolongado sin recibir datos de la PC, entonces se establece automáticamente el canal 1 en el enrutador de video (para permitir paso a la señal piloto hacia el transmisor). Siempre que haya tráfico exitoso de datos por parte de la PC, la cuenta de tiempo se limpia, de forma que mientras haya comunicación continua la cuenta de tiempo se mantiene siempre en un valor muy bajo o bien cero.



**Figura 7.7 – Diagrama de flujo de la rutina de control de tiempo del enrutador.**

En la figura siguiente se muestra el diagrama de flujo para la rutina de recepción de datos, la cual es invocada como respuesta a la interrupción del módulo USART. Como característica común de su diseño, los microcontroladores PIC proveen interrupciones tanto para los eventos de recepción así como los de transmisión de datos, sin embargo la funcionalidad de interrupción por envío es omitida debido a que la comunicación proveniente del controlador del conmutador se genera únicamente como respuesta a los comandos de la PC (el controlador del enrutador nunca inicia el tráfico de datos). Así pues, el envío de datos por parte de este sistema tiene su lugar como parte de la rutina ISR de recepción de datos.

Nótese que tras recibir la cabecera o los datos, el programa simplemente registra que se recibe correctamente dicha información y a continuación procede a salir de la interrupción, poniéndose a la espera del nuevo segmento de la trama de datos. La ejecución del comando correspondiente y la respuesta asociada se generarán únicamente tras recibir el último byte de la trama, es decir, el byte de checksum. Esto va en concordancia al protocolo de comunicación serial previamente expuesto.

Nótese también que se implementan varios chequeos de integridad de datos, con el fin de garantizar la confiabilidad del sistema y evitar disparos erráticos o comportamientos no deseados.

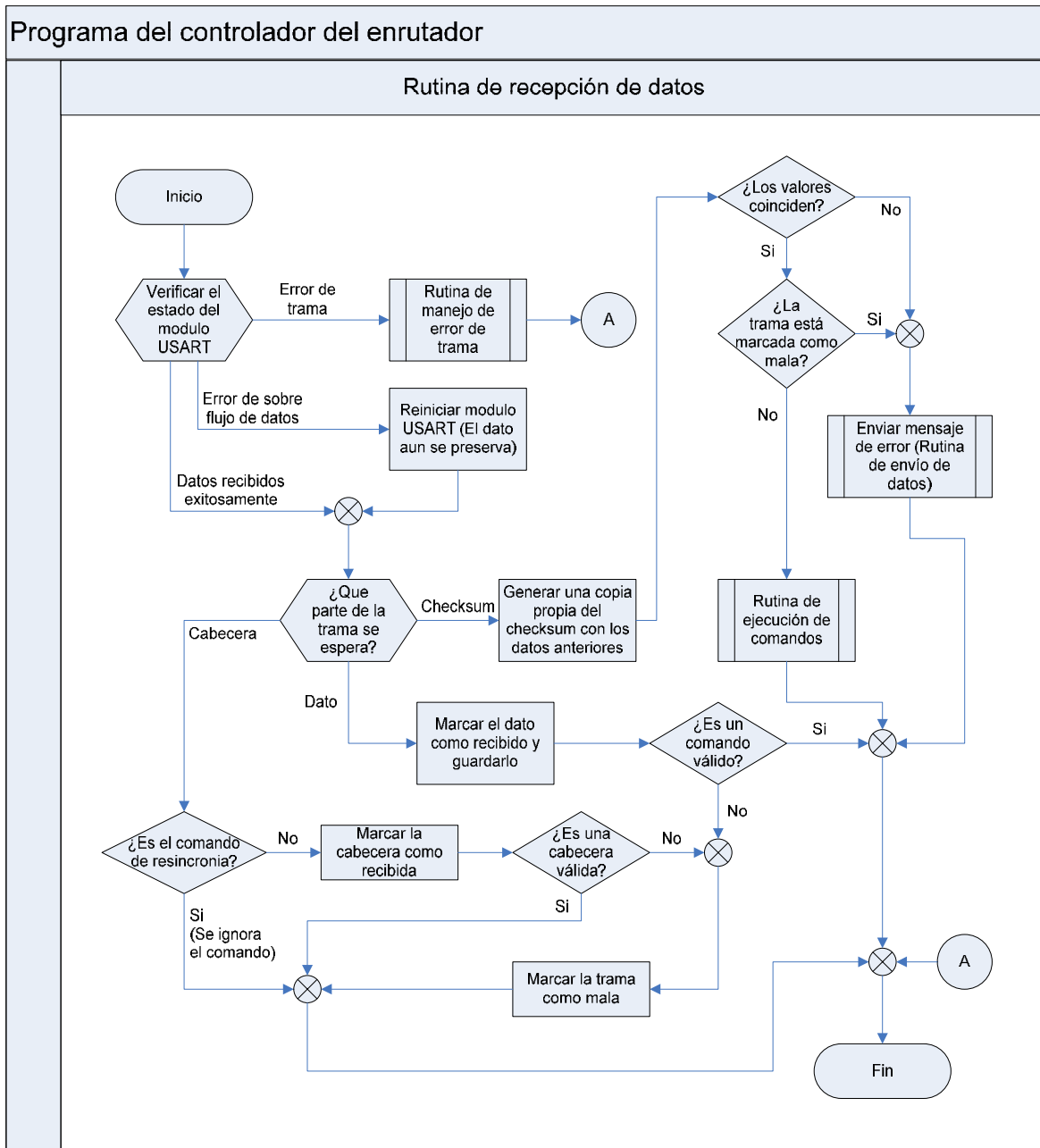


Figura 7.8 – Diagrama de flujo de la rutina de recepción de datos del controlador.

Las rutinas mostradas en la figura siguiente forman la parte operativa principal del control del enrutador de video. La rutina de ejecución de comandos es invocada tras la recepción

exitosa de un comando por parte de la rutina de recepción de datos del módulo USART, y su trabajo consiste en identificar los comandos provenientes de la PC para luego dar lugar a su ejecución. Por circunstancias del proceso de desarrollo del firmware se estimó inicialmente la existencia de más de un comando, sin embargo como medida de optimización la funcionalidad completa del sistema se implementó con el uso de un solo comando. De ahí que existe una mecánica de selección múltiple para identificar los comandos, pero sólo existe un único comando disponible. Dicha mecánica de identificación de comandos se preservará, en caso que futuras ampliaciones del sistema lo requieran.

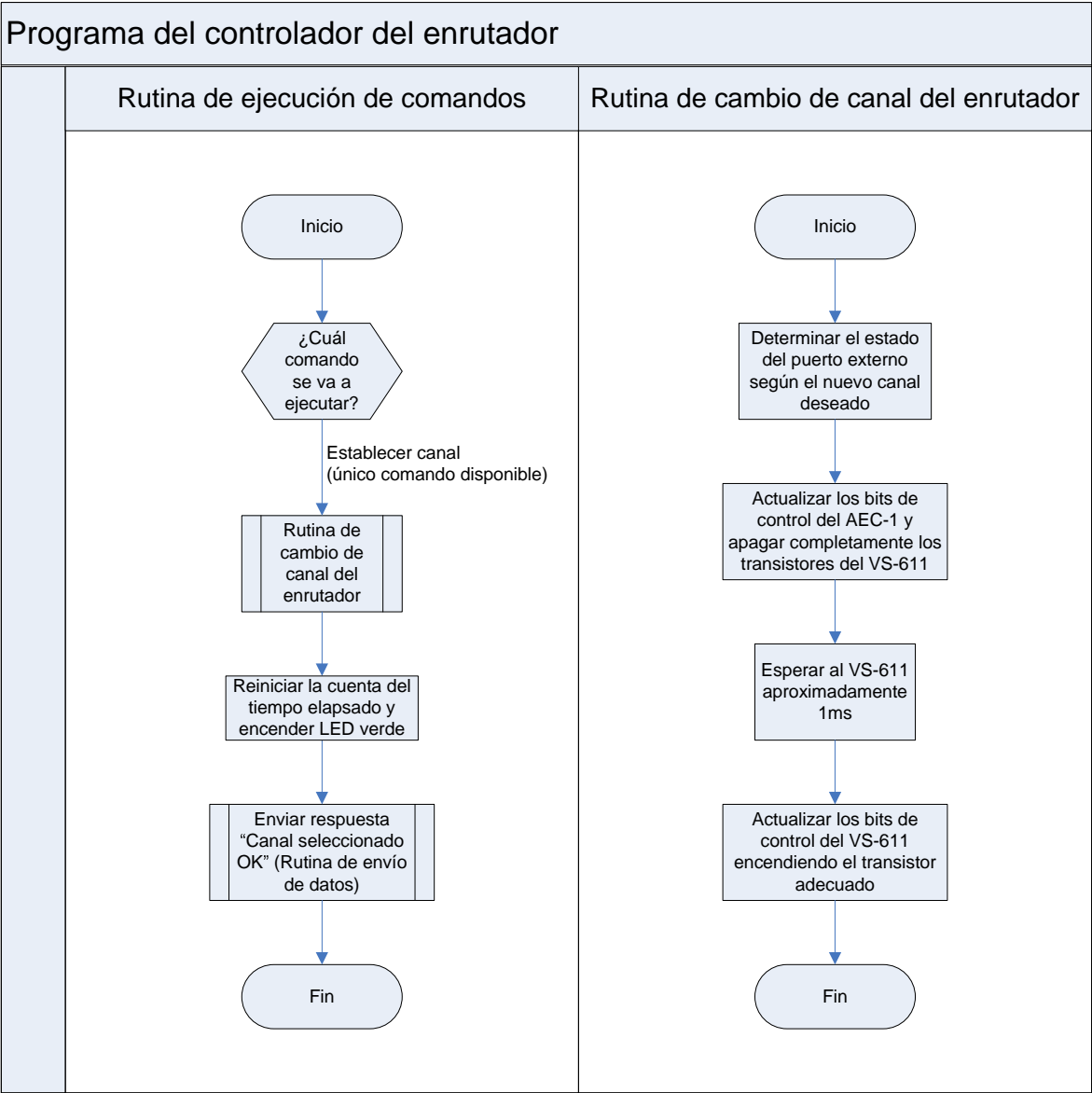


Figura 7.9 – Diagramas de flujo de las rutinas de manejo de hardware del controlador.

La rutina de cambio de canal genera la interacción electrónica necesaria para provocar el cambio de canal del enrutador de video. La implementación de las interfases para el VS-611 y el AEC-1VAA es simultánea, lo cual quiere decir que se puede conectar a este sistema cualquiera de los dos modelos de enrutador sin distinción alguna. Esto se hace por motivos de la disponibilidad limitada de las unidades. En la práctica es posible incluso la conexión simultánea de ambos modelos de enrutador sin que surjan mayores inconvenientes.

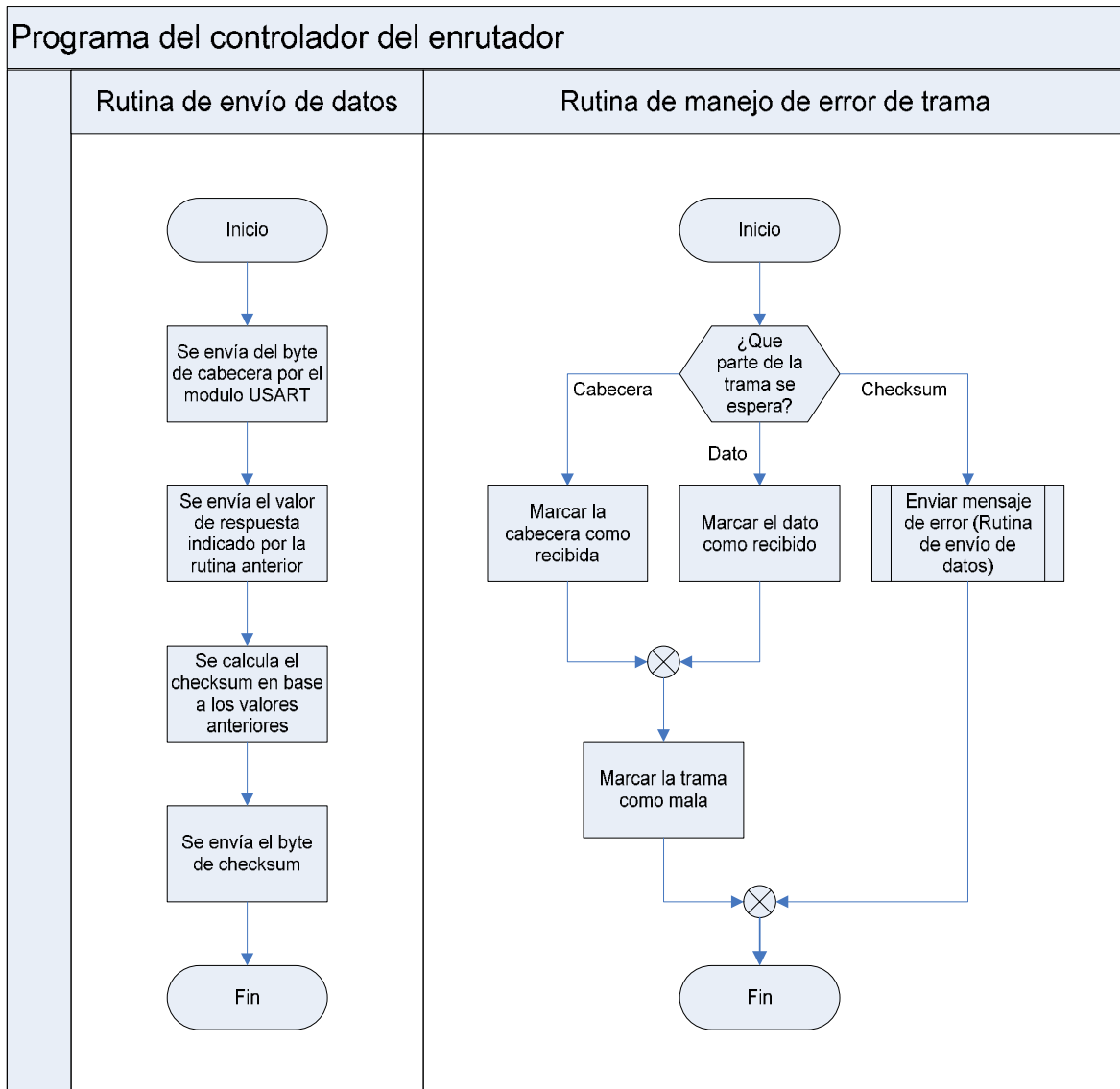
Tal como se mencionó en el apartado que describe al VS-611, se genera un retardo a la espera de que el enrutador registre un cambio en las líneas de control, de forma que el canal sea cambiado electrónicamente simulando el retardo “real” que un switch selector presenta.

