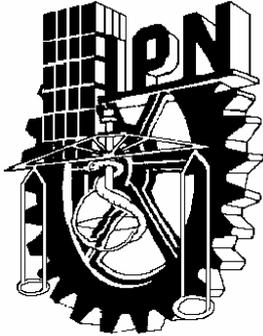


**INSTITUTO POLITECNICO NACIONAL**  
**CENTRO DE INVESTIGACION EN CIENCIA**  
**APLICADA Y TECNOLOGIA AVANZADA**



# **CICATA UNIDAD LEGARIA**

**“Enlace de comunicación para el control de dispositivos mecatrónicos a bordo de una unidad móvil.”**

*TESIS*

Que para obtener el grado de:

***"Maestro en Tecnología Avanzada"***

PRESENTA:

**Ing. Daniel Sánchez Guzmán**

**Director:**

**Dr. José Antonio Calderón Arenas.**

**Co-Director:**

**Dr. Pedro Guevara López.**

**AGOSTO 2006.**



# INSTITUTO POLITECNICO NACIONAL SECRETARIA DE INVESTIGACION Y POSGRADO

## ACTA DE REVISION DE TESIS

En la Ciudad de México, D. F. siendo las 11:00 horas del día 31 del mes de mayo del 2006 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CICATA-IPN para examinar la tesis de grado titulada: "Enlace de comunicación para el control de dispositivos mecatrónicos a bordo de una unidad móvil".

Presentada por el alumno:

**SANCHEZ**

Apellido paterno

**GUZMAN**

materno

**DANIEL**

nombre(s)

Con registro: 

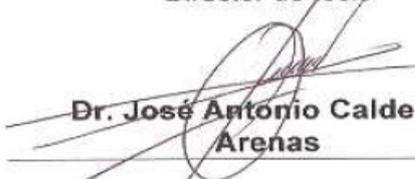
B	0	3	0	9	0	2
---	---	---	---	---	---	---

aspirante al grado de: **MAESTRIA EN TECNOLOGIA AVANZADA**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISION REVISORA

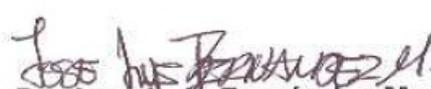
Director de tesis

  
**Dr. José Antonio Calderón Arenas**

Codirector de tesis

  
**Dr. Pedro Guevara López**

  
**Dr. José de Jesús Medel Juárez**

  
**Dr. José Luis Fernández Muñoz**

  
**Dr. Teodoro Rivera Montalvo**

EL PRESIDENTE DEL COLEGIO

  
**Dr. José Antonio Irán Díaz Góngora**

  
CENTRO DE INVESTIGACION  
EN CIENCIA APLICADA Y  
TECNOLOGIA AVANZADA



**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA CESION DE DERECHOS**

En la Ciudad de México Distrito Federal el día 08 del mes de Junio del año 2006, el (la) que suscribe C. Daniel Sánchez Guzmán alumno (a) del Programa de Postgrado en Tecnología Avanzada con número\* de registro B030902, adscrito a CICATA – LEGARIA (IPN), manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. José Antonio Calderón Arenas y cede los derechos del trabajo intitulado Enlace de comunicación para el control de dispositivos mecatrónicos a bordo de una unidad móvil., al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección dsanchez@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

  
Daniel Sánchez Guzmán.

Nombre y firma

## CONTENIDO

Agradecimientos.	i
Dedicatoria.	ii
Resumen.	iii
Abstract.	iv
<b>CAPITULO 1. INTRODUCCIÓN.</b>	
1.1.- Antecedentes.	1
1.2.- Objetivo.	1
1.3.- Justificación.	2
1.4.- Infraestructura.	3
1.4.1.- Recursos Humanos.	3
1.4.2.- Equipo e Instalaciones.	3
1.5.- Aplicación directa del trabajo.	4
1.6.- Metodología para el desarrollo del proyecto.	4
1.7.- Estructura de la presente tesis.	4
<b>CAPITULO 2. SISTEMA OPERATIVO.</b>	
2.1.- Características de Comunicación con un Sistema Operativo.	9
2.2.- Funciones Básicas de un Sistema Operativo.	11
2.3.- Estructura de un Sistema Operativo	11
2.4.- Categoría de los Sistemas Operativos.	13
<b>CAPITULO 3. AUDIO Y VÍDEO DIGITAL.</b>	
3.1.- Análisis de la Señal de Audio.	18
3.2.- Análisis de la Señal de Vídeo.	20
3.3.- Audio y Vídeo junto.	21
3.4.- Digitalización.	23
3.5.- Distribuir el vídeo.	24
<b>CAPITULO 4. COMUNICACIONES DE DATOS.</b>	
4.1.- Modelo de referencia OSI/ISO.	27
4.2.- Transmisión de Datos.	29
4.3.- Transmisión de Datos Analógicos y Digitales.	30
4.4.- Perturbaciones en la Transmisión.	31
<b>CAPITULO 5. PLANTEAMIENTO DEL PROBLEMA.</b>	
5.1.- Planteamiento del problema: Enlace de Bajada.	33
5.1.1.- Transmisión de datos.	33
5.2.- Planteamiento del problema: Enlace de Subida.	34
5.2.1.- Codificación de la Voz en Señal Digital y Tratamiento Digital.	35
5.3.- Desarrollo de la Propuesta.	37
5.3.- Desarrollo de la Propuesta para el Enlace de Bajada.	37
5.3.2.- Desarrollo de la propuesta del Enlace de Subida.	38
5.4.- Despliegue de Vídeo y Sonido en la Estación Remota.	38

CAPITULO 6. IMPLEMENTACIÓN DEL SISTEMA.	
6.1.- Implementación del Sistema Operativo.	40
6.2.- Instalar las Tarjetas de Red Inalámbricas.	41
6.3.- Diseño e implementación de Software para la transmisión de Audio y Vídeo.	42
CAPITULO 7. CONCLUSIONES.	49
ANEXO A.	A-1
ANEXO B.	B-1
BIBLIOGRAFÍA	

## INDICE DE FIGURAS.

Figura 1.1. Diagrama general de interconexión.	2
Figura 1.2. Diagrama a bloques del enlace de bajada.	3
Figura 1.3. Diagrama a bloques correspondiente al enlace de subida.	3
Figura 2.1. El Sistema Operativo como Interfaz de Usuario.	8
Figura 2.2. Interfaz en modo texto.	9
Figura 2.3. Interfaz en modo grafico (Graphical User Interface GUI)	10
Figura 2.4. Estructura de Microkernel.	12
Figura 2.5. El modelo Cliente-servidor.	12
Figura 2.6. El modelo Cliente-servidor en un sistema distribuido.	12
Figura 2.7. Sistema Operativo Distribuido.	15
Figura 3.1. Distribución mediante servidor http.	25
Figura 3.2. Distribución mediante servidor Streaming.	25
Figura 4.1. Modelo de referencia OSI/ISO.	27
Figura 5.1. Diagrama general de los procesos a realizar en la Estación Remota.	39
Figura 6.1. Arquitecturas de red inalámbrica.	41
Figura 6.2. Modelo para el procesamiento de medios.	43
Figura 6.3. Arquitectura de RTP.	44
Figura 6.4. Paquete RTP.	45

# AGRADECIMIENTOS.

---

Quiero agradecer primeramente a mi Madre por haberme dado la dicha de vivir y ser mi guía hacia el crecimiento y desarrollo, es gracias a su fe y esperanza depositada en mí que he logrado esta meta. También quiero agradecer a mis hermanos porque son las personas que me han hecho ver que los lazos de hermandad nos ayudan como seres humanos a progresar. De igual manera agradezco a mi familia materna por ser siempre el motor impulsor para mi formación tanto personal como profesional, es gracias a los valores y principios que me han inculcado que he logrado una meta más en la larga escalera de la vida. A mis amigos les agradezco su preocupación y su apoyo para poder seguir y aunque a veces se nos presentan obstáculos a lo largo de la vida es gracias al apoyo de todos ustedes que he logrado este objetivo tan importante para mí.

Mi infinito agradecimiento al Instituto Politécnico Nacional y al pueblo de México por su apoyo en la conclusión de mis estudios de postgrado, es gracias a este apoyo en la forma en que el país progresa y da un importante crecimiento, prometo no defraudarlos para que este apoyo que se dio en mí, sea transmitido a las futuras generaciones de mexicanos, para que nuestro país crezca y las brechas entre la desigualdad y la pobreza sean erradicadas lo más pronto posible. Así mismo agradezco a mi director de tesis el Dr. José Antonio Calderón Arenas por apoyarme desde un principio y en todo momento para lograr este proyecto, así como a todos mis profesores, compañeros y amigos que conocí dentro de la escuela, ya que es gracias a su compañerismo, experiencia y visión que haya comprendido el mundo en el que vivo y en donde estoy muy feliz de pertenecer a este lugar, ya que me siento profundamente responsable y comprometido con el desarrollo de nuestra nación.

Han sido muchas las personas que han participado conmigo en este proyecto de vida, posiblemente podría hacer un libro con los nombres y agradecimientos particulares, espero me comprendan si no los puedo mencionar a todos y quiero decirles a todos en general: ¡Gracias por su amor, apoyo, comprensión, esperanza, fe y amistad!

## **DEDICATORIA.**

---

A Laura, mi madre por ser el pilar de mi formación ya que gracias a su amor, cariño y comprensión me inculco los valores y principios necesarios para la vida y sin los cuales no sería yo una persona de bien para mi familia, la nación y el mundo entero.

A Fernando y Ricardo mis hermanos, por su apoyo, respaldo y amistad brindada, para lograr mis objetivos.

A Bernardo y Carmela mis abuelos maternos, por ser la fuente de sabiduría, brindándome su amor y cuidado.

A Jorge, Enrique, Arturo, Bernardo y Miguel Angel, mis tíos maternos, por que con ellos comprendí lo que es una familia en su totalidad.

A Magdalena, por ser la mujer de la cual estoy profundamente enamorado y con quien he compartido momentos inolvidables.

A mis amigos por su comprensión y esperanza.

No pueden faltar los agradecimientos a México, que es él motivo de inspiración, ya que este proyecto esta enfocado a resolver problemas del país.

## **RESUMEN.**

---

Este sistema esta diseñado para controlar robot a distancia. La principal característica de este control es que el robot realizará los mismos movimientos que realiza un brazo humano, utilizando un “guante” con sensores, para lograr esta meta hemos definido dos enlaces de comunicación de red. El primer enlace, llamado enlace de bajada, el cual se encarga de enviar las señales digitalizadas del “guante” al robot. El segundo enlace llamado el enlace de subida, encargado de enviar las señales de audio y video generadas en el robot. Este trabajo describe el análisis y diseño de estos dos enlaces de comunicaciones utilizados en todo el sistema.

## **ABSTRACT.**

---

This system was designed to control a robot at distance. The main characteristic of this control, is that the robot will do the same movements that the human arm does, using a “glove” with sensors; to achieve this goal we have defined two data network communication links. The first link, called the downward link, which sends the signals captured from the “glove” to the robot. The second one called the upward link that receives audio and video from the robot. This work describes the analysis and design of the communication links that serve to all the system.

# INTRODUCCIÓN.

---

*“La alegría de mirar y aprender es el don más bello de la naturaleza”.*

## 1.1.- Antecedentes.

La Tele-robótica es un campo en la actualidad de acelerado crecimiento, esto es dado por el desarrollo tecnológico correspondiente a diversos factores como procesadores más rápidos, nuevas técnicas de algoritmos y nuevas expectativas en el área de las tecnologías de la información. Tenemos en la actualidad a nivel mundial muchas aplicaciones que utilizan sistemas tele-robóticos entre las que encontramos exploraciones espaciales, aplicaciones biomédicas, y la realización de tareas peligrosas, entre otras muchas soluciones. El líder a nivel mundial es la NASA, la cual en la actualidad esta desarrollando aplicaciones comerciales que no sólo sean referentes a la exploración espacial.

Para comprender el campo sobre el cual se desarrolla la tele-robótica debemos de mencionar los tres grupos que abarca la robótica: robots industriales, teleoperadores, y telerobots. Los fundamentos sobre los que se desarrollan estos tres grupos son los mismos, en donde los robots industriales son los más conocidos y están diseñados para aplicaciones industriales de manufactura; los robots teleoperados tienen la ventaja sobre los robots industriales en que estos últimos proveen la destreza del ser humano, permitiéndonos por lo tanto transmitir esas habilidades de manera remota, pero tienen el inconveniente de tener limitantes en cuanto a su funcionamiento en tiempo real, ya que existen en la actualidad muchos factores que limitan la transmisión de información. Por último tenemos a los telerobots, estos integran tanto la programación dentro del robot como a la teleoperación, mejorando en mucho, características que en cualquiera de los dos grupos anteriores serían serias limitantes, obteniendo por lo tanto una sinergia que permite el desarrollo de nuevas aplicaciones y soluciones.

## 1.2.- Objetivo.

Diseñar un sistema Telemático que permita realizar la comunicación entre una estación base, como lo es una microcomputadora, y una estación móvil, como es un dispositivo Mecatrónico, pasando a través de una red de comunicación de datos como puede ser Internet vía una estación base (pudiendo ser esta un equipo de comunicaciones u otra

microcomputadora), con el propósito de enviar secuencias de control; así como la recepción de señales de manera remota de audio y vídeo utilizando tecnología de red existente, como una webcam y el uso de tecnologías de red inalámbrica. El desarrollo de este sistema esta compuesto desde el diseño de la interfaz gráfica, creación de las conexiones de red de manera lógica e instalación y puesta en marcha de los dispositivos tanto alámbricos como inalámbricos tanto para las señales enviadas como recibidas.

### 1.3.- Justificación.

Es indispensable comenzar a realizar sistemas Telemáticos, que no solo se enfoquen en la informática o en las telecomunicaciones, además este proyecto se propone como base para perfección de futuras generaciones que se interesen en el desarrollo de este tipo de sistemas. Los alcances de este proyecto están planteando resolver tareas realmente peligrosas, pero con pocos recursos económicos.

También creo que un proyecto como este, es muy difícil de realizar por la iniciativa privada, debido al costo en personal (Ingenieros Informáticos, Ing. Comunicaciones, Ing. Mecánicos, Ing. Electrónica). Además que la necesidad de este producto no es de muchas organizaciones, sino de pocas organizaciones pero que pueden servir a todo el país.