Al final de la rutina de ejecución de comandos siempre se reinicia la cuenta del tiempo elapsado, indicando que ha ocurrido una comunicación exitosa con la PC recientemente y evitando que se termine el tiempo límite de espera de la rutina de control de tiempo. Asimismo, el LED verde es encendido indicando al usuario que existe un enlace exitoso entre la PC y el control del enrutador.

Al final del proceso de ejecución del comando se envía una respuesta a la PC a través del puerto serial, indicándole a la misma que el comando de selección de canal se ha ejecutado exitosamente.

Las últimas rutinas a destacar son la de envío de datos por parte del enrutador a la PC y la de manejo de error de trama. La primera es una sucesión directa de eventos en la que se envía primero la cabecera, luego el byte de datos y finalmente se calcula el checksum para enviarlo. Dicha rutina verifica el estado del módulo serial a través de la técnica de “polling”

o “encuesta”<sup>16</sup>, y no termina hasta que el último byte es puesto en el búfer de salida del módulo USART.



**Figura 7.10 – Diagramas de flujo de la rutinas de manejo de datos del controlador.**

<sup>16</sup> Técnica de programación básica en la que se lee continuamente el estado de un periférico para verificar que ha completado una acción previamente iniciada. Es la alternativa básica al uso de interrupciones y suele ser usada únicamente en el caso que el CPU no tenga otra actividad relevante que hacer mientras espera.

Finalmente la rutina de manejo de error de trama<sup>17</sup> es una rutina que se llama en caso que el módulo USART indique ese tipo de error a través de sus banderas de estado. La idea básica detrás de ella es que si se genera una desalineación de los datos (lo cual el hardware detecta si no se presenta el bit de parada adecuadamente), entonces se pierde completamente la trama, por lo que la rutina o bien marca la trama como mala y se prosigue esperar a que la misma se termine de transmitir posteriormente, o bien procede a notificar directamente a la PC de que ocurrió un error si la desalineación se dio al final de la trama. Téngase en cuenta que la transmisión serial es completamente bidireccional<sup>18</sup> y de naturaleza asíncrona, eso quiere decir que las tramas pueden enviarse en cada sentido independientemente del tráfico de datos en el sentido opuesto. Por lo tanto si un error de alineación de trama ocurre en un sentido, no tiene porque afectar necesariamente a la comunicación en el sentido opuesto. Esto permite reportar la falla a la PC a pesar que los datos se han desalineado en el otro sentido de la comunicación.

---

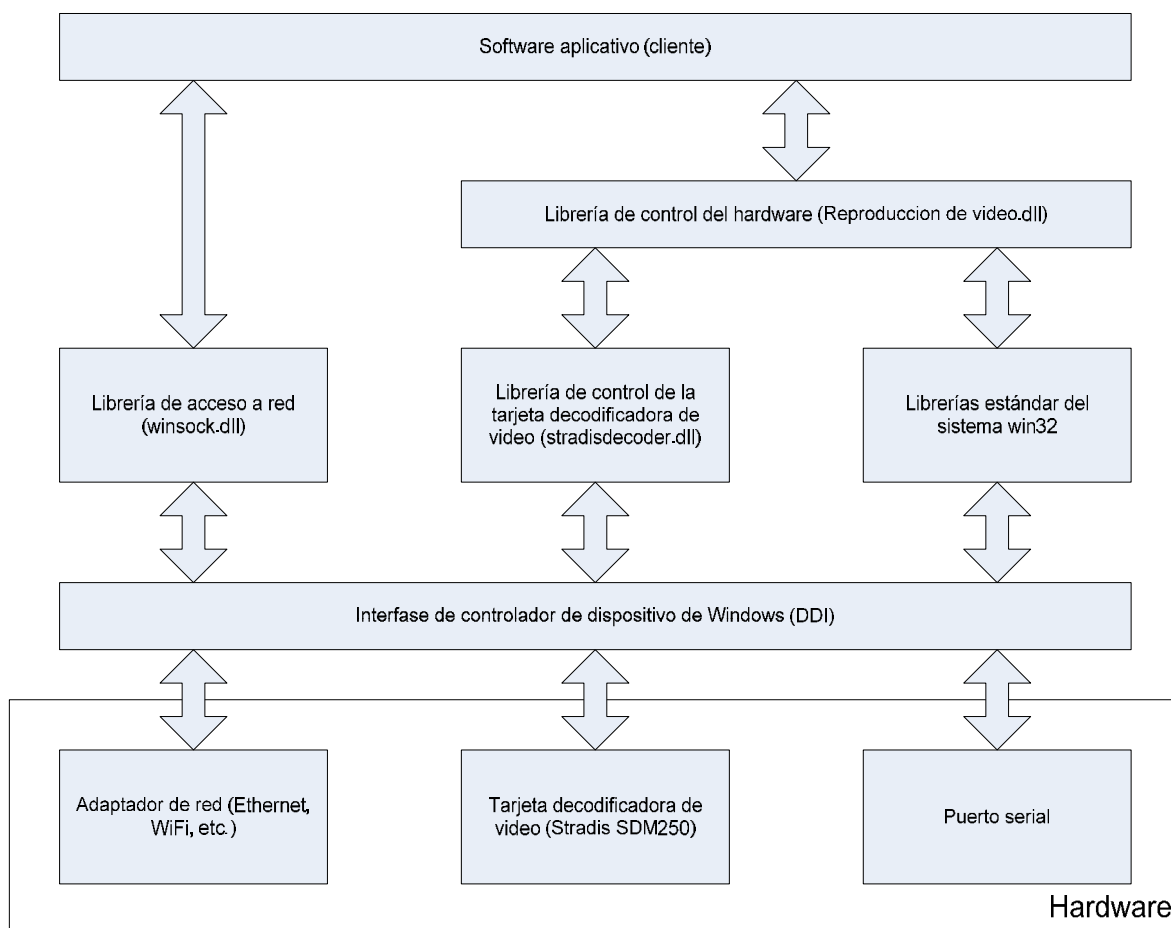
<sup>17</sup> Error conocido por su nombre en inglés como “Framing Error”, es generalmente producido ya sea por una velocidad de comunicación diferente a ambos extremos, un desplazamiento de fase de los relojes internos o bien por simple ruido o errores de comunicación. Este consiste en que la trama de datos seriales se desalinea, y por lo tanto ambos extremos quedan desorientados en cuanto a donde comienzan o terminan los datos. Esto implica necesariamente una pérdida de los datos transmitidos en el instante en que se produce.

<sup>18</sup> Conocido por su término en inglés como “Full Duplex”.

### **7.1.2 Software de control de las estaciones remotas.**

El software de control remoto comprende toda la gama de programas que residen dentro de las PC en los sitios remotos. Sus tareas se enfocan principalmente al manejo del hardware, así como a la reproducción de los videos y la comunicación en red.

La figura siguiente muestra la jerarquía general del software instalado en las PC de los sitios remotos. El software aplicativo, que se encuentra en el nivel superior, provee la interfase visible al usuario y permite al mismo realizar las configuraciones necesarias para que el sistema funcione de manera autónoma (controlado a través de una conexión de red), asimismo realiza el control de todos los componentes del sistema en manera conjunta.



**Figura 7.11 – Jerarquía del software y hardware de control remoto.**

Para poder manejar el hardware externo (decodificador y enrutador de video vía controlador), se hacen llamadas a una DLL<sup>19</sup> diseñada específicamente como parte del proyecto para dicho propósito. Dicha DLL es la piedra angular sobre la cual se basa todo el control del hardware, permitiendo a cualquier software aplicativo (cliente) acceder a las funciones de la misma en una forma universal.

La elección del diseño basado en DLL radica precisamente en la flexibilidad que la arquitectura misma provee, permitiendo a cualquier otro programa hacer uso de la misma sin importar que tipo de herramientas de desarrollo de software sean usadas para crearla, así como también puede hacerse uso de la misma independientemente del lenguaje de programación que se emplee. Así pues, tenemos que prácticamente todos los lenguajes de programación modernos y sus entornos de desarrollo proveen siempre de manera genérica una mecánica para acceder DLL externas, tanto las del sistema operativo como las desarrolladas específicamente para aplicaciones propias.

La librería de control de la tarjeta decodificadora de video (stradisdecoder.dll) permite realizar todas las operaciones asociadas a la misma, esta viene incluida dentro del software de controlador que la empresa Stradis distribuye con el producto. La programación utilizando esta DLL se ha realizado por medio del SDK que la empresa provee de forma separada.

La librería de acceso a red (winsock.dll), la interfase de controlador de dispositivo (DDI) y las librerías estándar forman parte integral del sistema operativo Windows, y proveen las funciones de bajo nivel necesarias para el control del hardware respectivo.

Como último detalle se hace la observación de que la comunicación por red para el manejo remoto del sistema es tarea exclusiva del software aplicativo (esta funcionalidad no es

---

<sup>19</sup> Término que en inglés significa “Dynamic Link Library” o bien “Librería de enlace dinámico”. Una DLL contiene instrucciones ejecutables como si se tratase de un programa, mas no se trata de un programa completo como tal, sino que sirve para ampliar la funcionalidad de otro software (con rutinas previamente programadas) de una forma globalmente reutilizable por todo el sistema operativo y los programas instalados.

proporcionada por la DLL de control de hardware), por lo que el cliente puede usar a su conveniencia la librería mas conveniente para dicho fin. En el sistema de control en cambio, la funcionalidad de red si esta implementada dentro de una DLL que será descrita más adelante. La mecánica completa de comunicación y control vía red de datos, así como la utilización de winsock.dll en el sistema se detallara más adelante.

#### 7.1.2.1 Librería (DLL) de control de hardware.

Como se menciona anteriormente, esta librería está diseñada específicamente para este proyecto con el fin de dar una solución general a la necesidad de acceso y control al hardware de la estación remota. Las tareas de la DLL son muy variadas, entre las que se incluyen:

- Inicialización del hardware bajo control (tarjeta decodificadora y controlador del enrutador).
- Gestionar el protocolo de comunicación serial con el controlador del enrutador, garantizando la comunicación continua del mismo y al mismo tiempo verificando que se encuentre operando adecuadamente.
- Proveer una interfase de selección de canal de enrutador al programa cliente, y efectuar los comandos seriales necesarios para dicho fin.
- Llevar control del estado del decodificador de videos (reproducción, pausa, video actual, etc.).
- Proveer una interfase de control del decodificador de video al programa cliente, y al mismo tiempo gestionar una cola de videos precargados para agilizar las transiciones entre videos.
- Finalizar las operaciones pendientes con el hardware y cerrar o desalojar todos los recursos usados cuando el programa cliente esta por terminar.

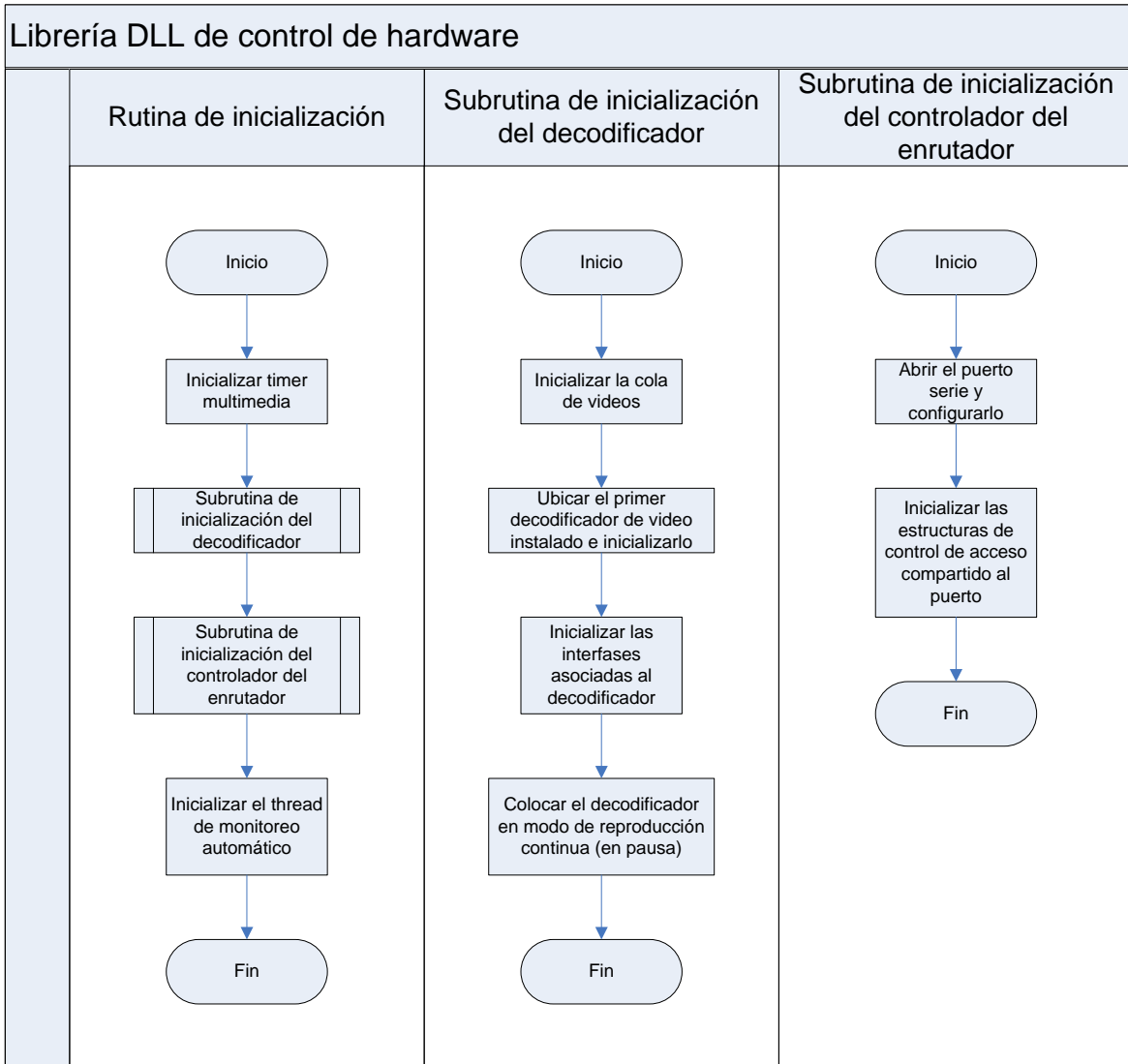
La DLL como tal no es un programa completo, sino mas bien que se compone de una serie de rutinas o funciones que deben ser invocadas por el programa cliente (es responsabilidad de dicho cliente invocar las funciones de la DLL en el orden correcto). Por lo tanto, sus diagramas de flujo corresponden más bien a rutinas aisladas que devuelven el control al

programa cliente en cuanto terminan. Las rutinas que exporta la DLL se detallan a continuación:

<b>Rutina</b>	<b>Nombre dentro de la DLL (en código)</b>
Rutina de inicialización.	InicializarSistemaVideo ( )
Rutina de finalización.	LiberarSistemaVideo ( )
Rutina de adición de videos en cola para su posterior reproducción (precarga).	AgregarVideoEnCola (NombreArchivo)
Rutina para reproducir del siguiente video de la cola.	ReproducirSiguienteVideo (Canal)
Rutina para detener la reproducción de video.	DetenerVideo (Canal)
Rutina para eliminar todos los videos esperando en la cola.	EliminarVideosEnCola ( )
Rutina de cambio de volumen de la salida de audio del decodificador.	CambiarVolumen (Volumen, Balance)
Rutina de cambio directo de canal del enrutador de video.	PasarCanalExterno (Canal)
Rutina de lectura del temporizador multimedia de alta precisión.	LeerTimer ( )

**Tabla 7.14 – Funciones exportadas por la librería (DLL) de control de hardware.**

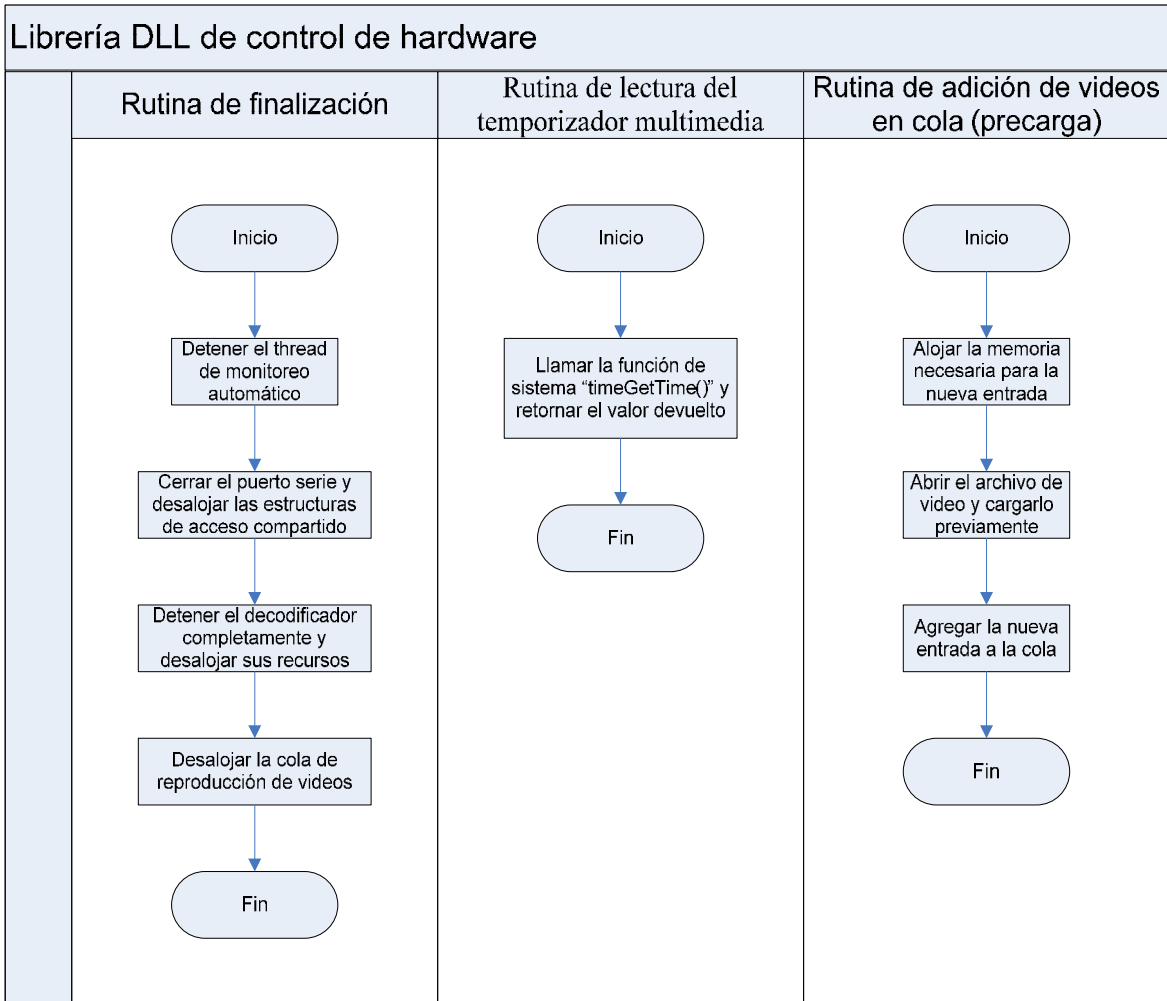
En la tabla anterior se muestra un nombre descriptivo para cada rutina en la columna izquierda, mientras que a la derecha se muestra el nombre de las funciones correspondientes, tal como aparecerían en los lenguajes de programación usados para elaborar el programa cliente. Bajo el nombre en código, se muestran entre paréntesis los argumentos de la función, si los tiene. Para propósitos de este documento se usarán los nombres descriptivos de la izquierda al referirse a cada rutina por ser más explícitos. A continuación se hará una breve descripción de cada rutina.



**Figura 7.12 – Diagrama de flujo de las rutinas de inicialización en la DLL.**

La figura anterior muestra el proceso completo de inicialización de la DLL de control de hardware. Es de aclarar que es responsabilidad del programa cliente llamar esta rutina antes de iniciar cualquier otra operación, caso contrario las demás rutinas de la DLL retornarán un código de error indicando que la DLL no ha sido inicializada.

El siguiente diagrama muestra una cantidad de rutinas variadas, incluidas la finalización de la DLL, la lectura del temporizador multimedia y la de precarga de videos en cola:



**Figura 7.13 – Diagrama de flujo de las rutinas de finalización, lectura de tiempo y adición de videos en cola en la DLL.**

La rutina de finalización, de manera análoga a la de inicialización, debe ser llamada al final del proceso por el programa cliente. Si el cliente falla en hacerlo, los recursos utilizados por la DLL no serán desalojados, lo que puede llevar a una fuga de recursos y errores potenciales por parte del sistema operativo al quedar los recursos pendientes de uso sin ningún programa que los utilice.

La rutina de lectura del temporizador multimedia simplemente provee un acceso a la función de sistema “timeGetTime()” de win32<sup>20</sup>, permitiendo al programa cliente el acceso a un reloj de alta precisión (1ms) en caso que las herramientas de desarrollo de cliente no provean mecanismos para la medición de lapsos de tiempos precisos (este es precisamente el caso de Visual Basic).

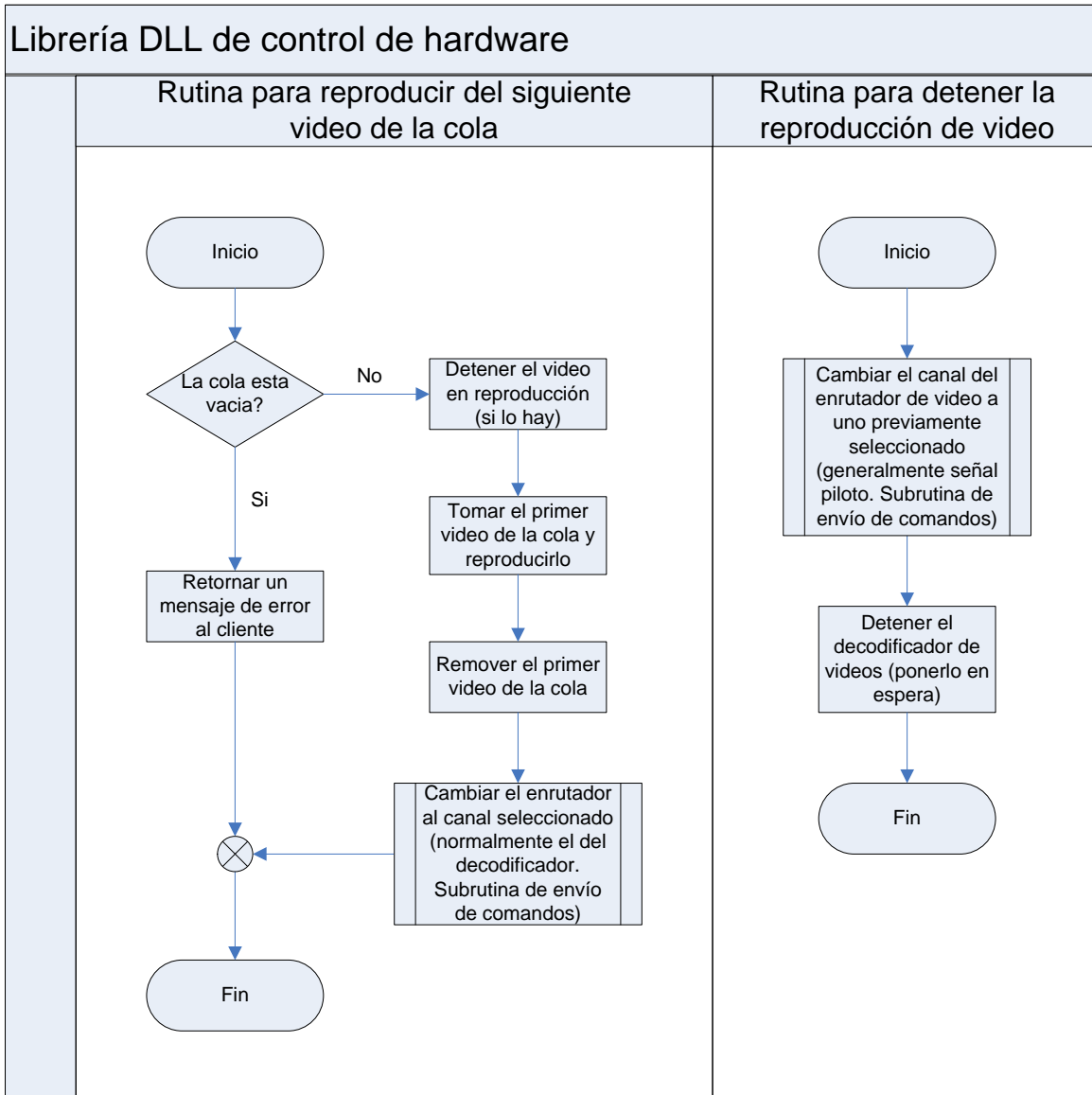
La rutina de adición de videos en cola sirve para preparar al decodificador de video para que reproduzca el siguiente video en forma inmediata. La precarga es necesaria porque el tiempo de carga inicial de un video es considerable (alrededor de un segundo), por lo que el proceso debe hacerse con anticipación para que el video pueda entrar en reproducción en el momento exacto deseado. Nótese que esta rutina no inicia la reproducción en sí del video, sino que más bien lo precarga y causa que el mismo quede “en cola”, esperando a ser reproducido en el instante en que se llama a la rutina para dicho cometido.

La figura a continuación muestra precisamente la rutina que permite la reproducción inmediata de los videos previamente cargados, nótese que la misma toma el primer video de la cola y lo reproduce, asumiendo que el decodificador de videos tiene los datos listos en su búfer. En caso de no haber ningún video en cola, la rutina retornará un código de error indicando esa condición.

Dentro de la misma figura se presenta la rutina que permite detener la reproducción de cualquier video en cualquier momento. Nótese que esta rutina se provee únicamente para casos especiales en donde se desee detener el video por circunstancias excepcionales. Normalmente los videos se detendrán solos al terminar de ser reproducidos, lo cual es el funcionamiento esperado del sistema.

---

<sup>20</sup> Win32 es el nombre interno dado por Microsoft al núcleo de los sistemas operativos Windows de 32 bits, incluyendo al conjunto de DLL de que forman parte del mismo sistema operativo y todos los servicios que proveen. Entre los sistemas que implementan win32 se tienen: Windows 95, Windows 98, Windows ME, Windows XP, Windows 2000 y Windows Vista.

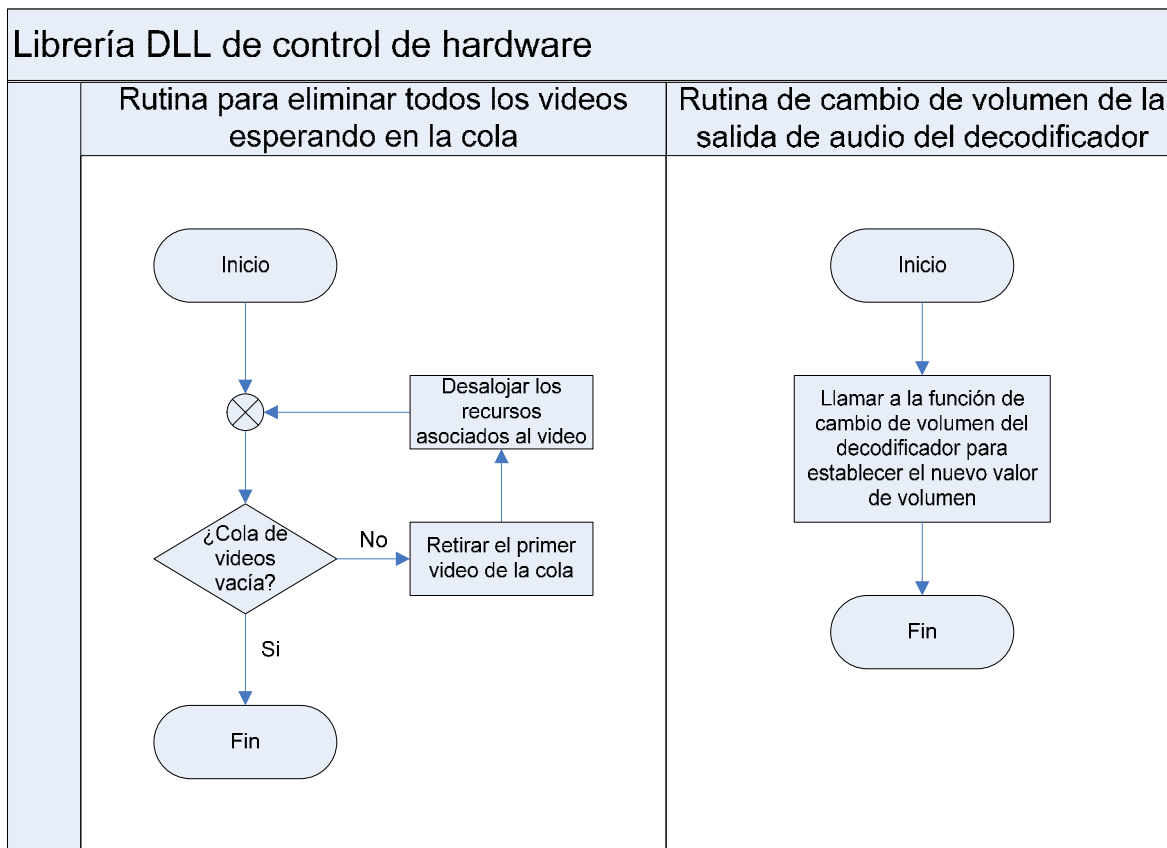


**Figura 7.14 – Diagrama de flujo de las rutinas de inicio y paro de la reproducción de videos en la DLL.**

Cabe señalar que las rutinas anteriores permiten cambiar canal del enrutador de video como parte del proceso, esto es para asegurar que al reproducir un video se seleccione el canal del enrutador correspondiente al decodificador de videos, mientras que al detener un video se seleccione otro canal distinto del asignado al decodificador que posea una fuente de señal válida.