La participación de la Telemática en la actualidad esta teniendo un papel preponderante dentro de diversas ramas, como lo es la industria, la medicina, la milicia, etc. Donde la presencia de la información es de suma importancia, recordemos que un sistema Telemático no se encarga sólo de la comunicación de datos, para ese propósito recurrimos a los Ingenieros en Telecomunicaciones, ni tampoco se encarga de diseñar sólo software, para eso recurrimos a los Ingenieros en Informática. La Ingeniería en Telemática se encarga de integrar todos los conocimientos en cuanto a transmisión de datos y diseño de sistemas informáticos para un mejor aprovechamiento de estos recursos.

El presente proyecto integrará todos los conocimientos que un especialista Telemática con Maestría en Tecnología Avanzada debe cubrir para poder hacer frente a las necesidades del país en un mundo en el que la globalización se esta haciendo presente y en donde se requieren ahora más que nunca gente con la capacidad de adaptación a un campo tan cambiante y de rápido crecimiento como lo es la tecnología en telecomunicaciones e informática. La siguiente figura muestra el diagrama general del sistema, así como el diagrama particular propuesto para los correspondientes enlaces de bajada y de subida.

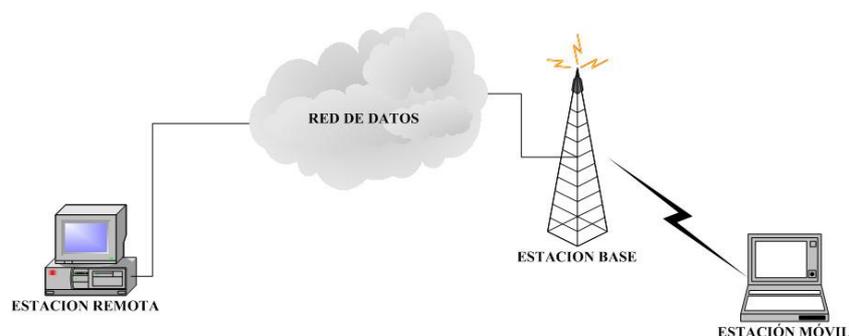


Figura 1.1.- Diagrama general de interconexión.

La siguiente figura nos muestra el diagrama a bloques correspondiente al enlace de bajada.

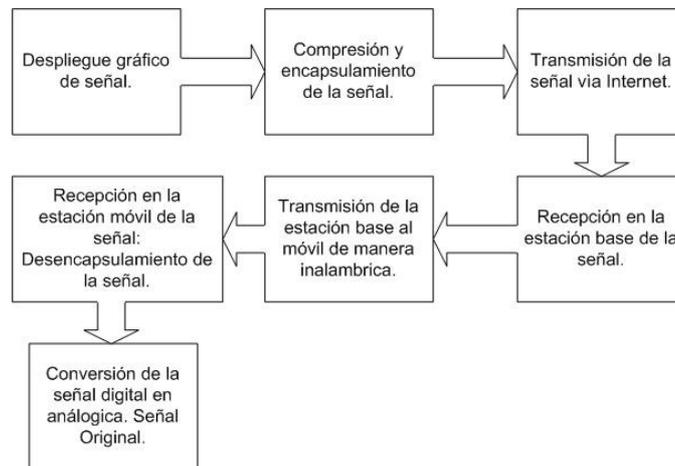


Figura 1.2.- Diagrama a bloques del enlace de bajada.

La siguiente figura nos muestra el diagrama a bloques correspondiente al enlace de subida.

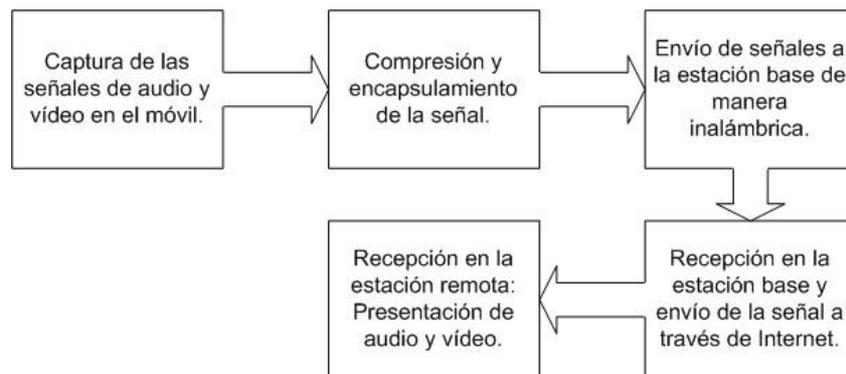


Figura 1.3.- Diagrama a bloques correspondiente al enlace de subida.

## 1.4.- Infraestructura.

### 1.4.1.- Recursos Humanos.

Alumno de Ingeniería en Telemática y un profesor director.

### 1.4.2.- Equipo e Instalaciones.

Dos computadoras con acceso a Internet; una PC portátil y una de escritorio, una conexión a la red, equipo de red como lo es un switch, tarjetas de red inalámbricas y puntos de acceso.

### 1.5.- Aplicación directa del trabajo.

- En la enseñanza de un sistema Telemático
- Realización de investigación en la mejora de bloques del sistema.

### 1.6.- Metodología para el desarrollo del proyecto.

Dentro del marco de referencia de la Ingeniería Telemática con Maestría en Tecnología Avanzada, pretendo realizar varias actividades de manera multidisciplinaria que me permitan hacer tareas integradoras para el correcto desarrollo del proyecto.

Los pasos a desarrollar dentro de una actividad determinada serán:

- **Investigación.** Esta parte corresponde a todo el proceso de investigación a cerca de los tipos de tecnologías, como pueden ser: características, aplicaciones, opciones y limitaciones que permite tanto el equipo como la programación. Esta es una de las partes más importantes puesto que determinarán el rumbo que ha de llevar el desarrollo del proyecto.
- **Pruebas.** Dentro de este aspecto se consideran las pruebas correspondientes del equipo o programas definidos en el punto anterior, para corroborar los parámetros correspondientes y necesarios para nuestro proyecto. De estas pruebas se observarán los resultados obtenidos para después pasar a la etapa de diseño, recordemos que aquí se observan las características básicas de la tecnología escogida, para después continuar con la integración de todos los módulos.
- **Diseño.** En este punto realizaré tanto el diseño como la puesta en marcha de los módulos desarrollados con anterioridad, aquí es donde se realiza la parte integradora de todos los componentes con los que tiene determinada etapa del proyecto. Esta etapa es de suma importancia puesto que es aquí donde se ponen a prueba las capacidades de investigación de la persona que desarrolla el proyecto puesto que se pueden presentar circunstancias que en la etapa de pruebas no se presentaron, puesto que esta etapa es sobre las características que otorga el fabricante bajo ciertas condiciones para probar funcionalidad básica, pero en la etapa de diseño entra en juego la capacidad del diseñador, puesto que es aquí donde todas las variables entran en juego.

### 1.7.- Estructura de la presente tesis.

El presente proyecto se divide en ocho capítulos, los cuales tienen el siguiente contenido. El Capítulo I, Introducción contiene los antecedentes y justificación del presente proyecto. El Capítulo II, Sistema Operativo, aporta los elementos teóricos necesarios para utilizar los sistemas operativos dentro del sistema. El Capítulo III, Audio y vídeo digital, aquí veremos los aspectos a considerar para la transmisión de señales analógicas para la visualización de manera remota del robot, como lo es el audio y vídeo, los cuales desplegaremos en la estación remota. El Capítulo IV, Comunicaciones de datos, describe los elementos para comprender como vamos a enviar la información a través de la red de comunicación de datos. El Capítulo V, Planteamiento del problema, trata los aspectos fundamentales sobre los que me baso para conjuntar todos los aspectos y variables a considerar en el sistema,

permitiéndome definir de la manera más amplia posible todo el proyecto en su conjunto. El Capítulo VI, Desarrollo de la propuesta, en el se describen los procesos a llevar a cabo para dar la solución práctica y física para cada uno de los enlaces de comunicaciones. El Capítulo VII, Validación del Sistema, describe los elementos construidos o implementados para integrar el prototipo final tales como interfaces de comunicación, códigos de programación, etc. Es aquí donde participamos de manera conjunta para unir los proyectos de telemática y mecatrónica. Por último el Capítulo VIII, Conclusiones, en el cual se evalúan los resultados obtenidos respecto a los objetivos planteados previamente y se exponen las razones para elegir las soluciones adoptadas en cada parte del problema.

## SISTEMA OPERATIVO.

---

*“Si buscas resultados distintos, no hagas siempre lo mismo”.*

El Sistema Operativo es una de las partes más importantes del presente proyecto, puesto que se encarga de todo el control lógico del sistema. Para entrar en detalles, definamos algunas características y conceptos que serán de utilidad para poder describir un Sistema Operativo y así poder comprender de una mejor manera como lo aplicaremos en nuestro proyecto.

Para comenzar, definamos el elemento físico en el que se centra toda la actividad informática, es decir, la computadora.

*Una computadora es una máquina de origen electrónico con una o más unidades de proceso y equipos periféricos controlados por programas almacenados en su memoria, que pueden realizar una gran variedad de trabajos. [1]*

Todo este componente físico es denominado comúnmente Hardware. Ahora bien, al ser una máquina programable tiene que contar con información que le indique como utilizar todas sus unidades físicas o herramientas que la componen, conocidas como recursos, para llevar a cabo el trabajo. Esta información es lo que denominamos soporte lógico o Software.

Una computadora sin Software sería una máquina inútil, mientras que con él puede almacenar, procesar y obtener información, editar textos, controlar el entorno, etc.

Un Sistema Operativo es un elemento constitutivo de software que en sus principios fue definido de la siguiente manera:

*Un Sistema Operativo es el soporte lógico que controla el funcionamiento del equipo físico. [1]*

En la actualidad no resulta sencilla su definición, de forma que pueden darse dos diferentes definiciones:

### Punto de vista del **USUARIO**.

Un Sistema Operativo es un conjunto de programas y funciones que ocultan los detalles del hardware, ofreciendo al usuario una vía más sencilla y flexible de acceso a la información.

El ocultar los detalles del hardware a usuarios y parte del personal de información tiene dos objetivos:

- **Abstracción.** La tendencia actual del software en toda su extensión es la de dar una visión global y abstracta de la computadora haciendo fácil su uso ocultando por completo la gestión interna.
- **Seguridad.** Existen instrucciones en la máquina que pueden parar la computadora, interferir procesos, etc. Por ello, es necesario restringir determinadas operaciones a los usuarios creando varios niveles de privilegio, de tal forma que cada usuario tenga protegida su información y sus procesos.

Punto de vista de **GESTOR DE RECURSOS.**

Un Sistema Operativo es el administrador de recursos ofrecidos por el hardware para alcanzar un eficaz rendimiento de los mismos.

Entre los principales recursos que administra se encuentran:

- Procesador.
- Memoria.
- Entrada/Salida.
- Información.

Los sistemas operativos construyen recursos de alto nivel, que denominaremos virtuales, a base de encubrir los realmente existentes de bajo nivel, que denominaremos físicos. Por lo tanto, desde el punto de vista del usuario o de un proceso, la máquina física es convertida por el Sistema Operativo en una máquina virtual, también conocida como máquina extendida, que a diferencia de la física, ofrece muchas más funciones y más cómodas de utilizar.

El Sistema Operativo proporciona, además, servicios de los que no dispone el hardware, como pueden ser, por ejemplo, la utilización de la computadora por varios usuarios simultáneamente, interacción entre usuario y programa en ejecución al mismo tiempo, etc.

Para concluir esta introducción, vamos a dar una definición más académica que las anteriores, basándonos en la definición previa de los términos SISTEMA y OPERATIVO.

- ◆ **Sistema.** Conjunto de personas, máquinas y objetos que, ordenadamente relacionadas entre sí, contribuyen a lograr un determinado objetivo.
- ◆ **Operativo.** Personas, máquinas y objetos que trabajan conjuntamente y consiguen el objetivo deseado.

De las definiciones anteriores podemos obtener la de Sistema Operativo de la siguiente forma:

*SISTEMA OPERATIVO. Conjunto de programas que ordenadamente relacionados entre sí, contribuyen a que la computadora y el usuario lleven a cabo correctamente su trabajo.*

Podemos resumir que los sistemas operativos cubren dos objetivos fundamentales:

- 1.- Facilitar el trabajo al usuario.
- 2.- Gestionar de forma eficiente los recursos.

Ahora bien, no todo el software que se ejecuta sobre el hardware se considera como Sistema Operativo. Existen dos tipos de programas que conviene distinguir:

- Programas de Sistema. Son los que manejan el hardware, controlan los procesos, hacen más cómodo el entorno de trabajo, y son propios del sistema operativo.
- Programas de Aplicación. Son los que resuelven un problema concreto de los usuarios y los que no son suministrados en el Sistema Operativo. Son programas diseñados y codificados por analistas y programadores de aplicaciones conjuntamente con los usuarios.

Podemos ver que el Sistema Operativo tiene una característica fundamental, la de servir como una interfaz, esto es, permite la gestión y comunicación entre todos los componentes de un sistema computarizado, incluyendo al usuario como un componente más de todo este sistema. El correcto aprovechamiento de todos estos recursos permite que se obtengan resultados lo más óptimos posibles. La siguiente figura ejemplifica este aspecto.



Figura 2.1. El Sistema Operativo como Interfaz de Usuario

La tecnología continúa en un vertiginoso desarrollo, del cual no hay certidumbre para definir hasta donde llegará, pero de algo si estamos seguros y es que cualquier equipo de hardware, para ser controlado, requiere de niveles superiores, los cuales son lógicos y necesitan de la programación para poder entrar en operación.

Como su nombre lo indica en este capítulo se decidirá que tipo de Sistema Operativo será utilizado para el desarrollo del Sistema Telemático propuesto. En los primeros párrafos se mencionó la importancia y se definió lo que es el Sistema Operativo, de ahí que el presente análisis sea un poco extenso, ya que no es fácil determinar un tipo de Sistema Operativo sin antes ver sus características y su comparación con otros sistemas.