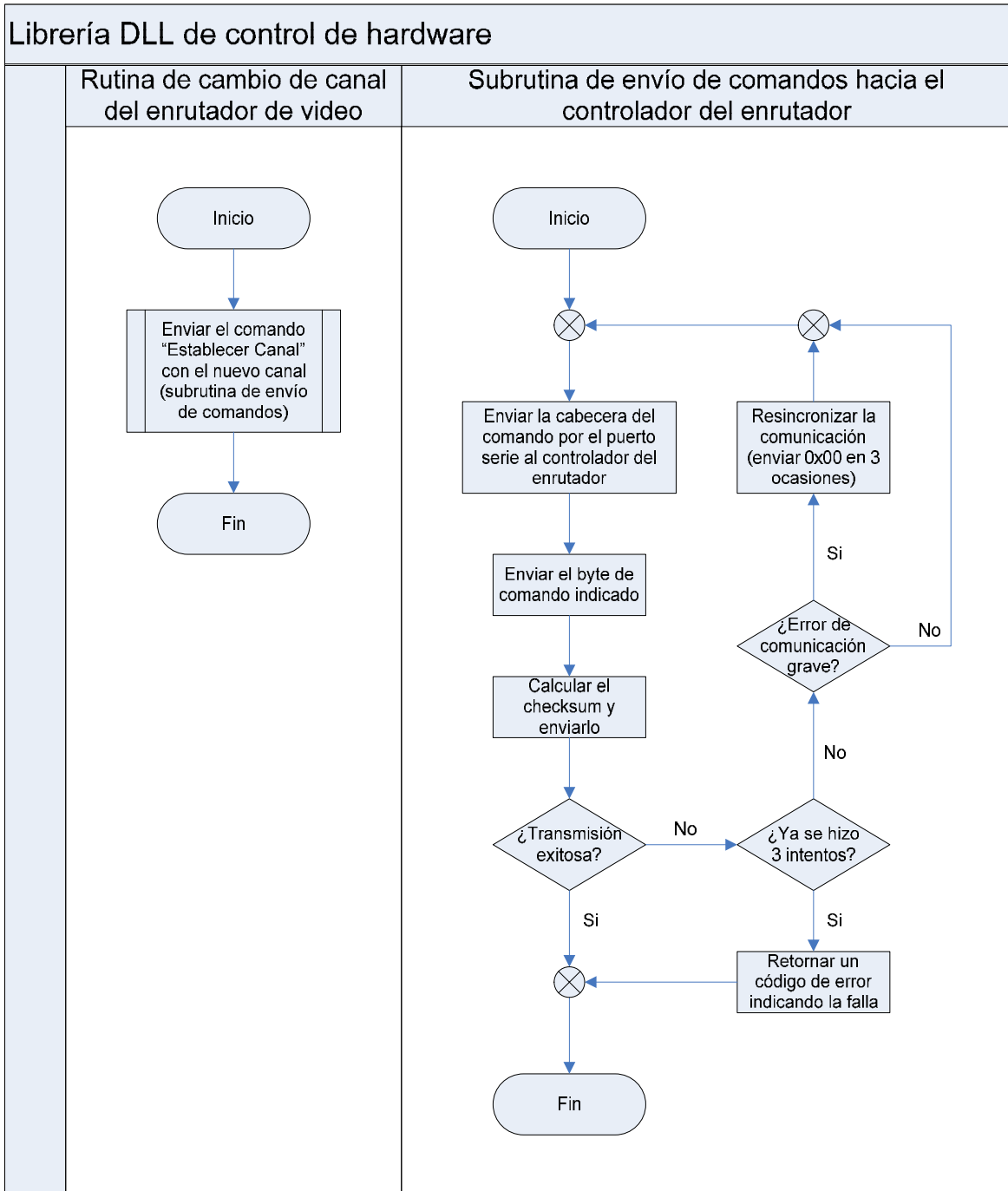
La figura a continuación presenta la rutina de eliminación de videos en cola (purgado o vaciado de la cola de videos). Esta rutina, al igual que la rutina que detiene los videos, está

pensada únicamente para circunstancias especiales en las que no se desea que los videos en cola salgan al aire. Así pues, la operación normal del sistema será que la cola de videos se irá vaciando a medida que los videos son reproducidos.



**Figura 7.15 – Diagramas de flujo de las rutinas de eliminación de todos los videos en cola y cambio de volumen del decodificador en la DLL.**

La rutina de cambio de volumen permite simplemente al programa cliente elegir el nivel de volumen de audio que se desea a la hora de reproducir los videos. Un aspecto importante de la función de control de volumen del decodificador, es que la misma permite únicamente atenuar la salida de audio; esto es, que se elige el nivel normal (seleccionado por defecto) o bien se atenúa el nivel de salida por debajo del normal. Esto es una característica implícita de la tarjeta decodificadora, aunque vale aclarar que el nivel de salida de audio está normalizado y debería ser el adecuado para todos los equipos que acepten señales de audio desde la tarjeta decodificadora.



**Figura 7.16 – Diagrama de flujo de las rutinas de cambio de canal del enrutador y comunicación con el controlador del enrutador en la DLL.**

Finalmente en la figura anterior se muestran las rutinas de la DLL correspondientes a la comunicación serie con el controlador del enrutador. La rutina de cambio de canal se encarga simplemente de iniciar una nueva comunicación con el control del enrutador, de tal forma que se seleccione el nuevo canal según lo solicite el programa cliente. Para ello se

vale de la subrutina de envío de comandos, la cual genera la trama de comunicación serial según se ha detallado anteriormente en la descripción del protocolo serial. Un detalle importante de la rutina es que permite la recuperación de errores de comunicación con el hardware, sin importar el tipo de errores que se generen. Bajo los casos en que los errores de comunicación sean leves (como un checksum malo), simplemente se procede a re-enviar la trama al controlador del enrutador. Para los casos en que ocurran errores graves (como pérdida de sincronía) entonces se procede a restablecer la sincronía de la comunicación con el controlador. El número máximo de intentos se limita a 3, pues generalmente las tramas se envían exitosamente al primer intento y en caso de fallar son siempre exitosas al segundo intento; Asimismo, se limita el número de intentos para retornar el control al programa cliente en un tiempo relativamente corto (para evitar que el programa cliente se quede congelado).

#### 7.1.2.2 Programa de control de la estación remota (servidor remoto de videos).

Este es el programa encargado de realizar el manejo de todo el sistema dentro de una estación remota en tiempo real, recibiendo los bloques de video en forma dinámica y reproduciéndolos en los momentos que se le indica.

Este programa en si carece de utilidades administrativas (la organización de los videos reproducidos no se realiza aquí), por lo que viene a actuar en realidad como un reproductor de video remoto simple que recibe comandos a distancia.

El rol de este software dentro de la topología de red de datos es la de un servidor, por lo que el sistema en cuestión se mantiene inicialmente a la escucha de un puerto de red de entrada, a la espera de la conexión de un solo cliente. En el momento en que el cliente se conecta, ambos entablan una comunicación dinámica en la cual intercambian comandos de reproducción de videos, lecturas de tiempo (para la sincronía), etc.

Dado que todas las estaciones remotas contienen software que actúa como servidor, y que un solo cliente se conecta a todos ellos para controlarlos simultáneamente, tenemos un

sistema cuya topología es muy similar a una red tipo estrella con el cliente como nodo central, convirtiéndose así en una especie de sistema cliente – multi servidor.

A continuación se detallará la operación de este software.



**Figura 7.17 – Ventana principal del software de la estación remota.**

Este sistema es prácticamente autónomo, por lo que no posee una interfase diseñada para que el usuario elija que contenido reproducir, sino más bien, para que el usuario pueda configurarlo y llevar un control de la actividad que él mismo realiza.

En la parte superior izquierda de la ventana se muestra el estado de la conexión, aquí se puede verificar el estado de comunicación en el que se encuentra el programa, el cronómetro en la parte superior indica la cuenta regresiva para el siguiente evento (próximo

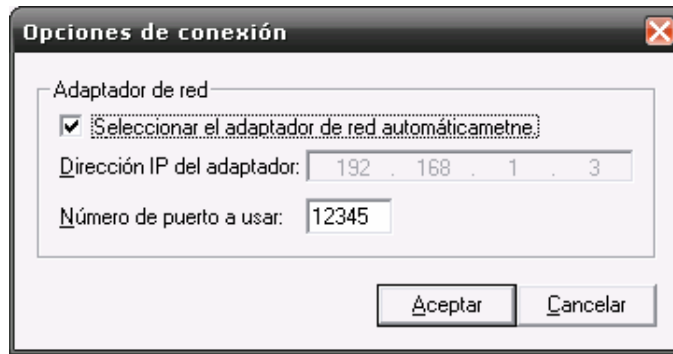
video, fin de video, etc.), mientras que en la parte central existe un botón que permite deshabilitar el sistema en caso de emergencias. En caso de presionar este botón aparecerá un cuadro de confirmación, para evitar activaciones accidentales. Cuando el sistema es deshabilitado, el enrutador de video se fija permanente a la señal piloto (de ahí que se llama “forzar señal piloto”) y por tanto los videos reproducidos no son transmitidos al aire. Cabe aclarar que los videos no paran ni dejan de ser reproducidos por la tarjeta decodificadora, y que en el momento en que se vuelve a presionar el botón (para desactivar el modo forzado) el enrutador vuelve inmediatamente al canal de la tarjeta decodificadora en caso que se reproduzca un video en ese instante.

La parte inferior de la ventana muestra el registro de eventos. Toda actividad que realiza el sistema es registrada aquí y guardada simultáneamente en un archivo de registro (log), aquí el usuario puede enterarse de todas las actividades, transacciones y eventos que ocurren en el sistema remoto. La opción “Limpiar registro de eventos” permite limpiar esta ventana para mejor visualización, mas no borra el archivo de registro del día. La opción “Desplazar registro de eventos automáticamente” deshabilita el desplazamiento automático de la ventana en caso que el usuario desee visualizar detenidamente los eventos que han ocurrido recientemente.

El menú de opciones contiene las siguientes selecciones:

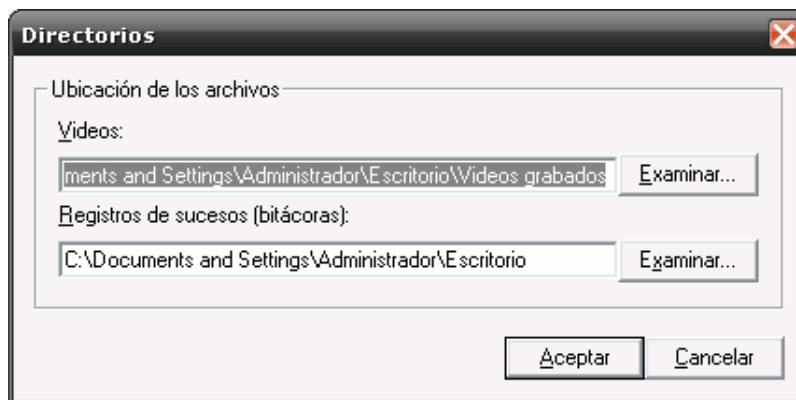
- Conexión.
- Directorios.
- Temporización del sistema.
- Audio.

Cada opción permite configurar un aspecto diferente del sistema, tal como se detallarán a continuación:



**Figura 7.18 – Cuadro de diálogo de opciones de conexión.**

Las opciones de conexión permiten seleccionar la forma en que el servidor aceptará las conexiones entrantes. Básicamente la configuración se enfoca al adaptador (o interfase) de red usado. La casilla de selección automática permite al programa seleccionar el adaptador de red por defecto del sistema. En caso de no activarse, el usuario debe proveer la dirección IP del adaptador por medio del cual se escuchará la conexión entrante del cliente. Asimismo, se puede designar el número de puerto TCP por medio del cual se realizará la conexión. Cabe aclarar que ambos extremos deberán usar el mismo número de puerto, y que el mismo no debe ser ocupado por ningún otro programa en ninguno de los extremos.



**Figura 7.19 – Cuadro de diálogo de selección de directorios.**

El cuadro de diálogo de selección de directorios permite fijar las rutas predeterminadas donde el programa localizará tanto los videos, así como los registros de sucesos. En caso de los videos, todo video que sea solicitado y que no posea una ruta absoluta (completamente resoluble), será buscado dentro del directorio designado aquí. La ruta de los registros de sucesos indica al programa donde almacenar los mismos. El sistema generará

automáticamente un archivo de registro de eventos cada día (cada archivo esta fechado), con el propósito de organizar mejor los registros y facilitar la búsqueda de los mismos, así como de no permitir que crezcan excesivamente.

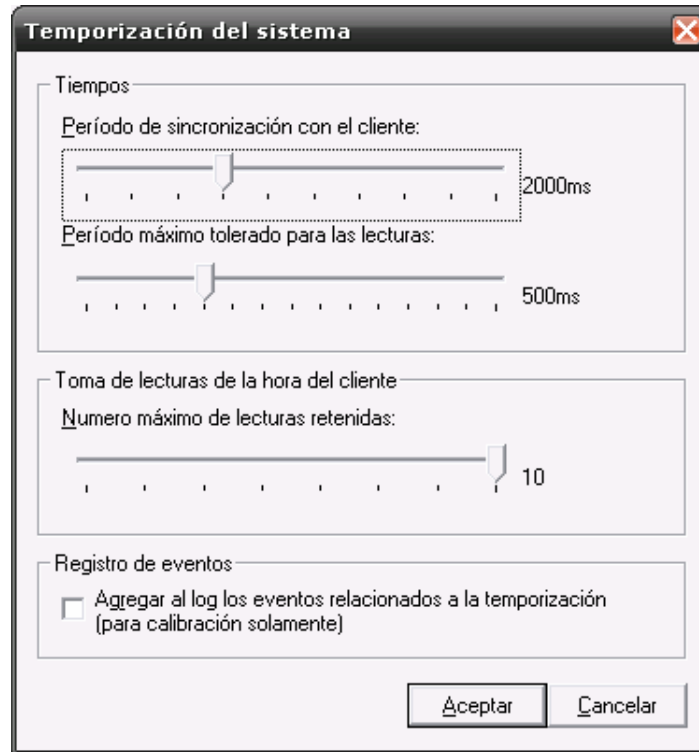


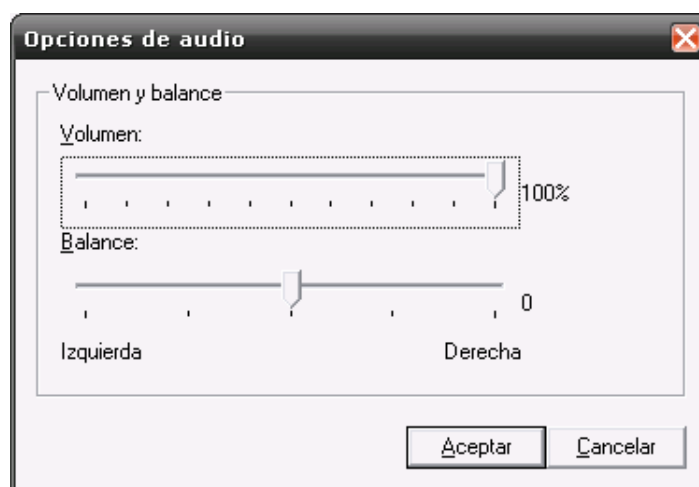
Figura 7.20 – Cuadro de diálogo de temporización del sistema.