Dentro del problema de la definición del Sistema Operativo tenemos que evaluar y analizar diferentes puntos que nos permitan tener el máximo aprovechamiento del sistema, recordemos que el ambiente donde operará este es bastante complejo y extenso, tiene muchas facetas y puntos que sólo analizándolo desde el punto de vista Telemático es posible comprender tanto la parte del comportamiento del sistema como el entorno sobre el cual operará, de ahí el que no sólo se proponga de manera arbitraria cualquier Sistema Operativo sino que sea necesario un estudio lo más exacto posible y que abarque diferentes aspectos que influyan en la determinación del Sistema Operativo.

El presente capítulo describe los aspectos que creemos son indispensables para el correcto funcionamiento del sistema, comenzando con el análisis de sus características y puntos que consideré para poder determinar el Sistema Operativo.

## 2.1.- Características de Comunicación con un Sistema Operativo.

Para poderse comunicar con la computadora, el usuario necesita de una interfaz, en donde la interacción hombre-máquina se da de manera directa, en un principio la comunicación era de manera mecánica, puesto que los primeros usuarios eran personas expertas con conocimiento técnico de la operación y puesta en marcha del sistema operativo, el problema principal de la comunicación era que sólo una persona o un pequeño número de ellas sabía el funcionamiento básico del sistema, si en determinado momento se presentaba un problema, era necesario recurrir a manuales complejos por su naturaleza y en donde las correcciones al sistema eran demasiado lentas, llegando a tomar hasta varios días para poder corregir un error por pequeño que este fuera dentro de la computadora.

Conforme el desarrollo computacional se presentó los ambientes de comunicación con la computadora también fueron evolucionando, permitiendo que el personal ya no requiriera ser tan calificado, puesto que el sistema computacional tenía ya cierta autonomía. Para describir más a detalle este tipo de comunicación tenemos lo que se conoce como interfaz de línea de comandos e interfaz gráfica, las cuales veremos a continuación.

### Interfaz de Línea de Comandos.

Es la forma de interfaz entre el sistema operativo y el usuario en la que este último escribe los comandos utilizando un lenguaje de comandos especial. Los sistemas con interfaces de líneas de comandos se consideran más difíciles de aprender y utilizar que los de las interfaces gráficas. Sin embargo, los sistemas basados en comandos son por lo general programables, lo que les otorga una flexibilidad que no tienen los sistemas basados en gráficos carentes de una interfaz de programación. A continuación presentamos un ejemplo de este tipo de interfaz:

```
$ ls -al ↓
total 8
drwxr-xr-x 2 rocky other 96/11
drwxrwxr-x 7 root  sys  96/01
-rw-r--r-- 1 rocky other 95/12
$ _
```

Figura 2.2. Interfaz en modo texto.

### Interfaz Gráfica de Usuario.

Es el tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el Mouse.

Para los autores de aplicaciones, las interfaces gráficas de usuario ofrecen un entorno que se encarga de la comunicación con el ordenador o computadora. Esto hace que el

programador pueda concentrarse en la funcionalidad, ya que no esta sujeto a los detalles de la visualización ni a la entrada a través del Mouse o el teclado. También permite a los programadores crear programas que realicen de la misma forma las tareas más frecuentes, como guardar un archivo, ya que la interfaz proporciona mecanismos estándar de control como ventanas y cuadros de diálogo. Otra ventaja es que las aplicaciones escritas para una interfaz gráfica de usuario son independientes de los dispositivos: a medida que la interfaz cambia para permitir el uso de nuevos dispositivos de entrada y salida, como un monitor de pantalla grande o un dispositivo óptico de almacenamiento, las aplicaciones pueden utilizarlos sin necesidad de cambios. La siguiente figura muestra un ejemplo de este tipo de interfaz.

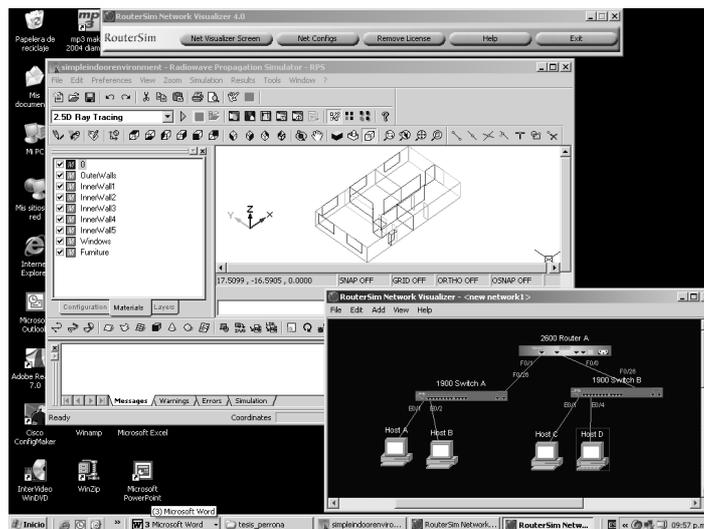


Figura 2.3. Interfaz en modo gráfico (Graphical User Interface GUI).

Los métodos de comunicación mencionados anteriormente nos permiten comenzar a definir los requerimientos del sistema de nuestro proyecto. Para nosotros es preponderante la comunicación entre el usuario y el sistema, pero de igual importancia se presenta la programación y adecuación de los diferentes programas, para trabajar de la mejor manera posible.

Ahora vamos a definir uno de los requerimientos del sistema operativo, en función de la comunicación entre este y el usuario:

***“El Sistema Operativo debe de tener la capacidad de brindar un ambiente gráfico, de fácil acceso e interacción con el usuario. A demás debe permitir la rápida adaptación y modificación de sus funciones mediante la programación del propio Sistema Operativo que puede ser en modo texto”.***

Una vez definido lo anterior, continuemos observando el entorno de los sistemas operativos, evaluando las funciones básicas que debe cumplir un Sistema Operativo.

## **2.2.- Funciones Básicas de un Sistema Operativo.**

Las funciones que debe realizar un sistema operativo nos permiten definir que es lo que en esencia ha de procesar y es una manera de poder distinguir entre los programas que son propios del sistema y los que son de aplicaciones, entre las diferentes funciones se encuentran:

- Interpreta los comandos que permiten al usuario comunicarse con el ordenador.
- Coordina y manipula el hardware de la computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el Mouse.
- Organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas.
- Gestiona los errores de hardware y la pérdida de datos.
- Servir de base para la creación del software logrando que equipos de marcas distintas funcionen de manera análoga, salvando las diferencias existentes entre ambos.
- Configura el entorno para el uso del software y los periféricos; dependiendo del tipo de máquina que se emplea, debe establecerse en forma lógica la disposición y características del equipo. Como por ejemplo, una microcomputadora tiene físicamente dos unidades de disco, puede simular el uso de otras unidades de disco, que pueden ser virtuales utilizando parte de la memoria principal para tal fin. En caso de estar conectado a una red, el sistema operativo se convierte en la plataforma de trabajo de los usuarios y es este quien controla los elementos o recursos que comparten. De igual forma, provee de protección a la información que almacena.
- Protección: Evitar que las acciones de un usuario afecten el trabajo que esta realizando otro usuario.

Aquí podemos definir otro de los requerimientos del sistema operativo que es:

*“El sistema operativo de nuestra elección debe de realizar las funciones básicas antes descritas, como mínimo, para poder tener un mejor control y administración de todos los recursos tanto de software como de hardware”.*

## **2.3.- Estructura de un Sistema Operativo.**

Dentro de la estructura del Sistema Operativo podemos identificar como es el tipo de arquitectura lógica del sistema, esto es, como opera el sistema de manera básica, describiendo la forma en que los procesos son ejecutados, las prioridades de estos, la forma en que se comunican y el tipo de ejecución que se debe de llevar a cabo.

En este apartado examinaremos cuatro estructuras distintas que ya han sido probadas, con el fin de tener una idea más extensa de cómo esta estructurado el sistema operativo. Veremos brevemente algunas estructuras de diseños de sistemas operativos.

### Estructura por microkernel.

Las funciones centrales de un Sistema Operativo son controladas por el núcleo (kernel) mientras que la interfaz del usuario es controlada por el entorno (shell). Por ejemplo, la parte más importante del DOS (Disk Operating System) es un programa con el nombre "COMMAND.COM". Este programa tiene dos partes. El kernel, que se mantiene en memoria en todo momento, contiene el código máquina de bajo nivel para manejar la administración de hardware para otros programas que necesitan estos servicios, y para la segunda parte del COMMAND.COM el shell, el cual es el interprete de comandos.

Las funciones de bajo nivel del Sistema Operativo y las funciones de interpretación de comandos están separadas, de tal forma que puede mantener el kernel DOS corriendo, pero utilizar una interfaz de usuario diferente. Esto es exactamente lo que sucede cuando cargamos un sistema como Microsoft Windows, el cual toma el lugar del shell, reemplazando la interfaz de línea de comandos con una interfaz gráfica del usuario. Existen muchos "shells" diferentes en el mercado, ejemplo: NDOS (Norton DOS), XTG, PCTOOLS, o inclusive el mismo Sistema Operativo MS-DOS a partir de la versión 5.0 incluyó un Shell llamado DOS SHELL. A continuación mostramos este tipo de estructura:

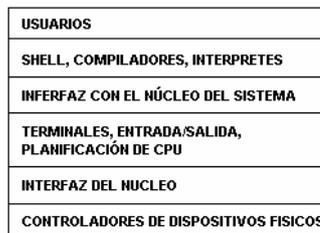


Figura 2.4. Estructura de Microkernel.

### Estructura cliente – servidor.

Una tendencia de los sistemas operativos modernos es la de explotar la idea de mover el código a capas superiores y eliminar la mayor parte posible del sistema operativo para mantener un núcleo mínimo. El punto de vista usual es el de implantar la mayoría de las funciones del sistema operativo en los procesos del usuario. Para solicitar un servicio, como la lectura de un bloque de cierto archivo, un proceso del usuario (denominado proceso cliente) envía la solicitud a un proceso servidor, que realiza entonces el trabajo y regresa la respuesta. En este modelo, que se muestra en la figura 17, lo único que hace el núcleo es controlar la comunicación entre los clientes y los servidores. Al separar el sistema operativo en partes, cada una de ellas controla una faceta del sistema, como el servidor de archivos, servidor de procesos, servidor de terminales o servidor de memoria, cada parte es pequeña y controlable. Además como todos los servidores se ejecutan como procesos en modo usuario y no en modo núcleo, no tienen acceso directo al hardware. En consecuencia si hay un error en el servidor de archivos, éste puede fallar, pero esto no afectará en general a toda la máquina.

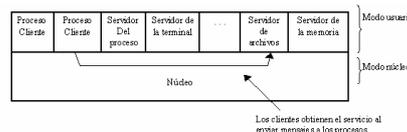


Figura 2.5. El modelo Cliente-servidor.

Otra de las ventajas del modelo Cliente-servidor es su capacidad de adaptación para su uso en los sistemas distribuidos (ver figura 2.5).

Si un cliente se comunica con un servidor mediante mensajes, el cliente no necesita saber si el mensaje se maneja en forma local, en su máquina, o si se envía por medio de una red a un servidor en una máquina remota. En lo que respecta al cliente, lo mismo ocurre en ambos casos: se envió una solicitud y se recibió una respuesta.

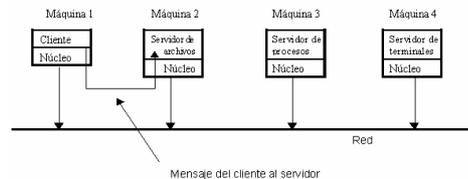


Figura 2.6. El modelo Cliente-servidor en un sistema distribuido.

La estructura de un Sistema Operativo nos permite identificar como es su comportamiento y como interactúa con los diferentes procesos, tanto de control de software como de hardware, para nosotros es importante que el sistema permita tener un control claro y de manera eficiente, he aquí otro requerimiento del sistema:

***“El Sistema Operativo debe de tener una estructura suficientemente robusta de tal manera que permita tener la interacción óptima de todos los procesos del sistema”.***

Al referirnos a robusta estamos haciendo énfasis en que el sistema es de operación crítica y debe responder de la mejor manera a todos los problemas que se presente, teniendo la capacidad en ciertas circunstancias de tomar las decisiones pertinentes.

## 2.4.- Categoría de los Sistemas Operativos.

Para la investigación de este proyecto fue necesario recopilar información de las diferentes formas en que los sistemas operativos son estructurados (como se describió anteriormente) y los tipos de categoría más comunes dentro de los sistemas operativos, encontrando características particulares y diferencias que nos permitan definir de una mejor forma el tipo de sistema operativo, obteniendo en este apartado otro criterio para la selección del sistema operativo, a continuación presentamos las diferentes categorías y al final presentamos que requerimientos adicionales debe de cumplir el sistema para la aplicación de nuestro proyecto en cuanto a categoría se refiere.

### Sistema Operativo Monousuario.

Los sistemas monousuarios son aquellos que nada más pueden atender a un sólo usuario, debido a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se este ejecutando.

Estos tipos de sistemas son muy simples, porque todos los dispositivos de entrada, salida y control dependen de la tarea que se esta utilizando, esto quiere decir, que las instrucciones que se dan, son procesadas de inmediato; ya que existe un solo usuario. Y están orientados principalmente por los microcomputadores.

### **Sistema Operativo Multiusuario.**

Es todo lo contrario a monousuario; y en esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

### **Tiempo Real.**

Un sistema operativo en tiempo real procesa las instrucciones recibidas al instante, y una vez que han sido procesadas muestra el resultado. Este tipo tiene relación con los sistemas operativos monousuarios, ya que existe un solo operador y no necesita compartir el procesador entre varias solicitudes.

Su característica principal es dar respuestas rápidas; por ejemplo en un caso de peligro se necesitarían respuestas inmediatas para evitar una catástrofe.

### **Sistema Operativo de Red (NOS).**

La principal función de un sistema operativo de red es ofrecer un mecanismo para transferir archivos de una máquina a otra. En este entorno, cada instalación mantiene su propio sistema de archivos local y si un usuario de la instalación A quiere acceder a un archivo en la instalación B, hay que copiar explícitamente el archivo de una estación a otra.

Internet proporciona un mecanismo para estas transferencias, a través del programa protocolo de transferencias de archivos FTP (File Transfer Protocol).