Las opciones de temporización modifican la forma en que este servidor sincroniza su hora con el cliente (nótese que sincronizan sus cronómetros internos, no la hora del sistema). Estas opciones permiten adecuar los parámetros de los algoritmos de sincronización para una mejor adaptabilidad a la red. Las opciones por defecto asumen una red con una latencia promedio de 200ms y con una variabilidad en el tiempo de viaje relativamente grande (el caso de Internet). En caso de poseerse una red más estable, puede ajustarse estas opciones para una mayor precisión del sistema.

El período de sincronización con el cliente fija los lapsos entre lecturas, es decir, la cantidad de tiempo que espera el servidor antes de iniciar una lectura de tiempo. Valores mejores permiten ajustar el reloj más rápidamente, pero sólo es beneficioso si los relojes locales no son muy estables y la red tiene una latencia muy pequeña.

El período máximo tolerado para las lecturas permite fijar el máximo tiempo de viaje tolerado del comando de sincronización (ver la sección del protocolo de comunicación para más detalles), si las lecturas exceden este tiempo, son descartadas. Valores mayores permiten trabajar en redes con más latencia, a costa de perder cierto grado de precisión cuando ocurren lecturas muy tardadas.

El número de lecturas retenidas permite ajustar la cantidad de lecturas recientes que participan del algoritmo de extrapolación de la hora del cliente. Entre más lecturas se tienen, más precisa es la hora estimada, pero menos susceptible es el sistema de actualizar su reloj rápidamente en caso de variaciones rápidas.



**Figura 7.21 – Cuadro de diálogo de opciones de audio.**

El cuadro de diálogo con las opciones de audio es muy simple, y permite fijar el nivel de volumen y balance de la salida de audio de la tarjeta decodificadora de videos. Esto es para un mejor ajuste del nivel de audio que se envía al transmisor.

Finalmente, el menú de ayuda contiene únicamente la opción “acerca de” donde se muestra la información del programa y su autor.

### 7.1.2.3 Protocolo de comunicación entre cliente y servidor vía red de datos.

En este apartado se procederá a explicar a mayor detalle el protocolo de red usado para la comunicación entre el sistema central (ubicado en el master) y las estaciones remotas. Como se detalló anteriormente, toda la comunicación entre los sitios se realiza a través de Internet, bien sea directamente (conectándose a una IP pública previamente adquirida), o bien a través de una VPN (conectándose a una IP privada).

El protocolo de red implementado hace utilización directa de la pila de protocolos TCP/IP<sup>21</sup>, a través de la API (librería) de sockets de Berkeley<sup>22</sup>, o más específicamente, de la implementación para Windows de la misma librería, conocida como “Windows Sockets”<sup>23</sup>. La utilización de este conjunto de protocolos y librerías permite un transporte rápido y eficiente de los datos entre los extremos de la red.

Este sistema hace uso de sockets orientados a conexión (sockets TCP/IP), los cuales proveen una especie de “tubería” por la cual los datos entran por un extremo y salen por el otro extremo en el mismo orden, con una garantía casi total que no han sido corruptos a lo largo del viaje. Una desventaja de la utilización de los sockets es que los mismos no proveen una fragmentación 100% automática de la información en paquetes (la fragmentación de los paquetes se logra por medio de un esfuerzo combinado entre el socket y el programa cliente que hace uso de él), por lo que el aplicativo que hace uso del socket

---

<sup>21</sup> TCP/IP es una de las suites (combinación de protocolos) más usadas en Internet, junto con UDP/IP. TCP/IP se caracteriza por proveer transporte confiable y seguro por medio de comprobación de errores y control de flujo.

<sup>22</sup> La librería de sockets de Berkeley (la cual obtiene su nombre de la universidad de California en que fue creada) es un proyecto de código libre que nació en el sistema Unix. Su principal utilidad consiste en brindar una interfase simplificada para comunicar 2 o más computadoras en red, ya sea a través de los protocolos TCP/IP (sockets orientados a conexión) o UDP/IP (sockets sin conexión).

<sup>23</sup> Windows Sockets es el nombre dado por Microsoft a la implementación de la librería de sockets de Berkeley. La portación de esta librería a Windows obedeció al hecho de que los sockets de Berkeley se convirtieron en un estándar de facto para la creación de programas que emplean conectividad de red y para mantener compatibilidad con los programas existentes que emplean esta API. Dentro del sistema operativo Windows, esta API reside principalmente en la librería de sistema “Winsock.dll”

debe ser programado de una manera específica para lograr una fragmentación adecuada de la información.

Luego, cuando una estructura o grupo de datos es enviada por un extremo, el otro extremo podrá retirarla con todos los bytes en el mismo orden en que fueron enviados por medio de su propio socket. Asimismo, se dice que ambos extremos mantienen una “conexión activa”, la cual perdura hasta que uno de los dos extremos cierra la conexión o bien la misma se termina a causa de un error de la red. Esto último permite llevar un control del inicio, cierre o interrupción de la conexión entre los extremos.

Toda conexión TCP/IP comienza con uno de los 2 extremos abriendo un puerto de entrada, a la espera de conexiones entrantes hasta que el otro extremo da inicio a la conexión poniéndose en contacto con él, con lo que comienza el intercambio de datos. Generalmente, el extremo a la escucha de conexiones entrantes actúa como servidor en el sistema (el caso de los reproductores de videos en los sitios remotos), mientras que el extremo que da inicio a la conexión se comporta como cliente (en este caso el sistema de control).

La tabla a continuación presenta la estructura general de todos los comandos diseñados para este sistema, los cuales son transferidos sobre la red TCP/IP:

<b>Segmento</b>	<b>Tamaño</b>
Header	4 bytes
Bloque de Datos	variable
CRC-32	4 bytes

**Tabla 7.15 – Formato general del comando TCP/IP.**

Esta estructura aplica tanto para el servidor como el cliente, y dado que estos datos son enviados sin mayor modificación por medio del socket, estos viajan propiamente como la carga o “payload” de los paquetes TCP.

Si bien tanto el protocolo TCP como la capa de enlace (Capa 2 – LLC del modelo OSI) proveen control de errores por medio de sus propios algoritmos de checksum, la verdad es

que no protegen de errores de lógica a nivel superior (como por ejemplo, la pérdida de sincronía de la trama de bytes entre el cliente y el servidor), ya que se limitan a transportar lo que se les indica. Por lo tanto, se agrega una comprobación adicional por medio de checksum del tipo CRC-32<sup>24</sup> para prevenir esta clase de errores. En caso que uno de los dos extremos encuentre un paquete malo, procederá a cerrar la conexión en forma unilateral.

A continuación se detallarán los comandos del sistema:

➤ Comando de identificación del cliente para con el servidor.

Segmento	Campo	Contenido	Tamaño
Header	Id. del comando	Numero que identifica a este comando (valor de 0).	4 bytes
Comando de identificación	Numero de bytes del nombre	Indica el tamaño de la cadena con el nombre del host cliente.	4 bytes
	Nombre del host	Cadena con el nombre de host del cliente en formato ASCIIZ <sup>25</sup> .	Variable
Checksum	CRC-32	Número usado para comprobar el contenido, generado a partir de los dos segmentos anteriores.	4 bytes

**Tabla 7.16 – Contenido del comando de identificación.**

Este comando es enviado por el cliente en el momento en que la conexión se realiza con éxito. Sirve para propósitos informativos del usuario solamente, para identificar al otro equipo que se encuentra conectado el servidor.

---

<sup>24</sup> El algoritmo CRC (cyclic redundancy check) es un mecanismo de comprobación de errores que genera un número casi único a partir de un grupo de bytes. El sufijo 32 indica que el número generado es de 32 bits (nivel de protección moderado). Existen varios tipos de CRC-32, pero el tipo usado en este sistema es uno de los más populares, mismo que usado en otras tecnologías tal como Ethernet, ZIP y PNG.

<sup>25</sup> El formato ASCIIZ es prácticamente idéntico al formato ASCII, con la excepción que las cadenas van seguidas de un 0 para indicar su final (normalmente conocido como “cero terminador” de la cadena).

➤ Comando de sincronización de reloj entre servidor y cliente.

Este comando permite al servidor por una parte solicitar al cliente (el sistema de control central) su hora local, y al cliente enviar su hora al servidor. La hora de los sistemas para este caso se está en unidades de milisegundos, y mide el tiempo que los sistemas han permanecido encendidos (Timer multimedia de Windows). Para propósitos prácticos del sistema, no conviene sincronizar los relojes en tiempo real, sino más bien sincronizar cronómetros de alta precisión por motivos de determinar los momentos en que se debe realizar la conmutación de las señales de video (toda medición de tiempos es relativa a la hora de encendido de los equipos).

Segmento	Campo	Contenido	Tamaño
Header	Id. del comando	Numero que identifica a este comando (valor de 1).	4 bytes
Comando de sincronización de reloj	Hora del servidor remoto	Lectura del cronometro del servidor a la hora del envío.	4 bytes
	Hora del cliente (sistema de control)	Lectura del cronometro del servidor a la hora de la responder.	4 bytes
Checksum	CRC-32	Número usado para comprobar el contenido, generado a partir de los dos segmentos anteriores.	4 bytes

**Tabla 7.17 – Contenido del comando de sincronización.**

Este comando es enviado de forma periódica por el servidor, a un período ajustable (ver sección del software de control del sistema remoto). El paquete de ida es idéntico al de retorno, conteniendo únicamente el valor de reloj del servidor. El cliente contesta con el mismo comando, pero agrega la lectura de su propio reloj a la estructura (reemplazando cualquier dato que el servidor hubiera puesto en ese campo, normalmente 0), de manera que cuando el paquete llega de regreso al servidor, este contiene tanto la hora de salida como la hora del cliente, y junto con la hora de llegada es posible determinar el tiempo de viaje del comando y así extrapolar el reloj actual del cliente en ese instante. Más adelante se verá con mayor detalle la técnica de sincronización de relojes utilizada.

➤ Comando de envío (y reproducción) de un bloque de videos.

<b>Segmento</b>	<b>Campo</b>	<b>Contenido</b>	<b>Tamaño</b>
Header	Id. del comando	Numero que identifica a este comando (valor de 2).	4 bytes
Comando de reproducción de bloque de videos	Hora de inicio del bloque	Hora relativa al reloj del cliente a la que el bloque de videos debe comenzar.	4 bytes
	Cantidad de videos	Número de videos contenidos en el comando.	4 bytes
Bloque de descripción de los videos	Descriptores de videos	Arreglo de descriptores de videos con una cantidad de entradas igual al número de videos en el comando.	variable
Bloque de rutas de archivo de los videos	Rutas de los videos	Contiene una cadena por cada video pasado en el comando, cada cadena indica la ruta del archivo de video a reproducir en orden correlativo a los descriptores de video.	Variable.
Checksum	CRC-32	Número usado para comprobar el contenido, generado a partir de todos los segmentos anteriores.	4 bytes

**Tabla 7.18 – Contenido del comando de envío de un bloque de videos.**

Este comando permite al cliente solicitar al servidor que reproduzca un bloque específico de videos en un momento determinado (interrumpiendo para ello la señal piloto). Todo el bloque de videos es pasado en un solo comando, y se incluye toda la información necesaria para reproducir cada video. El sistema remoto se encarga de todos los arreglos necesarios para reproducir el bloque antes de comenzar y después de terminar el mismo. En caso de enviarse más de un bloque de videos a la vez, el servidor remoto los ubica en cola en el orden de llegada. A continuación se muestra el contenido de cada descriptor de videos:

Segmento	Campo	Contenido	Tamaño
Descriptor de videos	Duración del video	Duración en milisegundos del video.	4 bytes
	Numero de bytes de la ruta	Cantidad de bytes que componen la cadena de la ruta asociada al video dentro el bloque de nombres de archivo.	4 bytes

**Tabla 7.19 – Contenido del descriptor de video.**

El comando completo contiene un descriptor por cada video, asimismo, el bloque de nombres de archivo contiene una cadena con una ruta por cada video (cada cadena se corresponde con su descriptor en orden correlativo). En caso que un video se repita en el bloque, se hace explícitamente y la información se envía replicada sin ningún esquema de compresión. Por último vale aclarar que las cadenas del bloque de rutas de archivo van colocadas una inmediatamente detrás de la otra sin alineación ni padding<sup>26</sup>, y en formato ASCIIZ.

➤ Comando de notificación de eventos.

Este comando tiene la finalidad de transportar informes acerca de la actividad del servidor al cliente, con el propósito de informar al usuario acerca del funcionamiento del sistema remoto y de poder llevar un registro de eventos en el sitio de control. Las notificaciones están clasificadas de acuerdo a códigos numéricos, esto con el propósito de reducir la cantidad de datos transmitidos y de poder facilitar el manejo de cada evento en el software del cliente. Asimismo, cada notificación puede acarrear consigo una cadena de texto opcional (en formato ASCIIZ), con información adicional de la notificación. Esta cadena de texto tiene un significado que va de acuerdo a la naturaleza de la notificación.

---

<sup>26</sup> El “padding” o relleno es una técnica que permite alinear los datos de las estructuras de tamaño variable en bloques de tamaño fijo por medio de agregar datos extra hasta completar un tamaño predefinido, esto con el propósito de lograr una mayor conveniencia y facilidad en el manejo de los datos. Esta técnica no es usada en este sistema con el propósito de ahorrar ancho de banda.