Suponga que un usuario quiere copiar un archivo A1, que reside en la estación B, a un archivo A2 en la estación local A. Primero, el usuario debe invocar el programa FTP, el cual solicita al usuario la información siguiente:

- a) El nombre de la estación a partir de la cual se efectuará la transferencia del archivo (es decir la estación B).
- b) La información de acceso, que verifica que el usuario tiene los privilegios de acceso apropiados en la estación B. Una vez efectuada esta comprobación, el usuario puede copiar el archivo A1 de B a A2 en A, ejecutando "get A1 to A2". En este esquema, la ubicación del archivo no es transparente para el usuario; tiene que saber exactamente donde esta cada archivo. Además los archivos no se comparten realmente, porque un usuario solo puede copiar un archivo de una estación a otra. Por lo tanto pueden existir varias copias del mismo archivo, lo que representa un desperdicio de espacio. Así mismo, si se modifican, estas copias no serán consistentes.

### **Sistemas operativos distribuidos.**

En un sistema operativo distribuido los usuarios pueden acceder a recursos remotos de la misma manera en que lo hacen para los recursos locales. La migración de datos y procesos de una instalación a otra queda bajo el control del sistema operativo distribuido.

Permiten distribuir trabajos, tareas o procesos, entre un conjunto de procesadores. Puede ser que este conjunto de procesadores esté en un equipo o en diferentes, en este caso es transparente para el usuario. Existen dos esquemas básicos de éstos. Un sistema fuertemente acoplado es aquel que comparte la memoria y un reloj global, cuyos tiempos de acceso son similares para todos los procesadores. En un sistema débilmente acoplado los procesadores no comparten ni memoria ni reloj, ya que cada uno cuenta con su memoria local.

Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se descompone otro componente debe de ser capaz de reemplazarlo. Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris 10, Mach, Chorus, Spring, Amoeba, Taos, etc.

Características de los Sistemas Operativos distribuidos:

- Colección de sistemas autónomos capaces de comunicación y cooperación mediante interconexiones hardware y software.
- Proporciona abstracción de máquina virtual a los usuarios.
- Objetivo clave es la transparencia.
- Generalmente proporcionan medios para la compartición global de recursos.

Servicios añadidos: denominación global, sistemas de archivos distribuidos, facilidades para distribución de cálculos (a través de comunicación de procesos internodos, llamadas a procedimientos remotos, etc.).

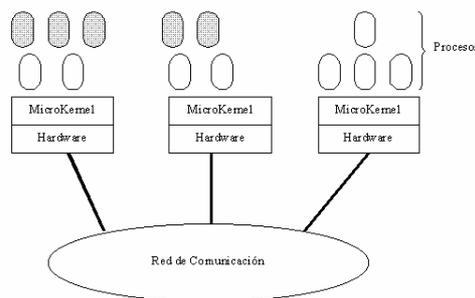


Figura 2.7.- Sistema Operativo Distribuido.

### Sistemas operativos multiprocesadores.

En los sistemas multiprocesador, los procesadores comparten la memoria y el reloj. Se incrementa la capacidad de procesamiento y la confiabilidad, son económicos.

- Multiprocesamiento simétrico: Cada procesador ejecuta una copia del sistema operativo.
- Multiprocesamiento asimétrico: Cada procesador tiene asignado una tarea específica, existe un procesador master que asigna tareas a los procesadores esclavos.

Las computadoras que tienen más de un CPU son llamadas multiproceso. Un sistema operativo multiproceso coordina las operaciones de las computadoras multiprocesadores.

Ya que cada CPU en una computadora de multiproceso puede estar ejecutando una instrucción, el otro procesador queda liberado para procesar otras instrucciones simultáneamente.

Los primeros Sistemas Operativos Multiproceso realizaban lo que se conoce como Multiproceso asimétrico. Una CPU principal retiene el control global de la computadora, así como el de los otros procesadores. Esto fue un primer paso hacia el multiproceso pero no fue la dirección ideal a seguir ya que la CPU principal podía convertirse en un cuello de botella.

En un sistema multiproceso simétrico, no existe una CPU controladora única. La barrera a vencer al implementar el multiproceso simétrico es que los sistemas operativos tienen que ser rediseñados o diseñados desde el principio para trabajar en un ambiente multiproceso. Las extensiones de UNIX, que soportan multiproceso asimétrico ya están disponibles y las extensiones simétricas se están haciendo disponibles. Windows XP de Microsoft soporta multiproceso simétrico.

Hemos descrito las diferentes categorías de los Sistemas Operativos, en donde podemos ver que cada una de ellas tiene sus características y propósitos, es interesante observar que un sistema operativo puede caer en dos o más categorías, por eso a veces es difícil describir la categoría de un determinado Sistema Operativo. Para nuestro proyecto podemos tener un sin número de categorías, pero creemos que las básicas pueden ser las siguientes:

- ***Monousuario. No se puede presentar el caso de que diferentes usuarios trabajen con el sistema, ya que esto repercutiría en el gasto de un número mayor de recursos, de ahí que definamos que el sistema es monousuario.***
- ***Red. La importancia de la red viene del hecho de que muchos de los procesos tienen que ver con comunicación de la familia de protocolos de Internet para poder tener una transmisión de diferentes señales a través de la red y poder comunicar el sistema en cualquier parte del mundo.***
- ***En cuanto a procesamiento podemos ver que es de tipo monoproceso, pero con la característica de multitarea.***

Concluimos este capítulo definiendo el sistema operativo a utilizar, el cual será por propósitos académicos Windows XP. Ya que nos ofrece las características básicas para la implementación de este proyecto.

## AUDIO Y VÍDEO DIGITAL.

---

*“La imaginación es más importante que el conocimiento”.*

Con el desarrollo del audio y el vídeo, aunado a la llegada de Internet se ha intentado hacer realidad uno de los viejos sueños del hombre, la vídeo conferencia, la transmisión de "multimedia" por Internet. Mucha gente ha visto películas futuristas donde se mostraban conversaciones a través de una pantalla que siempre identificamos con un monitor de ordenador (de hecho siempre lo asemejamos a una conversación gracias a un ordenador). Todo esto ya casi es una realidad, con limitaciones claro.

Los ordenadores necesarios están al alcance de todo el mundo, siempre y cuando no queramos hacer grandes cosas. Con un equipo que casi todo el mundo tiene (Pentium a 100MHz como mínimo) podemos hacer nuestras primeras pruebas en estos temas. Esto es algo que tenemos a favor, otro punto muy importante a tener en cuenta, es la velocidad de Internet. Internet actualmente es sinónimo de lentitud, aquellos que se conecten vía telefónica verán que es realmente lenta consiguiendo rangos de 1 Kbps con mucha suerte a horas punta y con un módem rápido (aproximadamente de unos 33.000 baudios), con RDSI pueden conseguirse medias de 4 o 5 Kbps, algo mejor pero aun no es suficiente como podemos observar. Un gran avance se ha alcanzado en la actualidad al tener líneas dedicadas bajo los servicios xDSL, la ventaja de estas tecnologías es el ancho de banda que pueden llegar a ofrecer, siendo el mínimo de 256 Kbps.

El desarrollo de la tecnología esta permitiendo que la transmisión de Audio y Vídeo sea cada vez de una mejor calidad, pero es importante tener presente que Internet no es una red diseñada para la transmisión de estos dos tipos de señales, la infraestructura actual de Internet fue diseñada para la transmisión de datos. Fue gracias a que se realizó la conversión a señales digitales de vídeo y audio como realmente se pudieron ver a estas dos señales como datos.

En el presente capítulo tenemos dos problemas a resolver, primero esta el relacionado con el análisis de audio para su codificación y compresión y de igual manera el análisis de vídeo. El presente capítulo tiene por objetivo presentar las características de cada una de las señales y como las interpretamos de manera digital.

¿Por qué incluir Audio y Vídeo? Para nuestro proyecto creemos indispensable que la transmisión de estas dos señales nos permitirá tener un mejor control del brazo, ya que

así podemos ver y escuchar el ambiente sobre el cual esta trabajando el brazo, dando así una sensación perceptible que permita al usuario quedar sumergido en el comportamiento del brazo de acuerdo a las recepciones de audio y vídeo.

Comenzamos analizando el audio primero y después la parte correspondiente a vídeo, para finalmente decidir como trataremos a ambas señales y sobre que algoritmos y programas nos apoyaremos para implementar la transmisión de Audio y Vídeo.

### **3.1.- Análisis de la Señal de Audio.**

Comencemos analizando de manera analógica las características del audio y quien es el encargado de percibir esta señal.

#### **Conceptos básicos de anatomía y audición.**

El proceso de audición se realiza a través del oído. Lo que entendemos por "oír" sonidos es sentir micro variaciones de presión en el aire que nos rodea, las cuales después serán interpretadas convenientemente por nuestro cerebro.

El oído es sensible en un rango de frecuencias que va desde los 20 Hz hasta los 20 KHz, aunque pocas personas son capaces de percibir vibraciones estacionarias de frecuencia superior a 17 KHz. Esta es la razón por la que muchos codificadores ofrecen la posibilidad de filtrar a 16 KHz para comprimir aún más. No obstante esa zona es importante. Aunque no seamos capaces de escuchar "algo" que esté en esa zona el oído es un sistema no lineal, produciéndose mezclas de frecuencias, de modo que indirectamente percibimos esas frecuencias. Eso es importante para sonidos con transitorios bruscos y es lo que marca la diferencia entre algo que suena bien (un concierto de música clásica) y algo que suena natural (un amplificador de 3.000 Watts, que tiene rangos de frecuencia aún mayores). No obstante para la mayoría de las aplicaciones no es necesaria tanta calidad.

En cuanto al rango dinámico del oído (definido como la relación entre la señal más fuerte y la más débil que puede oírse) es de 120 dB. La referencia en cuanto a intensidades sonoras se establece en  $20 \times 10^{-3} \text{ N/m}^2$ , que es precisamente el umbral de audición a 1 KHz. Por encima de 120 dB se produce la sensación de dolor, pudiendo provocar daños irreversibles. Un ejemplo de este tipo es un martillo neumático o un motor de reacción. Para poder representar todo el rango dinámico del oído mediante una señal digital necesitaríamos cuantificar las muestras con 20 bits. Sin embargo esto sólo es posible y necesario en equipos de gama muy alta, porque el ruido electrónico de algunos componentes puede ser superior al del valor mínimo del escalón de cuantificación. Además, el ruido ambiente que hay en la mayoría de los lugares es muy superior al de los sonidos más débiles representables. Todo esto hace que lo normal sea utilizar 16 bits, lo que da un rango dinámico de 96 dB. El nivel sonoro se suele medir en dB debido a que nuestro oído tiene sensibilidad logarítmica, es decir un sonido del doble de potencia que otro no es percibido como tal por el oído, sino que se necesita que sea 10 veces más intenso para que la sensación sonora sea del doble. Este efecto permite al oído tener un rango dinámico muy grande.

El oído consta de tres partes bien diferenciadas y delimitadas:

El oído externo. Está compuesto por el pabellón auditivo, el conducto auditivo y el tímpano. Básicamente actúa como una antena.

El oído medio. El elemento principal es la cadena de huesillos, que se encarga de transmitir las vibraciones acústicas con las mínimas pérdidas de potencia entre dos medios, aire en el tímpano y una solución salina en el oído interno.

El oído interno. En esta parte es donde realmente se realiza la audición. El órgano más destacable es la membrana basilar, que se encuentra dentro del caracol, y conecta con las fibras nerviosas que llegan al cerebro. Su comportamiento se asemeja al de una cuerda fija por un extremo y moviéndose por el otro.

Cuando una onda de presión llega al oído, lo atraviesa sufriendo diversas transformaciones hasta llegar a la membrana basilar. Ésta vibra como una cuerda fija en un extremo y moviéndose por el otro, con la particularidad de que en cada punto está sometida a una tensión diferente y es de diferente tamaño. Todo esto hace que ante una excitación de una determinada frecuencia, un cierto punto (en realidad una pequeña zona) vibre con mayor intensidad, estableciéndose una correspondencia entre frecuencia y posición del máximo de vibración, bajas frecuencias al final y altas frecuencias al principio. Este comportamiento se asemeja mucho al de un analizador de espectros. Esta vibración excita las terminaciones nerviosas del nervio auditivo, siendo mayor la excitación cuanto mayor es la vibración, que a su vez depende del nivel del sonido de entrada.

### **¿Cómo comprimimos audio?**

Cuando cuantificamos una señal, lo que hacemos es asignar a un rango de niveles a un único valor de reconstrucción. Esto hace más fácil discernir entre niveles de amplitud, reduciendo el efecto del ruido que se pueda añadir en una transmisión o en un proceso de lectura. El precio que hay que pagar es una distorsión de la señal, ya que esta no recupera su amplitud original en todos los puntos, sino un valor próximo, que nosotros le hemos asignado. Esta distorsión puede verse como ruido añadido, en una proporción que podemos controlar variando el número de niveles de cuantificación: cuantos más niveles, menos ruido.

Esto nos da la clave para comprimir audio. Si en una zona del espectro podemos introducir ruido sin que se oiga, realizamos una cuantificación menos fina (escalones de cuantificación más grandes, que se traduce en menos bits), mientras que en las zonas donde el ruido se hace audible, asignamos más bits. Es en este punto donde se diferencian unos codificadores de otros.

Un hecho que debe haber quedado claro ya es que al comprimir audio se pierde cierta calidad. Esta pérdida no es apreciable en circunstancias habituales, pero sin embargo se hace patente en entornos como la edición musical.

Imaginemos una señal ya comprimida. En una cierta banda, además de la señal ya presente, existe una cierta cantidad de ruido, resultado del proceso de compresión, y en una cantidad que depende directamente de la potencia de la señal original. Ahora tomamos el ecualizador

de algún reproductor y aumentamos la potencia de esa banda (podría ser el realce de graves o agudos). En esta nueva situación, las bandas próximas ya no enmascaran tanto, por lo que la cantidad de ruido admisible es menor, pero al aumentar la potencia de la señal con el ecualizador, también hemos aumentado la potencia de ruido, por lo que pasado un límite empieza a oírse claramente la distorsión.

Este ejemplo tan sencillo ilustra el gran inconveniente del sonido comprimido: no se puede post-procesar. No podemos tener una grabación de estudio, comprimirla y después procesarla, porque el resultado puede ser desastroso, y lo que es peor, lo que para unas grabaciones puede funcionar para otras seguramente no lo hará, lo que no permite automatizar procesos o hacer generalizaciones sobre el comportamiento de determinados efectos (como por ejemplo la reverberación).

### **Compresiones On-line.**

Estas compresiones son las utilizadas para oír audio en tiempo real. De aquí en adelante tenemos que considerar el tipo de transmisión, es decir, para quien se transmite, para esto tenemos varios casos:

A una sola persona o unicast. De este modo se emite sólo a una persona, directamente a ese ordenador; los paquetes (suponemos transmisión TCP/IP) van todos asignados con la IP de destino. Por ejemplo, si transmitimos a: "luna.gui.uva.es", sería el identificador del ordenador 157.88.36.190.

A un grupo reducido de personas o multicast. Aquí transmitimos a una serie de personas, normalmente a un dominio (un grupo de ordenadores con una IP más o menos parecida). Así lo transmitido, lo recibirá solo un grupo de personas (por ejemplo si queremos emitir para todos los ordenadores del Grupo Universitario de Informática sería emitir para 157.88.36.0).

A todo el mundo o broadcast, emitimos en lo que se considera alcance mundial, es decir, para todo el mundo que quiera recibirlo.

Tanto el broadcast como el multicast tienen un gran parecido, siendo su diferencia el alcance de la emisión (el número de routers por los que ha de pasar el paquete, el número ttl como se denomina).

Volviendo a las compresiones on-line, este tipo de compresiones tiene pocos estándares, siendo algunos de ellos los utilizados de manera offline (\*.ra, \*.mp3, \*.avi, etc.) Todos estos compresores tienen la desventaja de una calidad baja, debido a que así se puede aumentar la eficiencia de transmisión.

## **3.2.- Análisis de la Señal de Vídeo.**

Comunicarnos con otra persona solo mediante la voz es algo que esta bastante bien, de hecho lo usamos siempre que llamamos por teléfono, pero si a esto le añadimos el Vídeo conseguiremos un sistema de comunicación ideal a distancia y con un enorme acercamiento en lo correspondiente a la comunicación humana.

Con el Vídeo pasa lo mismo que con el audio, transmitir un segundo de Vídeo ocupa lo inimaginable, así si usamos uno de los medios de almacenamiento masivo (un CD-ROM), a máxima calidad (sistema PAL, 768x576, Truecolor, 30 frames o imágenes por segundo) solo tendríamos la posibilidad de tener 3 o 4 segundos, eso solo en cada CD. Por eso tenemos que hacer lo mismo que con el audio, comprimir.

Todo esta en desarrollo, aun no esta mejorado ya que se necesita un gran ancho de banda, muchísimo mayor que para el audio. Aunque comprimamos, si queremos enviar una imagen de 320x200 de resolución, multiplicando nos da cosa de 64000 pixeles que son aproximadamente 60 Kbps, pero si se quiere transmitirla con movimiento son 30 cuadros por segundo que multiplicando a lo anterior son 1.8 Mbps, esto es una cifra astronómica para las redes actuales con anchos de banda de 28 Kbps.

Tenemos que diferenciar el tipo de compresión: puede que no haya compresión alguna en los cuadros, que haya lo que se llama un cuadro maestro, es decir, no se transmite todo el fotograma cada vez, sino que hay un cuadro que se transmite entero y los demás son deformaciones de este (realmente los datos que varían) o un tipo híbrido con cuadro maestro, con compresión y con lo que se denomina predicción de cuadro, (algo muy nuevo) donde el ordenador prevé cual será el cambio del fotograma al fotograma siguiente.

### **3.3.- Audio y Vídeo junto.**

La transmisión de vídeo sobre redes de telecomunicaciones está llegando al punto de convertirse en un sistema habitual de comunicación debido al crecimiento masivo que ha supuesto Internet en estos últimos años. Lo estamos utilizando para ver películas o comunicarnos con conocidos, pero también se usa para dar clases remotas, para hacer diagnósticos en medicina, videoconferencia, distribución de TV, vídeo bajo demanda, para distribuir multimedia en Internet.

Debido a la necesidad de su uso que se plantea en el presente y futuro, se han proporcionado distintas soluciones y sucesivos formatos para mejorar su transmisión.

Estas aplicaciones normalmente demandan un elevado ancho de banda y a menudo crean cuellos de botella en las redes. Este es el gran problema al que esta sometida la transmisión de vídeo. ¿Por qué es el vídeo tan problemático?

#### **Vídeo digital.**

La transmisión digital y la distribución de información audiovisual nos permite la comunicación multimedia sobre las redes que soportan la comunicación de datos, brindando la posibilidad de enviar imágenes en movimiento a lugares remotos. Pero todavía tenemos inconvenientes para transmitirlo por red, debido a que nos encontramos con sucesos como lentitud entre la reproducción de imágenes, errores de transmisión, o pérdidas de datos.

Existen dos formas de transmisión de datos, analógica y digital. Una de las características del vídeo es que está compuesto por señales analógicas, con lo que se pueden dar las dos formas de transmisión. En los últimos años la transmisión de datos se ha volcado hacia el

mundo digital ya que supone una serie de ventajas frente a la transmisión analógica. Al verse la información reducida a un flujo de bits, se consigue una mayor protección contra posibles fallos ya que se pueden introducir mecanismos de detección de errores, se elimina el problema de las interferencias, podemos disminuir el efecto del ruido en los canales de comunicación, conseguir codificaciones más óptimas y encriptado, mezclar con otros tipos de información a través de un mismo canal, y poder manipular los datos con ordenadores.

Además si queremos difundir el vídeo por vías digitales, con lo que debe ser capturado en su formato analógico y almacenado digitalmente logrando así que sea menos propenso a degradarse durante la transmisión.

Existen dos tipos de redes de comunicación, de conmutación de circuitos y de conmutación de paquetes. En la conmutación de circuitos, donde la comunicación está permanentemente establecida durante toda la sesión, un determinado ancho de banda es asignado para la conexión, y el tiempo de descarga del vídeo puede predecirse, pero tienen la desventaja de que las sesiones son punto a punto y limitan la capacidad de usuarios.

En la conmutación de paquetes pueden acomodarse más fácilmente las conferencias multipunto. Aquí el ancho de banda está compartido pero es variable, lo que supone una importante mejora puesto que, si la razón de bits es fija la calidad de la imagen variará dependiendo del contenido de los fotogramas. Debe cumplirse que el ancho de banda, la resolución, y la compresión de audio sean idénticos para cada cliente que recibe el vídeo, lo que dificulta la configuración del sistema.

El vídeo es muy sensible al retardo de la red, ya que puede provocar cortes en la secuencia. La pérdida de alguna información en el vídeo sin comprimir no es muy relevante, ya que al perderse un fotograma, el siguiente fotograma proporciona la suficiente información para poder interpretar la secuencia. En cambio el vídeo comprimido es mucho más sensible a errores de transmisión, ya que las técnicas de compresión que se valen de la redundancia espacial y temporal pueden perder la información de esta redundancia y los efectos de la falta de datos pueden propagarse en los próximos fotogramas. Es por eso que actualmente la comunicación con vídeo vía Internet no promete una elevada fiabilidad de transmisión.

Algunas técnicas de compresión compensan esta sensibilidad a la pérdida de datos enviando la información completa sobre un fotograma cada cierto tiempo, incluso si los datos del fotograma no han cambiado. Esta técnica también es útil para los sistemas de múltiples clientes, para que los usuarios que acaban de conectarse, reciban las imágenes completas.

Nos podemos preguntar cuál es la tecnología de red adecuada para las aplicaciones de vídeo, pero siempre dependeremos del entorno en el que trabajemos. Por ejemplo si disponemos de un alto ancho de banda el tipo de red adecuada sería ATM; para un entorno de red de área local podríamos usar Fast Ethernet, y actualmente para que el usuario de Internet lo utilice existe, ADSL.

Pero la solución para resolver el cuello de botella del ancho de banda del vídeo no está en un solo tipo de red, sino en una infraestructura de red flexible que pueda manejar e integrar diferentes redes y que deje paso también a futuras redes sin cambiar el hardware. También debe ser capaz de negociar las variables de ancho de banda, resolución, número de

fotogramas por segundo y algoritmo de compresión de audio. Así que se necesita un nodo que permita la interconectividad entre todas las redes. Es el MCU (unidad de control multipunto). Cada red- RDSI, IP, ATM- usa protocolos específicos que definen la naturaleza de las ráfagas de vídeo. Las combinaciones de protocolos y estándares son muchas: para vídeo H.261 o H.263, CIF o QCIF, de 7.5 fps a 30 fps; y para audio G.711, G.728, G.722 o G.723. Por ejemplo en una conferencia múltiple el número de posibles combinaciones de estándares y protocolos es muy elevado y puede saturar el MCU. Muchos MCU no son capaces de negociar todas estas variables, forzando a los terminales de los clientes a reducir sus protocolos al más bajo común denominador de todos los participantes, bajando así la calidad del vídeo.

### 3.4.- Digitalización.

La información a digitalizar será la de las imágenes. Cada cuadro de la imagen es muestreado en unidades de píxeles, con lo que los datos a almacenar serán los correspondientes al color de cada pixel.

Tres componentes son necesarias y suficientes para representar el color y para ser interpretado por el ojo humano. El sistema de codificación de color usado es el RGB (Red, Green, Blue).

Para digitalizar una señal de vídeo analógico es necesario muestrear todas las líneas de vídeo activo. La información de brillo y color son tratadas de forma diferente por el sistema visual humano, ya que es más sensible al brillo que al color. Con lo que se usa un componente especial para representar la información del brillo, la luminancia, una para el color y la saturación, la crominancia. Cada muestra de color se codifica en señal Y-U-V (Y- luminancia, U y V crominancia) partiendo de los valores del sistema RGB.

Con este sistema las diferencias de color pueden ser muestreadas sin resultados visibles, lo que permite que la misma información sea codificada con menos ancho de banda.

Un ejemplo de conversión de señal analógica de televisión en color a una señal en vídeo digital sería:

Sistema PAL: 576 líneas activas, 25 fotogramas por segundo, para obtener 720 píxeles y 8 bit por muestra a 13,5Mhz:

- Luminancia(Y):  $720 \times 576 \times 25 \times 8 = 82,944$  Kbps.
- Crominancia(U):  $360 \times 576 \times 25 \times 8 = 41,472$  Kbps.
- Crominancia(V):  $360 \times 576 \times 25 \times 8 = 41,472$  Kbps.

Número total de bits: 165.888 Kbps (aprox. 166Mbps). Ninguno de los sistemas comunes de transmisión de vídeo proporciona transferencias suficientes para este caudal de información.

Las imágenes de vídeo están compuestas de información en el dominio del espacio y el tiempo. La información en el dominio del espacio es provista por los píxeles, y la información en el dominio del tiempo es provista por imágenes que cambian en el tiempo.

Puesto que los cambios entre cuadros colindantes son diminutos, los objetos aparentan moverse suavemente.

El valor de luminancia de cada pixel es cuantificado con ocho bits para el caso de imágenes blanco y negro. En el caso de imágenes de color, cada pixel mantiene la información de color asociada; una imagen completa es una composición de tres fotogramas, uno para cada componente de color, así los tres elementos de la información de luminancia designados como rojo, verde y azul, son cuantificados a ocho bits.

Pero la transmisión digital de vídeo tiene también alguna desventaja respecto a la analógica, por ejemplo, en una videoconferencia, cuando distintos usuarios envían sonido al mismo tiempo, si el proceso fuera analógico las distintas ondas se sumarían y podríamos escuchar el conjunto de todas ellas. Al ser digital, los datos llegan en paquetes entremezclados, lo que dificulta la comprensión.

### **Compresión de vídeo.**

La técnica de compresión de vídeo consiste de tres pasos fundamentalmente, primero el pre-procesamiento de la fuente de vídeo de entrada, paso en el cual se realiza el filtrado de la señal de entrada para remover componentes no útiles y el ruido que pudiera haber en esta. El segundo paso es la conversión de la señal a un formato intermedio común (CIF), y por último el paso de la compresión. Las imágenes comprimidas son transmitidas a través de la línea de transmisión digital y se hacen llegar al receptor donde son reconvertidas al formato común CIF y son desplegadas después de haber pasado por la etapa de post-procesamiento.

Mediante la compresión de la imagen se elimina información redundante. Se ayuda de la redundancia espacial y temporal. La redundancia temporal es reducida primero usando similitudes entre sucesivas imágenes, usando información de las imágenes ya enviadas. Cuando se usa esta técnica, sólo es necesario enviar la diferencia entre las imágenes, es decir las zonas de la imagen que han variado entre dos fotogramas consecutivos, lo que elimina la necesidad de transmitir la imagen completa. La compresión espacial se vale de las similitudes entre pixeles adyacentes en zonas de la imagen lisa, y de las frecuencias espaciales dominantes en zonas de color muy variado.

### **3.5.- Distribuir el vídeo.**

Muchas aplicaciones actuales como el vídeo requieren que los mismos datos de un servidor sean distribuidos a múltiples clientes. Si varios clientes solicitan los mismos datos y esta información fuera enviado una vez por cada cliente, estaríamos malgastando el ancho de banda ya que estaríamos transmitiendo la misma información varias veces por el mismo tramo de red y el número de clientes estaría limitado por el ancho de banda disponible. La solución es IP multicast. Soporta eficientemente este tipo de transmisión permitiendo al servidor enviar una sola copia de la información a múltiples clientes quienes deseen recibir la información.

Existen dos métodos para la distribución de contenido con audio y vídeo sobre la Web. El primer método usa un Servidor Web estándar para repartir los datos a un medio visualizador. El segundo método usa un servidor de Streaming.

¿Cómo funciona un servidor Web para distribución de vídeo? Una vez que disponemos del vídeo digitalizado el archivo será codificado o comprimido a un fichero para ser distribuido sobre una red con un específico ancho de banda como un módem de 56.6 Kbps. Entonces el fichero se almacena en un servidor Web. Ahora sólo hemos de crear una página Web en un servidor con un enlace al fichero, el cual cuando sea pulsado por un cliente permitirá la descarga automática.