Segmento	Campo	Contenido	Tamaño
Header	Id. del comando	Numero que identifica a este comando (valor de 3).	4 bytes
Comando de notificación de eventos	Código de notificación	Valor numérico que indica tipo de evento ocurrido.	4 bytes
	Numero de bytes de la cadena	Cantidad de bytes que componen a la cadena con la descripción adicional del evento (valor de 0 en caso que la cadena este ausente).	4 bytes
	Cadena con información adicional	Cadena opcional con información que varía según el tipo de notificación.	Variable
Checksum	CRC-32	Número usado para comprobar el contenido, generado a partir de los dos segmentos anteriores.	4 bytes

**Tabla 7.20 – Contenido del comando de notificación de eventos.**

En la siguiente tabla se detallan los códigos de notificación:

Notificación	Código numérico (decimal)	Información adicional en la cadena
Bloque de videos recibido exitosamente	0	Ninguna
Video reproducido exitosamente	1	Ruta del video
Error al cargar el video	2	Ruta del video
Error al reproducir el video	3	Ruta del video
El modo piloto ha sido forzado en el sitio remoto.	4	Ninguna
El modo piloto dejo de ser forzado en el sitio remoto.	5	Ninguna

**Tabla 7.21 – Códigos de notificación del sistema remoto.**

### **Algoritmo de sincronización de hora entre el cliente y el servidor.**

Debido a la naturaleza de la inserción de contenido de video en vivo en el sistema, se hace necesario contar con un mecanismo relativamente preciso de sincronización entre el sistema central y los servidores en los sitios remotos. Sin embargo, debido a la naturaleza de los osciladores, es imposible mantener una frecuencia y una sincronía perfecta en dos o más sistemas separados, ya que siempre, sea a mediano o largo plazo, ocurrirá un desplazamiento de los relojes, lo cual afecta considerablemente la precisión de las operaciones de inserción de videos. Este fenómeno conocido como corrimiento de fase es la principal fuente de error en todo sistema con temporización distribuida.

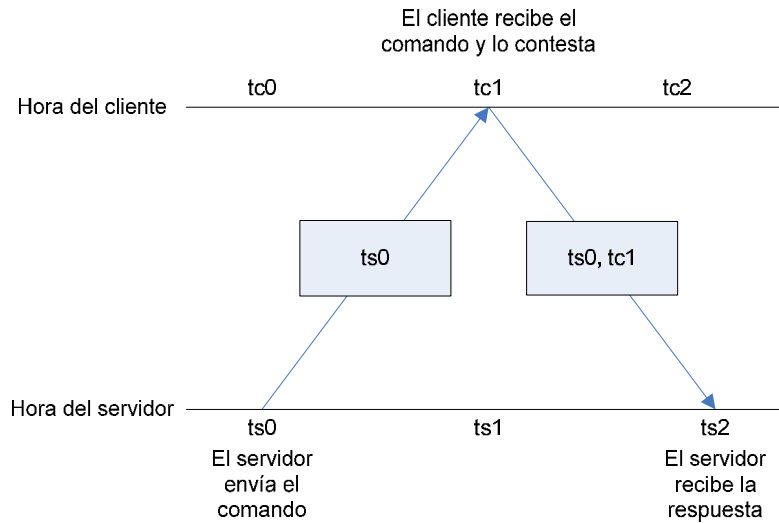
Tenemos pues, que cada servidor remoto cuenta de forma natural un cronómetro con una precisión al nivel de los milisegundos (basado en un oscilador de cristal dentro de la tarjeta madre del sistema), el cual sirve de base de tiempo local. Sin embargo debido al corrimiento de fase, la relativamente baja precisión de los cristales y a la alta frecuencia de operación de los mismos, el adelanto o atraso de relojes se hace evidente en lapsos cortos (cuestión de minutos, o aún segundos). Por lo tanto, se vuelve necesario poder realizar un ajuste o compensación automática al ritmo de marcha de los relojes del sistema para que avancen a un único ritmo con respecto a una hora principal.

Por lo tanto, en vista a esta necesidad, se ha elaborado un mecanismo que permite a cada servidor en cada sitio remoto sincronizarse a la hora del sitio central (el único cliente del sistema), para mantener un sistema de hora referenciado a una sola fuente. Dicha mecánica se vale de la comunicación por red de datos (Internet) que sostienen los nodos del sistema entre sí, aprovechando la topología en forma de estrella que posee el sistema (con el cliente de todos los servidores como nodo central). Bajo este esquema, los servidores son responsables de mantener un reloj en marcha, dicho reloj tiene una hora que es una estimación de la hora del sistema central, misma que es determinada en base a lecturas de tiempo realizadas a través de la red.

Los algoritmos utilizados para la sincronización de los sistemas son dos: El primero determina la hora del cliente por medio de una extrapolación y provee una lectura junto con

una tasa de error, mientras que el segundo toma todas las lecturas recientes y las promedia en forma ponderada, dando una mayor ponderación a las lecturas con error menor.

La técnica de lectura de la hora del cliente es una implementación del algoritmo de Cristian, que asume que los relojes locales de ambos sistemas son muy precisos durante el intervalo de medición (el cual depende de la latencia de la red y se encuentra en el orden de las decenas de milisegundos). Luego, la técnica consiste en enviar un paquete marcado con la hora de salida del servidor, el cual es contestado por el cliente que le agrega su hora local, y que finalmente es recibido nuevamente por el servidor que lo vuelve a marcar con su hora de retorno.



**Figura 7.22 – Algoritmo de Cristian.**

A partir del diagrama, se observa que al momento de recibir la respuesta, el servidor cuenta con 3 lecturas de tiempo: La hora del servidor a la que se envió el comando, la hora del cliente a la que el comando fue contestado, y la hora del servidor a la que se recibió la respuesta.

Luego, el tiempo completo de viaje es:

$$tv = ts2 - ts0$$

La hora del cliente se encuentra en algún momento entre la hora de salida ( $ts_0$ ) y la hora de retorno ( $ts_2$ ), por lo que se asume que la hora a la que llego está aproximadamente a la mitad del tiempo completo de viaje ( $ts_1$ ). Luego, como no se sabe realmente los tiempos de ida y de vuelta, pero como se asume que la contestación ocurrió en algún momento intermedio, se deduce que el error de la lectura debería ser:

$$\Delta t_{err} = \pm tv/2$$

Por lo tanto, tenemos que la hora del cliente para el momento en que el servidor recibe el debería ser la hora que envió el cliente más la mitad del tiempo total de viaje:

$$tc_2 \cong tc_1 + tv/2$$

$$tc_2 \cong tc_1 + \frac{ts_2 - ts_0}{2}$$

Agregando el factor de error, la estimación de la hora  $tc_2$  en el instante  $ts_2$  es:

$$tc_2 = tc_1 + \frac{ts_2 - ts_0}{2} \pm \frac{ts_2 - ts_0}{2}$$

Estas lecturas son tomadas en forma periódica, y a medida son tomadas, las lecturas anteriores son adelantadas en el tiempo usando como referencia el reloj local para aproximar la hora actual del cliente en todo momento.

El algoritmo de estimación consistirá entonces en promediar las ultimas lecturas (cuya cantidad puede ser variada en la configuración de temporización del programa), siendo cada lectura la representación de la hora actual del cliente y con propia su tasa de error.

Luego, el coeficiente de ponderación se calcula de la siguiente manera:

$$Cp = tv_{max} - tv$$

Donde  $tv_{max}$  representa el tiempo de viaje máximo tolerado (parámetro configurable en el sistema), si la lectura supera ese intervalo, la misma queda descartada, pues su coeficiente sería negativo.

Luego, la hora actual del cliente en base a las últimas lecturas (corridas todas al tiempo actual) es:

$$tc_{estimado} = \frac{\sum_{n=1}^{nl} tc_n * Cp_n}{\sum_{n=1}^{nl} Cp_n}$$

Donde  $nl$  es el número total de lecturas almacenadas,  $tc_n$  la  $n$ -ésima lectura de tiempo y  $Cp_n$  el coeficiente de la  $n$ -ésima lectura.

Finalmente, con la estimación de la hora actual ( $tc_{estimado}$ ) se hace una corrección del cronómetro con la hora actual del cliente según el servidor ( $tc_{servidor}$ ), de acuerdo a las siguientes reglas:

- Si  $tc_{estimado} > tc_{servidor}$  se avanza  $tc_{servidor}$  al doble de ritmo.
- Si  $tc_{estimado} < tc_{servidor}$  se avanza  $tc_{servidor}$  a la mitad del ritmo.
- Si  $tc_{estimado} \sim tc_{servidor}$ , se avanza  $tc_{servidor}$  al ritmo normal.

Se hace la observación de que el cronómetro del servidor que lleva la hora del cliente no se corrige nunca en forma inmediata, porque podrían producirse saltos bruscos a horas futuras (perdiendo los instantes exactos en que deberían darse los eventos) o bien saltos a horas pasadas (provocando que eventos futuros ocurran antes de su tiempo). Por ello se hacen ajustes suaves de la velocidad del reloj, de forma que sea incrementado monótonamente (siempre hacia delante) en todo momento, pero tratando de ajustarse a la hora estimada del cliente.

Finalmente, la hora del cliente según el servidor es inicializada en un momento inicial a partir de la primera lectura (se hace una asignación directa del valor), para luego comenzar a partir de ahí con el proceso de corrección y ajuste de la hora.

#### 7.1.2.4 Programa administrativo local (interfaz de usuario manual).

En este apartado se hará una descripción breve de cómo se utiliza el programa que permite a un usuario local (ubicado en el sitio remoto donde se encuentra el equipo de transmisión) administrar la reproducción de los videos. Cabe señalar que el sistema está diseñado para ser operado completamente a distancia, requiriéndose la presencia del personal técnico en el sitio remoto sólo en caso de situaciones especiales que requieran la presencia física. Por lo tanto, este sistema deberá ser considerado como un sistema administrativo de emergencia o de respaldo, en caso que se pierda la conexión a Internet en el sitio remoto por un período relativamente prolongado.

La ventana principal del programa junto con la descripción de los elementos que contiene se presenta a continuación:

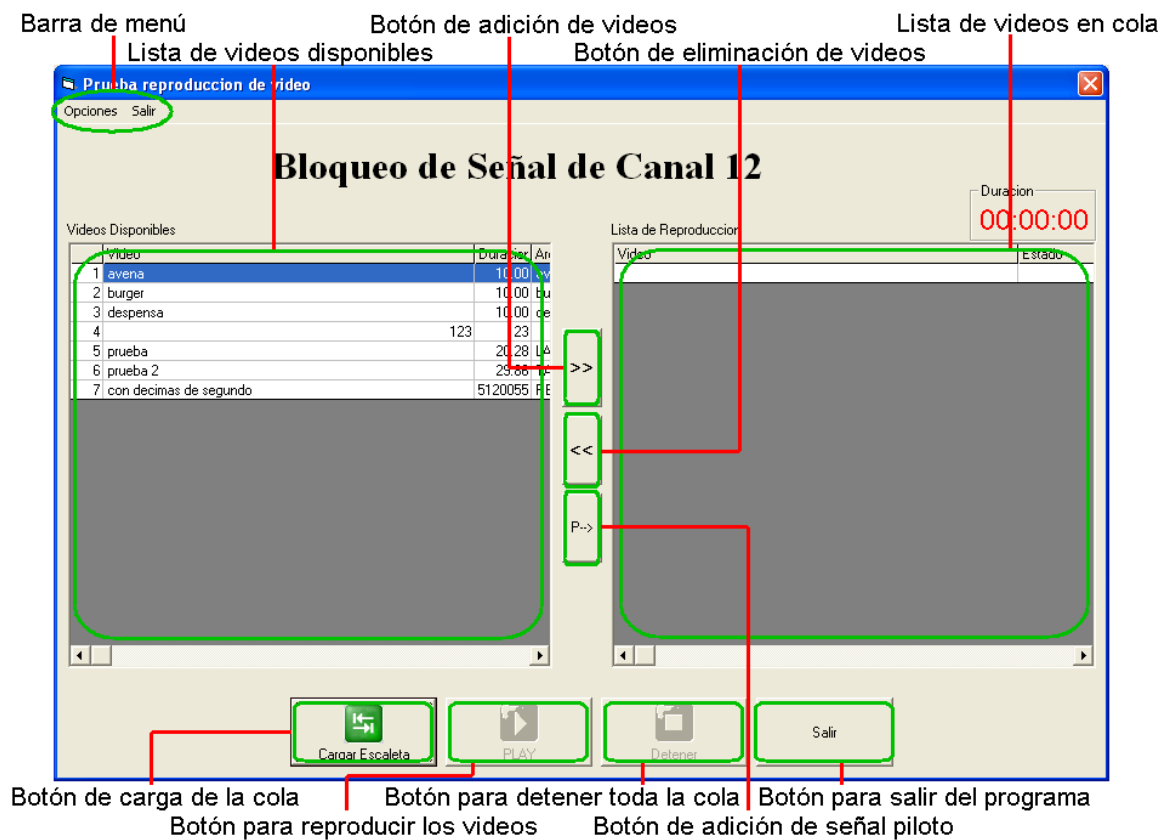
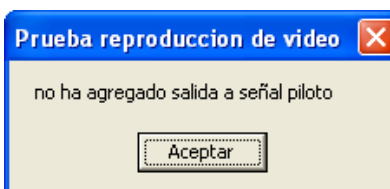


Figura 7.23 – Descripción de los controles de programa administrativo local.

El sistema ha sido diseñado para ser manejado de una forma sencilla, siendo prácticamente un reproductor de videos digital. Para que el sistema opere correctamente, se debe de indicar con anticipación cuales videos deberá reproducir así como el orden en que deberán pasar, para que luego el sistema reproduzca los videos automáticamente de acuerdo a la lista. Para colocar los videos en cola y reproducirlos, se deben seguir los siguientes pasos:

- Seleccione el primer video que desee reproducir de la lista de videos disponibles, y haga clic en el botón de adición de videos (>>).
- Repita el paso anterior para cada uno de los demás videos que desee reproducir. Los videos serán reproducidos en el mismo orden que son agregados, así que el orden de selección de los mismos es importante.
- Si ha cometido un error y desea eliminar el video de la cola, seleccione el video de la lista de videos en cola, luego presione el botón de eliminación de videos (<<).
- Una vez agregados todos los videos, presione el botón de adición de señal piloto (P->). Automáticamente se generará una entrada en la cola que indica que el sistema conmutará a la señal piloto una vez se ha reproducido el último video de la cola. En caso que olvide agregar la señal piloto al final de la lista, aparecerá el siguiente mensaje recordatorio:



**Figura 7.24 – Cuadro de mensaje que recuerda al usuario que no se ha agregado la señal piloto.**

- Con la lista de reproducción terminada, presione el botón de carga de la cola (cargar escaleta). Con esto se activará el botón para reproducir los videos (PLAY). A continuación se muestra un ejemplo de cómo se mira la ventana de la aplicación hasta este punto:



**Figura 7.25 – Ejemplo de la ventana del programa de control local cuando está listo para reproducir.**

- En el momento exacto indicado para insertar el contenido televisivo, presione el botón para reproducir los videos (PLAY). El sistema bloqueará automáticamente la señal piloto proveniente del estudio, y reproducirá los videos en el orden dispuesto en la cola, uno tras otro hasta el último, volviendo automáticamente a la señal piloto una vez termine.

Adicionalmente el programa está provisto un cuadro de diálogo de opciones, accesible directamente desde la barra de menú. Actualmente el cuadro de opciones contiene únicamente el control de volumen de salida de la tarjeta decodificadora de videos, el cual se muestra a continuación. No está demás indicar que para cambiar el volumen de la tarjeta decodificadora bastará con arrastrar el control hasta la posición deseada.

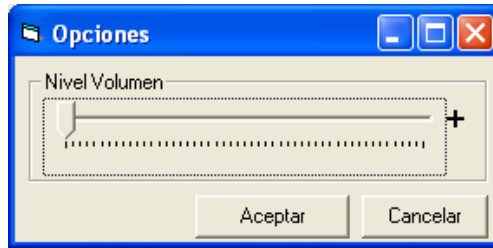


Figura 7.26 – Cuadro de diálogo de opciones del programa administrativo local.

## 7.2 Sistema de control de la estación local.

El sistema de control local lo comprenden todos los equipos y programas que residen en el sitio de control central, ubicado en las instalaciones principales de la empresa en Santa Elena, así como también los programas que estos ejecutan. El diagrama en bloques de este sistema se presenta a continuación:

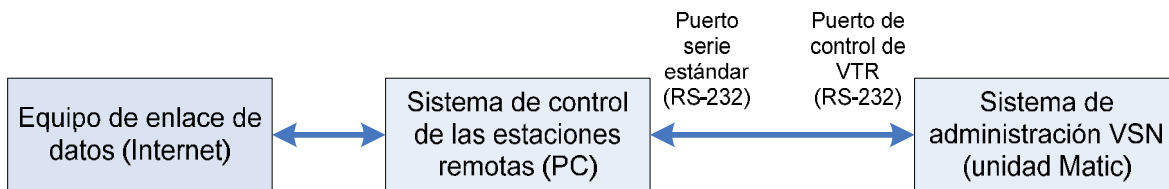


Figura 7.27 – Diagrama en bloques del sistema de control local.

Como puede verse, en esta sección el sistema es bastante simple. Todos los servidores remotos son manejados desde una única computadora con conexión a Internet. El elemento de interés aquí es la conexión que se hace al sistema VSN<sup>27</sup>, con el cual se puede obtener información de los momentos en que deben hacerse los cortes de la señal piloto, para insertar el contenido de las estaciones remotas.

---

<sup>27</sup> VSN es el nombre del sistema con que cuenta el Canal 12 para administrar todo el contenido televisivo que se transmite al aire. La integración de este sistema al VSN se hace evidente, ya que la señal piloto que es interrumpida en los sitios remotos es generada bajo el control de este mismo sistema.

La mecánica por medio de la cual se determinan los momentos en los cuales hacer los cortes consiste en conectar la computadora de control de los servidores remotos al VSN por medio de un puerto de control de VTR<sup>28</sup>, de tal suerte que el VSN notifica al sistema de control de los momentos en que se hacen los cortes por medio de un comando de pausa, que al ser recibido provoca una serie de eventos que desemboca en que se pasen los videos adecuados en las estaciones remotas en los momentos correspondientes. Así pues, la computadora dedicada al control hace las veces de una VTR emulada, comportándose como tal ante el VSN.

Un aspecto importante a señalar es que el sistema VSN es capaz de generar los comandos de VTR sólo al momento de iniciar un bloque o video, de tal manera que si se intenta reproducir los comerciales en el instante en que el comando de VTR es recibido, entonces los comandos llegarán muy tarde a los sitios remotos debido a la latencia de la red. Por tanto, es importante notificar a los servidores remotos de que los bloques de videos serán reproducidos con anticipación.

Así pues, tenemos que el sistema VSN notifica al sistema de control “un video antes” de realizar el corte, lo que da tiempo suficiente para enviar el comando de reproducción de videos a cada estación remota y al mismo tiempo compensar por la latencia de red (ver las secciones anteriores correspondientes al protocolo de comunicación).

Luego, tenemos que las funciones de este sistema son:

- Proveer al usuario del sistema una interfase que permita controlar el contenido televisivo que sustituye a la señal piloto en cada estación remota.
- Determinar, con la entrada de los comandos de VTR por parte del VSN, los momentos en que los bloques de videos serán reproducidos en cada estación remota.
- Llevar un registro de todos los eventos que reportan los servidores remotos que componen el sistema.

---

<sup>28</sup> Una VTR (Video Tape Recorder) es una unidad reproductora de videos digital que utiliza cinta magnética.

### **7.2.1 Software de control de la estación local.**

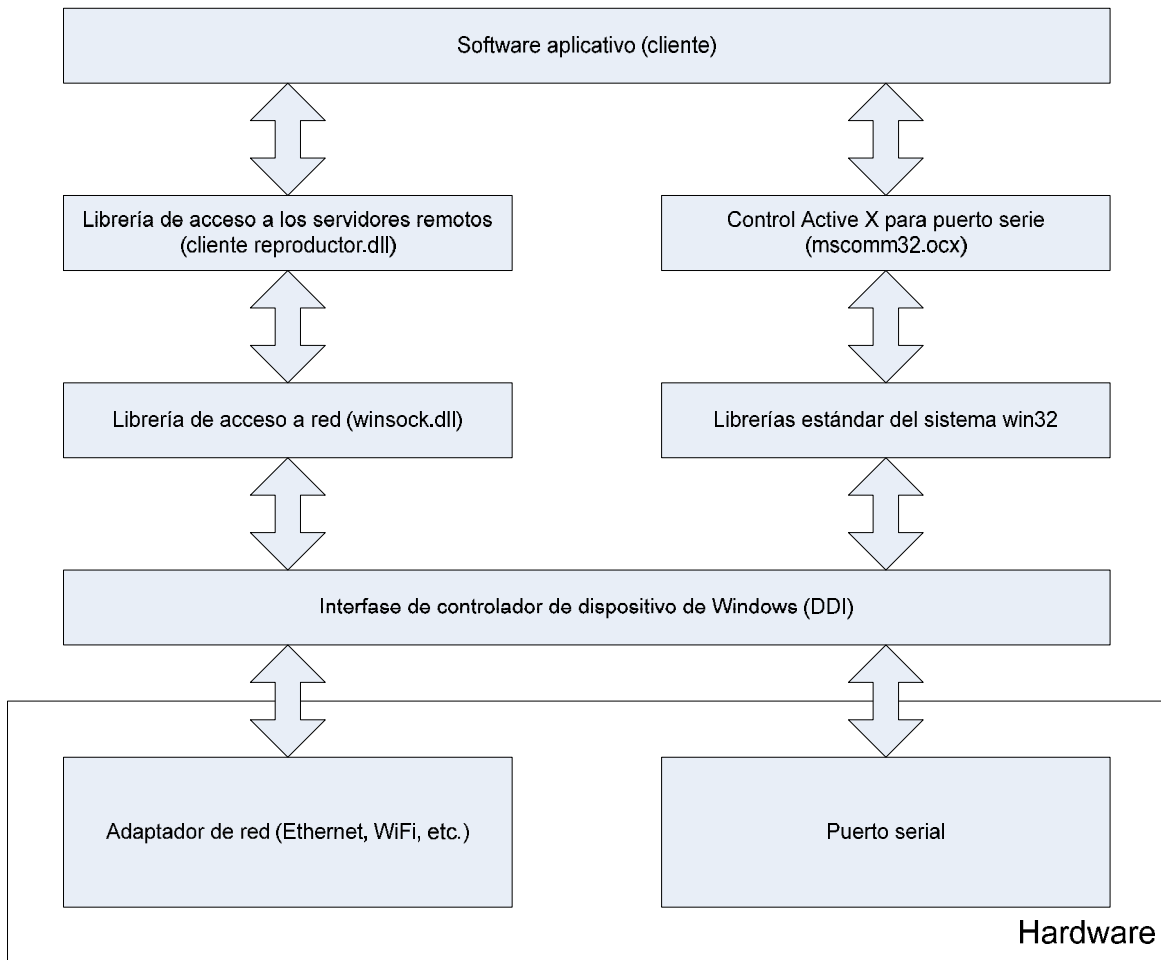
El software de control local comprende toda la gama de programas que residen dentro de la PC en el sitio de control. Sus tareas se enfocan principalmente a la organización de los servidores remotos, así como el contenido que estos reproducen.

La figura siguiente muestra la jerarquía general del software instalado en la PC del sitio de control. El software aplicativo, que se encuentra en el nivel superior, provee la interfase visible al usuario y permite al mismo administrar el contenido que es reproducido en cada servidor, asimismo, éste lleva control de la temporización de los bloques de videos, manteniendo para todo esto, una conexión activa con cada servidor de video.

A diferencia de los servidores remotos, la conectividad de red se encuentra encapsulada dentro de una DLL, esto con el objetivo de flexibilizar el software aplicativo que gestiona la administración. Así pues, tenemos que puede usarse cualquier herramienta de desarrollo que se estime conveniente para realizar el sistema administrativo, mientras que la conectividad de red, que es una sección relativamente compleja, queda como un segmento fácilmente reutilizable y compatible con todo lenguaje. Para este caso, se considera que el sistema administrativo se encuentra desarrollado bajo Visual Basic 6.0

La presencia del control mscomm32.ocx en la jerarquía se debe a que el mismo es utilizado dentro del lenguaje Visual Basic para tener acceso a las funciones del puerto serie. Asimismo, la librería winsock.dll es usada a su vez por la DLL de acceso a los servidores para gestionar todas las comunicaciones por la red de datos (Internet).

Finalmente, las capas inferiores de la jerarquía corresponden a los niveles inferiores del sistema operativo Windows (DDI), las cuales por medio de los controladores permiten el acceso al hardware.



**Figura 7.28 – Jerarquía del software y hardware de la estación de control.**

### 7.2.1.1 Librería (DLL) de conexión para el cliente del reproductor remoto.

Como se mencionó anteriormente, la conexión a los servidores remotos se realiza con la ayuda de una librería DLL especialmente dedicada a dicho fin. En este apartado se describirá la interfase de esta librería, la cual gestiona automáticamente todos los detalles concernientes a la conexión, incluyendo la respuesta automática a los comandos que envían los servidores remotos. A continuación se detallarán las funciones que exporta la DLL.

#### ➤ Función de inicialización de la DLL.

Sintaxis:

```
CodigoError = InicializarClienteReproductor(hVentanaPadre, hControl)
```

Esta función permite inicializar la Librería DLL, por lo que debe ser llamada antes de iniciar cualquier otra operación con la misma. Los argumentos `hVentanaPadre` y `hControl` definen los handles<sup>29</sup> (manejadores) de la ventana principal del programa cliente y del control de notificación. El control de notificación debe ser un botón convencional, el cual debe estar oculto (propiedad “visible” en valor falso o ausente). La DLL envía mensajes de clic de dicho botón hacia la ventana principal del programa cuando desea avisar al programa principal que hay una notificación pendiente de ser retirada. Si alguno o ambos handles son nulos, entonces la DLL no genera mensajes de clic y por tanto es responsabilidad del programa retirar las notificaciones por medio del método de polling o encuesta (ver más adelante la función de retiro de mensajes). El valor de retorno es un código de error que indica el nivel de éxito (o fracaso) de la función cuando es invocada.

➤ Función de desalojo de la DLL.

Sintaxis:

```
LiberarClienteReproductor()
```

Esta función permite desalojar la librería DLL, y debe ser llamada antes que el programa termine, para evitar el desperdicio de recursos y errores potenciales posteriores. Esta función no posee valor de retorno.

➤ Función de retiro de notificaciones.

Sintaxis:

```
CodigoError = RetirarNotificacion(NotificacionCR)
```

Esta función permite retirar cualquier notificación que pudiera estar pendiente en la cola de notificaciones de la DLL. El argumento `NotificacionCR` es una estructura pasada por referencia que es llenada con la información de la notificación al retorno. El valor de

---

<sup>29</sup> En los sistemas operativos, muchos de los recursos públicamente accesibles utilizan la modalidad de manejadores o “handles” para designar las instancias de los recursos asignados a las aplicaciones. Un handle no es más que un número entero que el sistema operativo asigna un recurso en particular, para identificarlo durante los procesos que lo manipulan.

retorno es un código de error que indica el nivel de éxito (o fracaso) de la función cuando es invocada y en caso de que se use la técnica de polling (sin mensajes de clic) para retirar las notificaciones, indica con un código especial que no existen notificaciones en el momento que la función fue llamada.

La estructura pasada por referencia tiene los siguientes miembros:

<b>Nombre del miembro</b>	<b>Descripción.</b>
IdConexión	Identificador asociado al servidor de video que generó el evento.
CodigoNotificacionCR	Código de la notificación.
CadenaDetalle	Cadena con información adicional relacionada a la notificación.

**Tabla 7.22 – Estructura de la notificación de la DLL de conexión a servidores remotos.**

Los códigos de notificación posibles se listan a continuación:

<b>Nombre</b>	<b>Valor</b>	<b>Descripción</b>
CNCR_BLOQUE_ACEPTADO	0	El servidor remoto recibió el bloque de videos con éxito.
CNCR_VIDEO_REPRODUCIDO	1	El servidor remoto reprodujo exitosamente un video. La cadena con información adicional contiene la ruta del video.
CNCR_VIDEO_NO_CARGADO	2	El servidor remoto no pudo cargar el archivo de video (posiblemente invalido o no existe). La cadena contiene la ruta del video.
CNCR_ERROR_REPRODUCCION	3	El servidor remoto no pudo reproducir el video (error interno). La cadena contiene la ruta del video.
CNCR_PILOTO_FORZADA	4	Un usuario activó el modo de señal piloto forzada en este servidor.
CNCR_PILOTO_NORMAL	5	El modo de señal piloto forzada ha sido desactivado.
CNCR_CONEXION_PERDIDA	6	El servidor remoto cerró la conexión.

Nombre	Valor	Descripción
CNCR_CONEXION_ABORTADA	7	La conexión con el servidor se perdió por un error en la red

**Tabla 7.23 - Códigos de notificación de la DLL de conexión a servidores remotos.**

- Función de conexión a un servidor remoto vía nombre de host.

Sintaxis:

```
CodigoError = IniciarConexionSitioRemotoHN( IdConexion, NombreHostRemoto,
                                             PuertoRemoto, Interfase)
```

Esta función permite iniciar una nueva conexión a un servidor remoto, identificando al mismo por su nombre de host (que bien puede ser un nombre que pertenece a un dominio DNS). El argumento `IdConexion` identifica la conexión particular con el servidor remoto en cuestión, y debe ser un número entre 0 y 15 que se elige arbitrariamente; Todas las transacciones asociadas a ese servidor deberán usar ese mismo ID para referirse al servidor remoto y pueden realizarse hasta 16 conexiones simultáneas. Los argumentos `NombreHostRemoto` y `PuertoRemoto` contienen la información referente al servidor, mientras que `Interfase` contiene la dirección IP del adaptador o interfase de red por medio de la cual se entablará la conexión (en caso de pasarse la IP 0.0.0.0 en este argumento, se elige el adaptador de red de forma automática). El valor de retorno es un código de error que indica el nivel de éxito (o fracaso) de la función cuando es invocada.