El protocolo usado es el HTTP (Hypertext Transport Protocol), que opera en la parte alta del TCP el cual maneja la transferencia de datos. Este protocolo no está optimizado para aplicaciones en tiempo real, TCP manda primero datos con una baja razón, y gradualmente va incrementando la razón hasta que el destinatario comunica una pérdida de paquetes. Entonces el TCP asume que ha llegado al límite del ancho de banda y vuelve a enviar datos a baja velocidad, y volverá a incrementar la velocidad repitiendo el proceso anterior. TCP se asegura de una fiable transmisión de datos con la retransmisión de los paquetes perdidos, recordemos que TCP es un protocolo orientado a conexión. Sin embargo lo que no puede asegurar es que todos los paquetes recientes llegarán al cliente para ser visualizados a tiempo, con lo que podremos experimentar pérdida de imágenes en las secuencias de vídeo.

La siguiente figura muestra como estarían interconectados los equipos para este tipo de distribución.

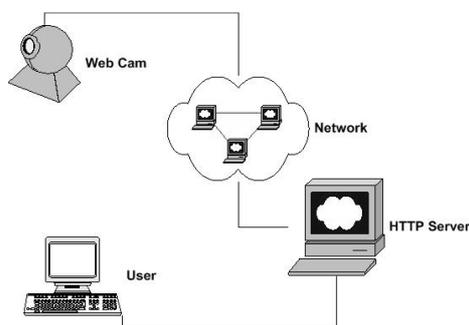


Figura 3.1. Distribución mediante servidor HTTP.

Streaming Video, o vídeo en tiempo real, es la tecnología que permite la transmisión y recepción de imágenes y sonidos de manera continua a través de una red. A diferencia de otros formatos de audio y vídeo, en los que es necesario esperar que el archivo sea cargado en el equipo para su visualización, esta tecnología permite apreciar el contenido conforme se va teniendo acceso a la información del archivo.

El servidor de Streaming permite visionar el vídeo de forma continua porque hace uso de un buffer, donde van cargándose algunos segundos de la secuencia antes de que sean mostrados. Entonces cuando se detecta un periodo de congestión de red, se visualizarán los datos que tenemos ya almacenados en el buffer. De esta forma el cliente obtiene los datos tan rápido como el servidor y la red lo permitan.

El Streaming puede decirse que funciona de forma inteligente ya que asegura al usuario que recibirá la más alta calidad posible dependiendo de la velocidad de conexión o de los problemas de conexión de la red. Tradicionalmente la congestión de la red forzaba al usuario a detener la visualización del vídeo almacenando en un buffer la información para posteriormente continuar mostrando la secuencia. Con los nuevos formatos de Streaming como el MPEG-4, el cliente y el servidor pueden degradar la calidad de forma inteligente para asegurar una reproducción continua del vídeo. La siguiente figura nos muestra este tipo de interconexión.

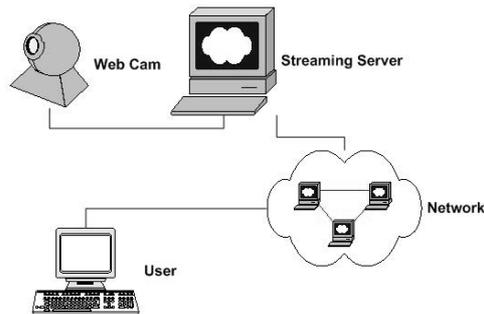


Figura 3.2.- Distribución mediante servidor Streaming.

Podemos concluir en este capítulo que la transmisión de audio y vídeo a través de la red ha tenido que llevar a varias organizaciones a estandarizar, desarrollar y aplicar técnicas cada vez más eficientes para crear un nuevo tipo de comunicación antes impensable dentro de nuestro mundo, pero gracias a la evolución de las tecnologías es como podemos vislumbrar un nuevo futuro, en donde cada individuo o grupo de personas se encuentren comunicadas y permitan la convergencia de todas las culturas para un mejor crecimiento y desarrollo de la humanidad.

# 4

## COMUNICACIONES DE DATOS.

---

*“La vida es hermosa, vivirla no es una casualidad”.*

En este capítulo presentamos los elementos que debemos de considerar para la transmisión de señales a través de nuestro sistema de telemático. Partiendo del modelo de referencia OSI (Open Systems Interconection) propuesto por ISO (International Standard Organization) podremos ubicar los componentes que interactúan en nuestro enlace de comunicaciones, dando una descripción estratificada de los elementos que se pueden involucrar en nuestro enlace de comunicaciones.

La transmisión de señales la realizaremos con las siguientes características:

- El ancho de banda de las señales de control será de 20Hz.
- Serán Totalmente digitales.
- Utilizaremos la infraestructura de Internet.
- La programación será mediante sockets.

### 4.1.- Modelo de referencia OSI/ISO.

Partiendo de este modelo de referencia es como se puede exponer de manera más explícita la forma en como se visualiza el proyecto visto desde el punto de vista de las comunicaciones de datos. La siguiente figura nos muestra la forma en que este modelo se estratifica:



Figura 4.1.- Modelo de referencia OSI/ISO.

**Física.** Se refiere a todos los aspectos electrónicos y mecánicos de los diferentes dispositivos que tienen como objetivo transmitir señales a través de un medio de transmisión dado, también aquí se deben de contemplar todas las características y las diferentes naturalezas por las que tendrá que ser transmitida nuestra señal; esto es, no podemos ver a un bit sólo como la representación de un uno o cero lógico puesto que omitiríamos muchos detalles de la transmisión de información, viendo desde el punto de vista telemático tenemos lo siguiente: De manera física tenemos almacenado en la memoria de la computadora un estado lógico (uno o cero) que me representa cierta información, este bit de información tiene que ser transmitido a través de un canal de comunicaciones, interactuando la electrónica que permite la transferencia del bit de la memoria de la computadora a la tarjeta de comunicaciones (en este caso una tarjeta que utilice tecnología Ethernet), aquí la naturaleza de la señal es puramente eléctrica y requiere de cierto tiempo para hacer esa transferencia, ya estando en la memoria temporal de la tarjeta de comunicaciones se tiene que enviar ese bit a través del medio de transmisión que utiliza la tecnología antes mencionada (para nuestro caso un cable par trenzado categoría 5 o 6 según sea el caso) aquí entra el concepto de tiempo de propagación que para el presente proyecto es despreciable, esta señal se envía a un equipo que hace una conversión de la señal eléctrica en una onda electromagnética (punto de acceso inalámbrico) aquí ya tenemos presente un cambio en la naturaleza de la señal, además de tener otro tipo de factores que si pueden incidir en el tiempo de propagación, al ser ahora el aire el medio de transmisión para ser utilizado.

Hasta aquí no hemos considerado toda la electrónica que se debe de involucrar para la digitalización de las señales provenientes del brazo humano, esto ya que si lo vemos desde el punto de vista de comunicaciones la parte física correspondiente a la digitalización es para nosotros una capa superior; la capa de Aplicación. Ahora bien, de todas estas características se encarga la capa física dentro del modelo de referencia OSI.

**Enlace de datos.** Esta capa dentro del modelo se encarga de hacer los algoritmos correspondientes para el control de flujo y detección de errores, se utiliza un tipo de direcciones conocidas como físicas y son las direcciones que sirven para identificar de manera unívoca un equipo de comunicaciones dentro de una red de datos. La particularidad de esta capa dentro de nuestro proyecto esta implícita en la tecnología que lleguemos a utilizar.

**Red.** Esta capa se encarga de proveer el direccionamiento de información entre dos entidades dentro de una red de datos, como impresoras, pc's de escritorio o portátiles y cualquier otro dispositivo que pueda ser integrado dentro de una red. Aquí se utilizan por lo regular direcciones IP (Internet Protocol).

**Transporte.** Esta capa se encarga de establecer el flujo de información y los tipos de conexiones a realizar, esto es, ya sea que la información que enviemos este orientada a conexión o no orientada a conexión. Aquí utilizamos por lo general dos protocolos de comunicaciones correspondientes a estas capas, el TCP (Transfer Control Protocol) para el caso de la transmisión de señales provenientes del brazo humano y el UDP (User Datagram Protocol) para la transmisión de audio y video.

**Sesión.** Esta capa se encarga de establecer los parámetros y la forma en que los datos deben de ser transmitidos para las aplicaciones superiores. Para el presente proyecto, esto esta en función de la programación que se lleve a cabo para el usuario final.

**Presentación.** En esta capa se tiene la información y se le da el formato necesario para ser enviada a la siguiente capa, también se encarga de llevar a acabo la organización de la información para ser enviada a las capas inferiores. De igual manera para nosotros estas capas están más en función de la programación que llevemos a cabo para la correspondiente transmisión de la información.

**Aplicación.** Esta es la programación final que ve el usuario y la cual nos proporciona la interfaz o interfaces necesarias para la transmisión de información, y despliegue de señales de audio, video, control, etc. Aquí participamos de manera más activa puesto que esto involucra todo el desarrollo de la aplicación para el usuario, así como la integración de los diferentes programas que serán utilizados para el control y transmisión de información al y desde el robot.

## 4.2.- Transmisión de Datos.

### Terminología utilizada en transmisión de datos.

Los medios de transmisión pueden ser:

- Guiados si las ondas electromagnéticas van encaminadas a lo largo de un camino físico; no guiados si el medio es sin encauzar (aire, agua, etc.)
- Simplex si la señal es unidireccional; half-duplex si ambas estaciones pueden transmitir pero no ambas al mismo tiempo; full-duplex si ambas estaciones pueden transmitir al mismo tiempo.

### Frecuencia, Espectro y Ancho de Banda.

Conceptos en el dominio temporal. Una señal, en el ámbito temporal, puede ser continua o discreta. Puede ser periódica o no periódica. Una señal es periódica si se repite en intervalos de tiempo fijos llamados periodo.

Conceptos del dominio de la frecuencia. En la práctica, una señal electromagnética está compuesta por muchas frecuencias. Si todas las frecuencias son múltiplos de una dada, esa frecuencia se llama frecuencia fundamental. El periodo (o inversa de la frecuencia) de la señal suma de componentes es el periodo de la frecuencia fundamental. Se puede demostrar que cualquier señal está constituida por diversas frecuencias de una señal seno. El espectro de una señal es el conjunto de frecuencias que constituyen la señal.

El ancho de banda es la anchura del espectro. Muchas señales tienen un ancho de banda infinito, pero la mayoría de la energía está concentrada en un ancho de banda pequeño. Si una señal tiene una componente de frecuencia 0, es una componente continua.

Relación entre la velocidad de transmisión y el ancho de banda. El medio de transmisión de las señales limita mucho las componentes de frecuencia a las que puede ir la señal, por lo que el medio sólo permite la transmisión de cierto ancho de banda.

En el caso de ondas cuadradas (binarias), estas se pueden simular con ondas senoidales en las que la señal sólo contenga múltiplos impares de la frecuencia fundamental. Cuanto más ancho de banda, más se asemeja la función seno (multifrecuencia) a la onda cuadrada. Pero generalmente es suficiente con las tres primeras componentes.

Se puede demostrar que al duplicar el ancho de banda, se duplica la velocidad de transmisión a la que puede ir la señal.

Al considerar que el ancho de banda de una señal está concentrado sobre una frecuencia central, al aumentar esta, aumenta la velocidad potencial de transmitir la señal.

Pero al aumentar el ancho de banda, aumenta el coste de transmisión de la señal aunque disminuye la distorsión y la posibilidad de ocurrencia de errores.

### **4.3.- Transmisión de Datos Analógicos y Digitales.**

Los datos analógicos toman valores continuos y los digitales, valores discretos. Una señal analógica es una señal continua que se propaga por ciertos medios. Una señal digital es una serie de pulsos que se transmiten a través de un cable ya que son pulsos eléctricos.

Los datos analógicos se pueden representar por una señal electromagnética con el mismo espectro que los datos. Los datos digitales se suelen representar por una serie de pulsos de tensión que representan los valores binarios de la señal. La transmisión analógica es una forma de transmitir señales analógicas (que pueden contener datos analógicos o datos digitales). El problema de la transmisión analógica es que la señal se debilita con la distancia, por lo que hay que utilizar amplificadores de señal cada cierta distancia. A su vez la transmisión digital tiene el problema de que la señal se atenúa y distorsiona con la distancia, por lo que cada cierta distancia hay que introducir repetidores de señal.

Últimamente se utiliza mucho la transmisión digital debido a que:

- La tecnología digital se ha abaratado mucho.
- Al usar repetidores en vez de amplificadores, el ruido y otras distorsiones no es acumulativo.
- La utilización de banda ancha es más aprovechada por la tecnología digital.
- Los datos transportados se pueden cifrar y por tanto hay más seguridad en la información.
- Al tratar digitalmente todas las señales, se pueden integrar servicios de datos analógicos (voz, vídeo, etc.) con digitales como texto y otros.

#### **4.4.- Perturbaciones en la Transmisión.**

##### **Atenuación.**

La energía de una señal decae con la distancia, esto debido al medio de propagación, por lo que hay que asegurarse que llegue con la suficiente energía como para ser captada por la electrónica del receptor y además, el ruido debe ser sensiblemente menor que la señal original (para mantener la energía de la señal se utilizan amplificadores o repetidores).

Debido a que la atenuación varía en función de la frecuencia, las señales analógicas llegan distorsionadas, por lo que hay que utilizar sistemas que le devuelvan a la señal sus características iniciales (usando bobinas que cambian las características eléctricas o amplificando más las frecuencias más altas).

##### **Distorsión de retardo.**

Debido a que en medios guiados, la velocidad de propagación de una señal varía con la frecuencia, hay frecuencias que llegan antes que otras dentro de la misma señal y por tanto las diferentes componentes en frecuencia de la señal llegan en instantes diferentes al receptor. Para atenuar este problema se usan técnicas de ecualización.

##### **Ruido.**

El ruido es toda aquella señal que se inserta entre el emisor y el receptor de una señal dada. Hay diferentes tipos de ruido: ruido térmico debido a la agitación térmica de electrones dentro del conductor, ruido de intermodulación cuando distintas frecuencias comparten el mismo medio de transmisión, diafonía se produce cuando hay un acoplamiento entre las líneas que transportan las señales y el ruido impulsivo se trata de pulsos discontinuos de poca duración y de gran amplitud que afectan a la señal.

##### **Capacidad del canal.**

Se llama capacidad del canal a la velocidad a la que se pueden transmitir los datos en un canal de comunicación de datos. La velocidad de los datos es la velocidad expresada en bits por segundo a la que se pueden transmitir los datos.

El ancho de banda es aquel ancho de banda de la señal transmitida y que está limitado por el transmisor y por la naturaleza del medio de transmisión. La tasa de errores es la razón a la que ocurren errores.