- Función de conexión a un servidor remoto vía dirección IP.

Sintaxis:

```
CodigoError = IniciarConexionSitioRemotoIP( IdConexion, IPRemota,
                                             PuertoRemoto, Interfase)
```

Esta función opera de la misma manera que la anterior, con la distinción de que ahora se identifica al servidor remoto por su dirección IP en vez de su nombre de host. Dicha dirección es pasada en el argumento `IPRemota`.

- Función de desconexión de un servidor remoto.

Sintaxis:

```
CodigoError = TerminarConexionSitioRemoto(IdConexion)
```

Esta función permite finalizar la conexión con el servidor remoto especificado en el parámetro `IdConexion`. El valor de retorno es un código de error que indica el nivel de éxito (o fracaso) de la función cuando es invocada.

- Función de reconexión a un servidor remoto.

```
CodigoError = ReiniciarConexionSitioRemoto(IdConexion)
```

Esta función permite restablecer una conexión en caso que sea perdida, con el servidor remoto designado por el argumento `IdConexion`. Todos los códigos de retorno asociados a las funciones de conexión normales aplican para esta función.

- Función de adición de videos a al bloque actual.

```
CodigoError = AgregarVideoEnBloque( IdConexion, RutaVideo,  
                                     DuracionMilisegundos)
```

El trabajo de esta función es la de agregar los videos especificados al bloque de videos actual. Los bloques de videos son independientes para cada conexión, y están inicialmente vacíos. El argumento `IdConexion` especifica el servidor para el cual se agrega un video al bloque, mientras que `RutaVideo` y `DuracionMilisegundos` contienen la información asociada al video en sí. Una vez todos los videos son agregados al bloque actual para un servidor, pueden ser despachados con la siguiente función. El valor de retorno es un código de error que indica el nivel de éxito (o fracaso) de la función cuando es invocada.

- Función de reproducción del bloque actual de videos.

Sintaxis:

```
CodigoError = ReproducirBloque(IdConexion, RetardoMilisegundos)
```

Esta última función permite despachar un bloque de videos previamente elaborado con la función anterior al servidor de videos designado por el argumento IdConexion. El argumento RetardoMilisegundos indica cuánto tiempo hay que esperar (a partir del momento en que se llama esta función), para reproducir el video. La DLL junto con el software en el otro extremo harán la compensación de latencia necesaria para garantizar que el video sea reproducido en el momento adecuado en el servidor remoto. El valor de retorno es un código de error que indica el nivel de éxito (o fracaso) de la función cuando es invocada.

Finalmente, la siguiente tabla lista todos los posibles códigos de retorno para todas las funciones anteriores:

<b>Nombre</b>	<b>Valor</b>	<b>Descripción</b>
CECR_EXITO	0	Operación completada exitosamente
CECR_YA_INICIALIZADO	1	DLL o subcomponente ya inicializado
CECR_NO_INICIALIZADO	2	DLL o subcomponente sin inicializar
CECR_ARGUMENTO_INVALIDO	3	Argumento de función invalido
CECR_MEMORIA_INSUFICIENTE	4	No se pudo alojar memoria
CECR_CREAR_EVENTO	5	No se pudo crear un objeto de evento
CECR_CREAR_THREAD	6	No se pudo crear un thread
CECR_INICIALIZAR_WINSOCK	7	No se pudo inicializar la librería de Windows Sockets
CECR_VERSION_WINSOCK_MUY_ANTIGUA	8	La versión de Windows Sockets detectada es muy antigua
CECR_OBTENER_NOMBRE_HOST_LOCAL	9	Imposible leer el nombre del host local
CECR_YA_CONECTADO	10	Cliente de red ya conectado
CECR_NO_CONECTADO	11	Cliente de red no conectado aun
CECR_CREAR_SOCKET	12	No se pudo crear un socket al conectar
CECR_ASOCIAR_INTERFASE	13	No se pudo asociar el adaptador de red

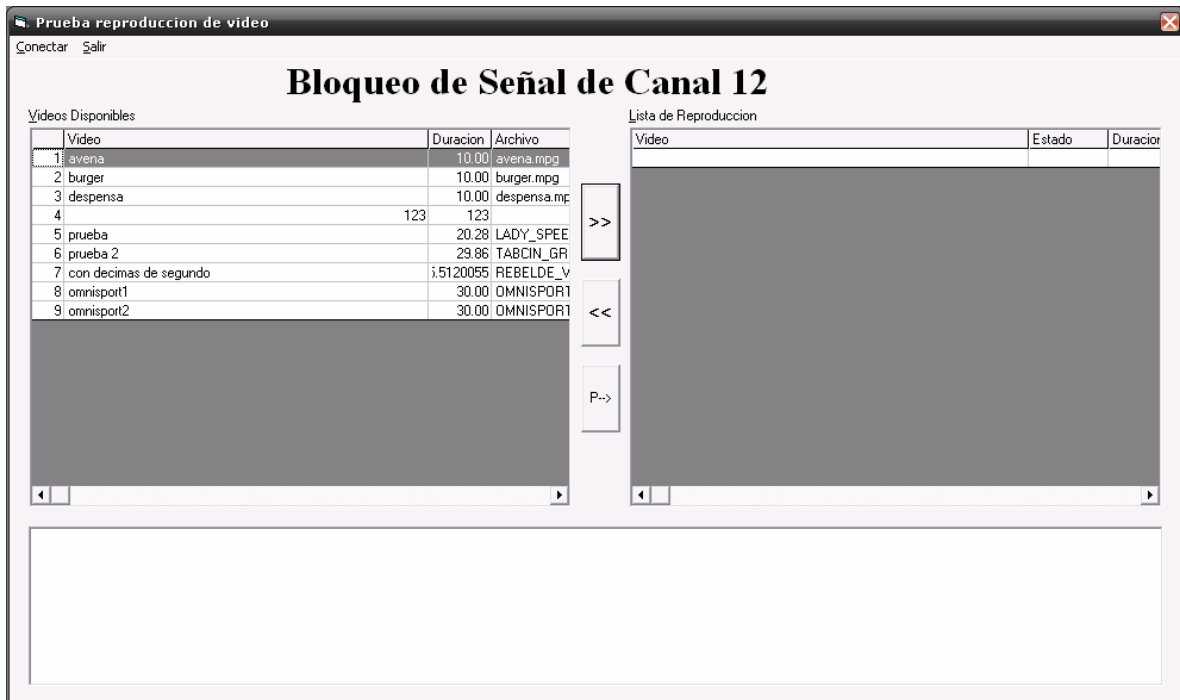
<b>Nombre</b>	<b>Valor</b>	<b>Descripción</b>
CECR_RESOLVER_IP_HOST	14	Imposible resolver nombre del host a una IP
CECR_INICIAR_CONEXION	15	Imposible conectar con el servidor
CECR_FALLO_RED	16	Ha ocurrido un error en la red
CECR_ENVIAR_DATOS	17	Error al enviar datos
CECR_RECIBIR_DATOS	18	Error al recibir datos
CECR_CONEXION_ABORTADA	19	La conexión se ha perdido
CECR_CONEXION_CERRADA	20	El servidor cerro la conexión
CECR_BLOQUE_VACIO	21	El bloque de videos esta vacío
CECR_SIN_NOTIFICACIONES	22	No hay notificaciones para retirar
CECR_DESCONOCIDO	23	Error desconocido

**Tabla 7.24 – Códigos de retorno de las funciones de la DLL de conexión a servidores remotos.**

### 7.2.1.2 Software de control y administración local.

En este apartado se procederá a describir el software administrativo que se ejecuta en el sitio de control, el cual permite al usuario administrar los videos que se reproducen en una estación remota.

En la figura siguiente se muestra la apariencia de la interfase de usuario del programa administrativo. Como puede verse, su diseño es muy similar al programa de control local. Asimismo, su uso es muy similar, ya que los bloques de videos son construidos de la misma manera que el programa anterior, agregando los videos disponibles de la lista de la izquierda usando los botones centrales, y luego agregando los segmentos de señal piloto al final de los bloques.



**Figura 7.29 – Interfase del programa administrativo del sitio de control.**

Sin embargo, existen diferencias fundamentales con respecto al otro sistema, particularmente que ahora es posible generar más de un bloque de videos por medio de agregar varios bloques de señal piloto y que los botones de reproducción (inicio, paro y carga de escaleta) han sido omitidos.

Asimismo ahora existe una ventana de registro, la cual muestra los eventos más recientes ocurridos en el servidor remoto (gracias a los comandos de notificación del servidor). Por otra parte, los bloques son ahora activados por la generación de un comando de “pausa” para VTR, el cual es enviado por el sistema VSN.

Así pues, tenemos que cuando el programa percibe el comando de pausa a través del puerto serie envía el bloque de videos al frente de la cola, el cual esta delimitado por el primer bloque de señal piloto. Es posible pasar un bloque vacío al servidor, por medio de colocar solamente el bloque de piloto. De esta forma, el sistema hace caso omiso del siguiente comando de pausa y no reproduce video alguno en el servidor remoto.

En el momento en que se recibe el comando de pausa de el VSN, el software envía el bloque al servidor remoto de videos, indicándole que reproduzca dicho bloque dentro de una cantidad definida de tiempo (20 segundos), por lo tanto, el sistema VSN deberá ser configurado para que emita el comando esa misma cantidad de tiempo antes de que sea el verdadero momento de reproducir los videos (un video de 20 segundos antes del bloque de comerciales insertado).

En la barra de menú se encuentra la opción de conectarse al servidor remoto. Al activar esta opción, el programa intentará conectarse al servidor remoto definido en el archivo de configuración. El archivo de configuración puede ser modificado para configurar la conexión a cualquier servidor, bastará para ello editar el archivo y definir la IP así como el número de puerto a utilizar (los números de puerto deben coincidir en ambos extremos y no deben ser usados por ningún otro programa en ambos sistemas).

Como puede apreciarse, el proceso de reproducción de videos es semiautomático, requiriéndose nada más que el usuario defina los bloques de videos para cada inserción y definiendo en el sistema VSN los momentos en que los bloques deban ser enviados. De esta forma es posible elaborar la escaleta con los videos con varias horas de anticipación y permitir que el sistema funcione prácticamente solo durante intervalos relativamente prolongados.

## **VIII. Plan de trabajo.**

### **8.1 Tiempo.**

El tiempo de desarrollo del proyecto se estima en 9 meses. La mecánica de trabajo consistirá en desarrollar el sistema por etapas, probando y refinando el funcionamiento de las mismas conforme progrese el proyecto. Cada etapa terminada se irá integrando al sistema, hasta llevarlo a completitud.

### **8.2 Recursos materiales.**

En cuanto a estos recursos, la empresa beneficiada del Canal 12 de televisión ha acordado aportar todo el recurso material necesario. Los recursos a usar incluyen locales de trabajo, herramientas, equipos electrónicos, equipos de cómputo, herramientas de software, etc. La naturaleza exacta de los recursos a usar será determinada a medida el proyecto avance.

## IX. Cronograma de actividades.

Actividad	Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7				Mes 8				Mes 9			
	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4
Investigación y elección del equipo electrónico	■																																			
Investigación y elección de las herramientas de software	■																																			
Desarrollo del prototipo de hardware de conmutación					■																															
Desarrollo del prototipo de software de conmutación					■																															
Creación del prototipo de reproducción de video remoto									■																											
Creación del software de control de reproducción de video									■																											
Creación del sistema remoto (con control local solamente)													■																							
Primera defensa del proyecto																																				
Elaboración del programa remoto de control																	■																			
Desarrollo del programa principal de control (master)																	■																			
Creación del sistema de transferencia de archivos																					■															
Creación del sistema de prueba																									■											
Creación del plan de implementación																													■							
Segunda defensa del proyecto																																	■			

## **X. Bibliografía y Fuentes de Consulta.**

### Manuales de usuario y hojas técnicas:

- [1] Adrienne Electronics Corporation, Model AEC-1 10x1 Routing Switcher Instruction Manual, Fourth Edition, Enero 1997.
  
- [2] Kramer Electronics Ltd., Vertical Interval Switchers User Manual - Models: VS-1211, VS-1011, VS-811, VS-611, VS-411 With Balanced Audio, Rev 2.
  
- [3] Microchip Technology Inc., PIC16F627A/628A/648A Data Sheet (40044F), Revision F, Marzo 2007.
  
- [4] Maxim Integrated Products, MAX220–MAX249 - +5V-Powered, Multichannel RS-232 Drivers/Receivers Datasheet, Rev 15, Enero 2006.
  
- [5] Fairchild Semiconductor, 2N3904 Datasheet, Rev A.
  
- [6] STMicroelectronics, L78LxxAB, L78LxxAC, L78LxxC Positive Voltage Regulators Datasheet, Rev14, Septiembre 2007.

### Enlaces de Internet:

Adrienne Electronics Corporation: <http://www.adrielec.com/>

(Última revisión: 17 de Enero de 2008).

Hoja web con los detalles del enrutador de video AEC-1VAA:

<http://www.adrielec.com/aec-1.htm> (Última revisión: 17 de Enero de 2008).

Kramer Electronics: <http://www.kramerelectronics.com/>

(Última revisión: 17 de Enero de 2008).

Hoja web con los detalles del enrutador de video VS-611:

<http://www.kramerelectronics.com/indexes/item.asp?desc=304>

(Última revisión: 17 de Enero de 2008).

Microchip Technology Inc.: <http://www.microchip.com>

(Última revisión: 17 de Enero de 2008).

Hoja web con los detalles del microcontrolador PIC16F628A:

[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1335&dDocName=en010210](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010210) (Última revisión: 17 de Enero de 2008).

Maxim Integrated Products: <http://www.maxim-ic.com/>

(Última revisión: 17 de Enero de 2008).

Hoja web con los detalles del MAX232:

[http://www.maxim-ic.com/quick\\_view2.cfm/qv\\_pk/1798/t/al](http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798/t/al)

(Última revisión: 17 de Enero de 2008).

Fairchild Semiconductor (Fabricante del 2N3904):

<http://www.fairchildsemi.com/> (Última revisión: 17 de Enero de 2008).

STMicroelectronics: <http://www.st.com/> (Última revisión: 17 de Enero de 2008).

Hoja web con los detalles del 78L05ACZ:

<http://www.st.com/stonline/products/literature/ds/2145/178l05ac.htm>

(Última revisión: 17 de Enero de 2008).

This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.