Para un ancho de banda determinado es aconsejable la mayor velocidad de transmisión posible pero de forma que no se supere la tasa de errores aconsejable. Para conseguir esto, el mayor inconveniente es el ruido.

Para un ancho de banda dado  $W$ , la mayor velocidad de transmisión posible es  $2W$ , pero si se permite (con señales digitales) codificar más de un bit en cada ciclo, es posible transmitir más cantidad de información.

La formulación de Nyquist nos dice que aumentando los niveles de tensión diferenciables en la señal, es posible incrementar la cantidad de información transmitida.

$$C = 2W \log_2 M$$

El problema de esta técnica es que el receptor debe de ser capaz de diferenciar más niveles de tensión en la señal recibida, cosa que es dificultada por el ruido.

Cuanto mayor es la velocidad de transmisión, mayor es el daño que puede ocasionar el ruido.

Shannon propuso la fórmula que relaciona la potencia de la señal (S), la potencia del ruido (N), la capacidad del canal (C) y el ancho de banda (W).

$$C = W \log_2 (1 + S/N)$$

Esta capacidad es la capacidad máxima teórica de cantidad de transmisión, pero en la realidad, es menor debido a que no se ha tenido en cuenta nada más que el ruido térmico.

Podemos ver que para la parte de redes es necesario tener bien definido que tipo de arquitectura a nivel de redes debemos de emplear, para nuestros propósitos utilizaremos la familia de protocolos TCP/IP, ya que nos dan la mejor flexibilidad y nos permiten tener un mejor control de la información, utilizaremos IPv4 como el protocolo de trabajo.

## PLANTEAMIENTO DEL PROBLEMA.

---

*“Cada día sabemos más y entendemos menos”.*

En el presente capítulo daremos paso a plantear desde los dos enfoques principales, que son los enlaces de comunicación tanto de bajada como de subida, así como las propuestas de solución para cada una de estas situaciones.

### **5.1.- Planteamiento del problema: Enlace de Bajada.**

El problema principal del enlace de bajada es hacer que la información digitalizada del movimiento de un brazo humano sea transmitida al brazo robot en un lugar lejano. Para poder lograr este objetivo se debe de hacer un análisis de la transmisión de datos, y por todas las operaciones involucradas para poder enviar esta información.

#### **5.1.1.- Transmisión de los Datos.**

Para transmitir señales digitales a través de cualquier canal, primero es necesario implementar servicios que son básicos para la transmisión de señales; tales como sincronización, señalización, implementación de un código de línea, línea de transmisión, diseño de circuitos de regeneración de señal. Estos servicios que nos proporciona la capa física ya están implementados en la tecnología a utilizar, esto es la tarjeta de red, el punto de acceso y la tarjeta de red inalámbrica contienen esta implementación.

Para la parte de control de flujo, códigos de corrección de errores, ventaneo de tramas, contingencias de error, tenemos los servicios que nos proporciona la capa de enlace de datos del modelo OSI. De igual manera la tecnología a emplear nos proporciona estas características dentro de la electrónica comprendida en la tarjeta de red.

Para la creación de tablas de ruteo, asignación de direcciones, asociación de direcciones, que son servicios proporcionados por la capa de red del modelo OSI, nos apoyaremos en la programación de sockets, la ventaja de utilizar este tipo de programación es que muchas de

las características del enlace de comunicaciones se abstraen en el concepto de manejo de archivos, ya que así es como se puede visualizar este tipo de comunicaciones.

La implementación del enlace de bajada esta fundamentada en el uso de protocolos de uso extendido, para este proyecto utilizaré la familia de protocolos TCP/IP (transfer Control Protocol/Internet Protocol), la justificación se basa en el hecho de tener a nuestra disposición la tecnología ya preparada y la facilidad de programación en diferentes lenguajes, en particular en Java y Visual Basic. Ahorrando así tiempo y enfocándonos en la solución de la propuesta.

## **5.2.- Planteamiento del Problema: Enlace de Subida.**

Tradicionalmente, las señales procedentes de una fuente de información analógica han sido transmitidas en forma analógica; sin embargo, desde hace algunos años la aplicación de las técnicas digitales a la transmisión de esas señales ha revolucionado el mundo de las telecomunicaciones. Mediante tales técnicas, las señales analógicas son transformadas en señales digitales, representadas por símbolos discretos, tal como la información procedente de las señales telegráficas o de datos. La unidad de información en este caso se conoce como bit, concepto que representa a la cantidad de información que se deriva del conocimiento de la aparición de uno de dos sucesos equiprobables. Esta situación permite un uso más eficaz de los diferentes medios de transmisión y un tratamiento homogéneo de las señales con independencia de la fuente de información original.

Dentro del canal de comunicación hay que distinguir tres partes básicas. La primera, denominada genéricamente como el transmisor, es la encargada de modificar o cambiar la naturaleza de las señales que recibe de la fuente. La segunda, el medio de transmisión, que recibe las señales convenientemente modificadas por el transmisor y las transporta hasta el punto deseado. Y por último el receptor, que recoge las señales del medio, las vuelve a transformar en su naturaleza origen y las entrega a su destino. En todos estos procesos que tienen lugar en el canal de comunicación, las señales son perturbadas por diversas distorsiones y por la aparición de otras señales espurias indeseables. Esto quiere decir que, en el extremo receptor, la señal aparecerá mezclada con otras componentes surgidas durante la transmisión y ajenas por tanto, a la información original. Todos estos efectos se conocen con el nombre genérico de ruido; y el grado de calidad de la comunicación, por lo que al ruido se refiere, dependerá de la diferencia que exista, en el extremo de destino, entre la potencia de la señal y la de dicho ruido, que la enmascara.

Para hacer que la información llegue con la suficiente calidad, dos son las actuaciones posibles. Una es conseguir que el ruido acumulado durante la comunicación no resulte excesivo; dicho en términos más correctos, conseguir que la relación entre la potencia de la señal y la potencia del ruido no sea inferior a un determinado valor límite. Otro recurso es el de aumentar la redundancia de la información que se transmite. Tal redundancia puede definirse como un exceso de información, sobre la estrictamente necesaria, para caracterizar el mensaje.

Volviendo al esquema básico de una comunicación, se ha mencionado que el transmisor modifica la naturaleza de la información antes de entregarla al medio de transmisión. ¿Por

qué es necesaria esta transformación? Hay dos razones fundamentales para ello: la primera y más importante desde el punto de vista teórico del proceso de comunicación, es la necesidad de adecuar la naturaleza de las señales a transmitir a la del medio de transmisión. En el caso de transmisión de datos a través de redes de computadoras y por la naturaleza de las señales se tienen diferentes anchos de banda, los cuales a su vez deben de ser adecuados al ancho de banda que soporte la red, con la finalidad de no perder eficiencia en la comunicación. La segunda razón, de índole económica, la constituye la conveniencia de aprovechar al máximo un determinado medio de transmisión, a base de transmitir por él no sólo una comunicación, sino varias. Se trata pues, de empaquetar varias comunicaciones, y transmitir simultáneamente el conjunto. Para realizar esta operación, denominada multicanalización, es preciso, como en el caso anterior, transformar las señales.

Resumiendo, las funciones básicas que realiza el transmisor en un canal de comunicación son: adecuar la naturaleza de las señales al medio empleado, llevar a cabo la multicanalización necesaria entre varios canales y proporcionar la “redundancia” precisa, según el sistema de comunicación de que se trate. Para desarrollar esto se utilizan diversas técnicas de digitalización. Cada técnica corresponde a la fuente que se desea analizar, para este caso debemos tener en cuenta que estas señales son: Audio y Vídeo.

En el presente planteamiento describiremos las fases por las que pasa una determinada señal para ser digitalizada y como es tratada la señal digital a través de redes de comunicación de datos.

Por ahora analizaremos como señal analógica la correspondiente a la voz ya que facilita su comprensión para el presente proyecto, pero debemos hacer mención que este proceso se puede trasladar a cualquier señal analógica que pretenda ser utilizada por sistemas de cómputo como señal digital.

### **5.2.1.- Codificación de la Voz en Señal Digital y Tratamiento Digital.**

Cuando se tiene una señal analógica se deben de conocer ciertos parámetros que nos permitan identificar de manera teórica a dicha señal, esto con la finalidad de poder modelar la señal y así diseñar los diferentes dispositivos que nos permitan capturarla y trabajar con ella. Primero debemos de conocer el ancho de banda de la señal, recordemos que el ancho de banda nos permite saber sobre que rango de frecuencias esta una señal, para así poder hacer el muestreo correspondiente. Para medir el ancho de banda de una señal se puede hacer mediante equipo de medición del laboratorio de electrónica como es el osciloscopio el cual puede ser analógico o digital, o también con equipo más complejo como un procesador digital de señales (DSP por sus siglas en inglés). El ancho de banda de la señal analógica de voz comprende el rango de los 300 a 4000 Hz. Aproximadamente. Una vez conocido el ancho de banda de la señal procedemos a obtener la velocidad de muestreo, la cual se calcula mediante el teorema de Nyquist que corresponde a la formula:

$$B = 2 * A_b$$

Donde B es la velocidad de muestreo y  $A_b$  es el ancho de banda de la señal analógica en banda base. El teorema anterior nos describe que para nosotros poder digitalizar una señal

debemos de hacer un muestreo o capturar la señal analógica al doble de la velocidad de esta, para así poder tener muestras suficientes y tener la capacidad de reconstrucción de la señal. Entonces aquí la velocidad de muestreo para la voz es de 8000 Hz. Ya que tenemos los datos anteriores se procede realizar propiamente el muestreo de la señal, el cual se realiza con equipo electrónico que trabaje a la velocidad de muestreo y realice la captura de la señal de manera electrónica teniendo así muestras de tensiones de voltaje o corriente después de que la señal ha sido muestreada.

Ya que hemos obtenido la velocidad de muestreo, se hace un filtraje de la señal para eliminar ruido que se encuentra mezclado con la señal original. Para hacer esto hacemos pasar la señal a través de un filtro pasa-bajas, diseñado de tal manera que las frecuencias de corte sean iguales al valor del ancho de banda de la señal en banda base. Después de tener el filtraje de la señal se procede a su respectivo muestreo, mediante circuitos electrónicos es como obtenemos las muestras de la señal, el resultado será la señal muestreada y filtrada lista para ser cuantizada. La cuantización es el proceso de asignación de valores de voltaje conocidos y que serán utilizados para asignar código o palabras a los valores capturados, esta operación introduce algunos errores en cuanto a los niveles ya que asigna valores fijos con valores que pueden tener una amplitud baja o alta y se puede llegar a confundir el cuantificador, ocasionando la pérdida de muestras.

Para solucionar parte de este problema se debe de tener en cuenta la forma en que se comporta la señal, así para la captura de audio se observo que las muestras están en la mayoría de los casos sobre cierto nivel de amplitud y con un comportamiento logarítmico, permitiendo así definir dos tipos de cuantizaciones logarítmicas que son los más utilizados en la actualidad, la ley A y ley  $\mu$  (mu), las cuales se utilizan en los canales de comunicación de voz a través de las centrales telefónicas. El resultado es una señal discreta filtrada y cuantizada, con niveles conocidos y lista para ser codificada.

El último proceso para tener nuestra señal digital consiste en codificar la señal cuantificada, esto se hace asignando palabras de n bits a los correspondientes niveles de cuantización, donde la cantidad de bits utilizados nos indica la precisión con la que digitalizamos la señal. El resultado es una palabra digital que puede ser interpretada por la computadora de manera más sencilla y adecuada al medio digital, que es donde se trabajará de aquí en adelante.

Hasta aquí hemos visto el proceso de digitalización de una señal analógica, el cual se debe de realizar para las señales de audio y vídeo respectivamente. El tratamiento de video es en esencia lo mismo, solo que aquí se contemplan otros aspectos y características de la naturaleza propias de la señal para obtener una imagen digitalizada.

Ahora que tenemos nuestros datos de manera digital entramos en el aspecto de la comunicación de este tipo datos a través de redes de computadoras. La comunicación de datos a través de este tipo de infraestructura ha tenido un rápido crecimiento en los últimos años, permitiendo que nuevas tecnologías emerjan y logrando una mejor integración de los servicios que una computadora puede ofrecer.

En la actualidad se puede tener comunicación de audio y video a través de redes de computadoras, pero, debemos recordar que dentro de las redes de computadoras existen diferentes tecnologías que son totalmente distintas y que por sí solas no se pueden comunicar de manera eficiente, para solucionar este problema surgieron los estándares de

comunicación de datos, como el modelo OSI de ISO y la familia de protocolos TCP/IP por citar algunos ejemplos.

Partiendo de que existen diferentes tecnologías surge la pregunta, ¿Cómo se lleva a cabo esta comunicación?, anteriormente se tenía el gran problema de que diferentes redes de computadoras no se podían comunicar entre ellas, ya que habían surgido un sin número de arquitecturas heterogéneas que no permitían la comunicación entre diferentes redes, una vez que los fabricantes notaron la falta de operabilidad de comunicación entre diferentes redes y la insatisfacción de los clientes se comenzaron a crear grupos de estandarización para conectar diferentes redes de computadoras, el resultado fue la creación de estándares estratificados o por capas, donde cada capa de un estándar se dedica a ciertas operaciones de comunicación permitiendo así una mejor interoperabilidad entre diferentes redes de computadoras. La ventaja de esta estratificación es la independencia de los datos para la comunicación, ya que la comunicación entre capas debe de mantener consistente la homogeneidad de estos datos, permitiendo así trabajar con computadoras, redes y sistemas operativos diferentes.

Aquí nos auxiliamos de la familia de protocolos TCP/IP, que son un conjunto de protocolos desarrollados en los años sesenta para la intercomunicación de computadoras a través de distintas redes, diseñados originalmente para el ejército de los Estados Unidos de Norteamérica, en particular el proyecto ARPA. Para este tipo de comunicación de datos se debe de conocer la dirección fuente y la dirección destino respectivamente la cual esta compuesta de 4 octetos (lo que se conoce como dirección IP), para poder comunicar a dos computadoras y hacer uso de las aplicaciones de manera independiente, aunque las computadoras tengan una arquitectura independiente.

Aquí podemos enviar nuestras señales digitales sin ningún problema a través de una red de computadoras y preocuparnos por la parte de aplicación, siendo transparente para nosotros la comunicación de los datos a través del medio físico. El resultado es la comunicación de datos a través de la red y sin gran dificultad de manipular en el ámbito digital.

A continuación describiremos el desarrollo del problema, el cual contiene la forma en la cual se llevaron a cabo las tareas para poder obtener el correspondiente enlace de subida para nuestro sistema de comunicación.

Para el enlace de subida se cuenta con tecnología de fácil adquisición como lo es una webcam y micrófono como dispositivos de la captura de señales analógicas para vídeo y audio respectivamente. Además de existir lenguajes de programación capaces de permitirnos interactuar con los dispositivos antes mencionados, así como realizar su correspondiente programación para la transmisión de datos a través de redes.

### **5.3. Desarrollo de la Propuesta.**

#### **5.3.1.- Desarrollo de la Propuesta para el Enlace de Bajada.**

##### **Implementación de la transmisión de los datos.**

Como ya explicamos la implementación de la transmisión será basándonos en la familia de protocolos TCP/IP gracias a este tipo de protocolos, nosotros no tenemos necesidad de

conocer más que como utilizar la capa a la que nos conectaremos, en nuestro caso nos situaremos en la capa de transporte de datos donde la primitiva de esta capa es conocida como socket. Ahora, desde el punto de vista de la capa de transporte sólo observamos la comunicación como un servidor y un cliente. Dado que la estación remota es la que nos provee de la información es el servidor, y la móvil es el cliente, esto desde el punto de vista del enlace de bajada.

La primitiva socket tiene como parámetros la dirección destino, el modo de enrutamiento, el puerto de conexión y los datos. En el Anexo A se tiene el código correspondiente para la creación del socket. Basándonos en el lenguaje de programación C bajo un sistema operativo Linux, esto con la finalidad de demostrar que esta primitiva nos ayuda en mucho la implementación de un enlace de comunicaciones bajo redes de datos.

Para la parte del cliente se muestra en el mismo anexo el código necesario para la implementación bajo el lenguaje de programación Java.

### **5.3.2.- Desarrollo de la Propuesta del Enlace de Subida.**

A continuación describiremos los procesos que se llevaron a cabo para la implantación del presente sistema telemático referente al enlace de subida, mencionaremos el enfoque global que presenta cada modulo dentro del sistema total para las actividades que permitieron el llevar a cabo el sistema, y como se fueron concatenando estos módulos con el proyecto en general.

Dentro de la implementación del proyecto tenemos a considerar las siguientes actividades:

- Instalar el sistema operativo indicado en la móvil, estación remota y estación base. Para este caso Windows XP.
- Instalar las tarjetas de red inalámbricas.
- Realizar el programa de captura de video.
- Realizar el programa de captura de audio.
- Realizar el programa de recepción en la remota.

A continuación describiremos cada una de las actividades antes mencionadas y como se llevó a cabo su implementación.

### **5.4.- Despliegue de Vídeo y Sonido en la Estación Remota.**

En esta parte la Estación Remota recibe los paquetes enviados por la Estación Base los cuales contienen los cuadros capturados por la cámara de video y las señales de audio capturadas por el micrófono los cuales están instalados en la Estación Móvil, es aquí donde la Estación Remota en función de los puertos que tenga habilitados para la recepción de señales sabrá si se trata de datos con contenido de vídeo o de audio. El despliegue de audio se realizará en las bocinas de la computadora y el despliegue de video en el monitor de la computadora.

Recordemos que para la programación de sockets es necesario habilitar los puertos sobre los cuales trabajaremos para esto nosotros definiremos sobre que puertos vamos a realizar la programación. De ahí la relativa facilidad de poder controlar el audio y video de manera independiente pero con el mismo esquema de programación para cada una de las señales.

A continuación enumeramos el esquema de programación seguido tanto para las señales de vídeo como de audio:

1. Establecer conexión de los sockets y la correspondiente comunicación.
2. Estar a la espera de los paquetes correspondientes.
3. Recibir los paquetes correspondientes ya sea de audio o vídeo.
4. Preparar los dispositivos de despliegue correspondientes, monitor o bocinas.
5. Hacer el despliegue.
6. Regresar al paso 2.

Recordando y para simplificar un poco llamemos a todas las actividades mencionadas anteriormente como: "Recepción de Vídeo y Audio".

Para concluir hagamos un esquema general de los procesos que intervienen en la estación remota:

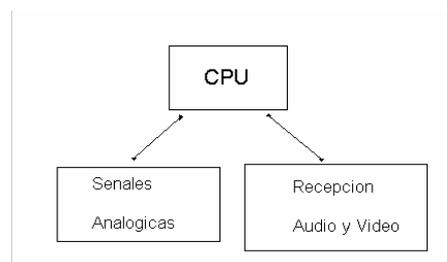


Figura 5.1. Diagrama general de los procesos a realizar en la Estación Remota.

La finalidad de ver los procesos anteriores de una manera global es la de permitirnos tener una mejor comprensión de todo el sistema. Ahora para nuestro propósito, veremos que bajo el paradigma de programación cliente-servidor, la estación remota es un cliente y el móvil es el servidor, esto nos permite observar que quien hará las peticiones de las señales referentes al Audio y Vídeo será la Estación Remota, la Estación Móvil, se encarga de capturar estas señales y proveerlas a la estación remota como lo haría un servidor de datos dentro de una red de computadoras.



## IMPLEMENTACIÓN DEL SISTEMA.

---

*“El que no posee el don de maravillarse ni de entusiasmarse más le valdría estar muerto, porque sus ojos están cerrados”.*

Ya definido los aspectos a considerar al plantear nuestro problema del enlace de comunicaciones, debemos de poner manos a la obra. En este apartado describimos la instalación de los equipos y como se llevo a cabo la programación mencionando las herramientas de las cuales nos auxiliamos para poder llevar a cabo el proyecto.

### **6.1.- Implementación del Sistema Operativo.**

El Sistema Operativo a utilizar dentro de nuestro proyecto es Windows XP, ya que nos permite una relativa facilidad en cuanto al manejo de conectividad para nuestro propósito de redes inalámbricas.

Como se había mencionado anteriormente en la descripción de los sistemas operativos Windows XP, proviene de la familia Windows NT, donde, recordando un poco, Windows NT esta diseñado para un ambiente en redes de computadoras de ahí que nos permita una facilidad en cuanto a la forma de operar del sistema operativo bajo un ambiente de comunicaciones de datos.

Es muy importante hacer mención de la programación referente a vídeo y audio, aquí utilizaremos Java, como lenguaje de programación por su facilidad de ser multiplataforma y permitirnos también una comunicación de datos a través de la red utilizando protocolos de comunicación mediante sockets además de ser un lenguaje de distribución gratuita y fácil de obtener.

Así Java no tiene la dificultad de correr bajo un ambiente en Linux o en Windows, aquí tenemos la ventaja de poder tener diferentes plataformas y un lenguaje de programación que no tiene dificultad de comunicar procesos a nivel usuario sin tener problemas con la infraestructura colocada debajo de él, haciendo de esta manera coexistir diferentes ambientes para un mismo objetivo como lo es la comunicación de datos a través de redes de computadoras.

Podemos ver que para este caso los sistemas operativos pueden coexistir de manera independiente, ya que el lenguaje de programación es el que nos provee de la comunicación entre los diferentes procesos a nivel usuario, otro punto importante es la facilidad de utilizar los protocolos de comunicaciones TCP/IP, los cuales son admitidos por el sistema operativo, permitiéndonos tener la comunicación de datos a través de la red de manera transparente para el usuario.

## 6.2.- Instalar las Tarjetas de Red Inalámbricas.

La instalación de las tarjetas de red inalámbricas es fundamental, ya que permite la comunicación entre dos computadoras sin el uso de cables, siendo de suma importancia para el control remoto de nuestro móvil.

Para este proyecto utilizamos tarjetas de red del mismo fabricante para no tener complicaciones en cuanto a interoperabilidad, pero debemos recordar que la estandarización es fundamental para este tipo de tarjetas ya que no es necesario que ambas sean del mismo fabricante para poder establecer comunicación entre ellas, por lo general el estándar de comunicación entre este tipo de tarjetas es el 802.11g, definido por el IEEE.

El tipo de tarjetas que utilizamos las fábrica la compañía 3Com, a demás el postgrado nos otorgó todas las facilidades para utilizar las tarjetas de red y punto de acceso inalámbricos las cuales nos han ayudado mucho en este proyecto.

La arquitectura general de una red inalámbrica se presenta en dos formas: modo ad-hoc o mediante un punto de acceso. El modo ad-hoc permite la comunicación entre dos computadoras de manera directa, permitiéndonos comunicar datos entre estas dos computadoras. El punto de acceso nos da la posibilidad de tener conectadas varias computadoras entre sí, al cumplir la función similar a la de un concentrador.

La instalación de la tarjeta de red inalámbrica en el móvil se realizo mediante el uso del disco de instalación que entrega el fabricante al momento de la compra y que tiene los archivos controladores del hardware.

El siguiente diagrama ejemplifica los dos tipos de arquitectura básica de una red inalámbrica:

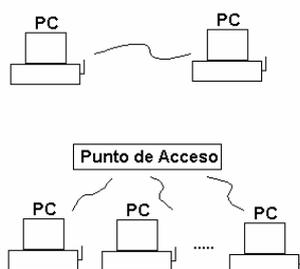


Figura 6.1. Arquitecturas de red inalámbrica, arriba modo ad-hoc, abajo usando punto de acceso.

Aquí retomamos el punto referente a la forma en como esta conformada nuestra arquitectura de red, si fuera necesario se tendría que crear un programa en la estación base que serviría de puente para la interconexión entre la estación remota y el móvil y trabajar en modo ad-hoc, o se puede aprovechar un punto de acceso para la comunicación entre la estación remota y el móvil de manera directa, donde el punto de acceso cumple la función de transformar las direcciones IP entre las tarjetas de red alámbricas y las tarjetas de red inalámbricas, permitiéndonos ver una red homogénea en donde la parte física es transparente para el usuario y la comunicación se establece de manera puntual, esto es Remota-Móvil.

### **6.3.- Diseño e implementación de Software para la transmisión de Audio y Vídeo.**

Para la captura de audio y vídeo se utilizó el lenguaje de programación Java, el cual tiene las siguientes características: multiplataforma, gratuito y orientado a objetos. Describamos un poco más sus características para poder comprender este lenguaje de programación.

**Multiplataforma.** Una de las principales características de Java es que es independiente del sistema operativo en el cual se trabaje, podemos tener aplicaciones que se comuniquen a través de una red de datos ya sea en maquinas con sistema operativo Windows XP y Linux. La forma en que JAVA implementa esta característica es porque este lenguaje de programación esta orientado a ser interprete apoyándose en lo que JAVA conoce como JVM (Máquina Virtual de Java ó Java Virtual Machine), la cual se encarga de hacer la traducción del lenguaje Java a lenguaje máquina.

**Gratuito.** Java es desarrollado por SUN Microsystems, la finalidad de la gratuidad es el poder implementarse en la mayor cantidad de sistemas posibles, con un lenguaje de programación altamente potente y de fácil acceso, que cada vez esta ganando mayores adeptos dentro de la comunidad informática mundial.

**Orientado a Objetos.** Java es un lenguaje de programación de cuarta generación, orientado a objetos lo que facilita la programación de las aplicaciones, permitiendo a los desarrolladores preocuparse por las aplicaciones y dejando atrás los problemas de conectividad y de bajo nivel, del cual se encarga la JVM.

Para la creación de los programas que se encargan de capturar audio y vídeo utilizamos Java como lenguaje de programación base y una API de programación que es una extensión de Java, la cual es JMF (Java Media Framework), la cual provee del soporte de multimedia como transmisión de audio y video a través de redes de datos. A continuación describiremos un poco más a detalle esta API de programación.

Trabajando con Medios basados en tiempo.

Cualquier dato que varia con respecto al tiempo puede ser caracterizado como un medio basado en tiempo. Clips de audio, secuencias MIDI, secuencias de vídeo y animaciones, son formas comunes de medios basados en tiempo. Estos datos pueden ser obtenidos de una variedad de fuentes, como pueden ser archivos locales o a través de la red, cámaras, micrófonos y grabaciones en vivo.

A continuación describiremos las principales características de los medios basados en tiempo y explicaremos el uso de esos medios en términos del proceso fundamental para llevar a cabo su modelado.

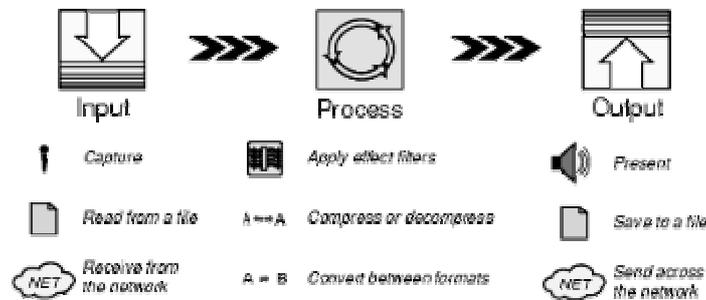


Figura 6.2. Modelo para el procesamiento de medios.

### Flujo de Medios.

Una de las principales características de los medios basados en tiempo es que requieren recursos en tiempo y procesador. Una vez que el flujo de datos comienza, existe un estricto límite en tiempo que debe ser conocido, ambos en términos de recepción y presentación de los datos. Por esta razón, los medios basados en tiempo son nombrados como flujo de medios, el cual debe ser distribuido en cierto tiempo como un flujo y de igual manera debe de ser recibido y procesado dentro de un marco de tiempo particular para producir resultados aceptables.

Por ejemplo, cuando una película es reproducida, y si los medios no pueden ser reproducidos rápidamente, habrá pausas molestas y retardos. De igual manera en la recepción, si esta no se da de manera fluida caemos en el problema de tener retardos o cuadros faltantes en la reproducción lo que ocasiona la pérdida de consistencia en la reproducción de los medios.

### Tipo de Contenido.

Es el formato en el cual los datos son almacenados, es necesario tener un formato puesto que es el que nos dice como vamos a realizar el tratamiento de los datos. Quick Time, MPEG y WAV son algunos ejemplos de formatos que podemos ver en la actualidad. Para nuestro caso tipo de contenido es sinónimo de tipo de archivo, el cual es usado porque los medios son adquiridos de otras fuentes diferentes a la de un archivo.

### Flujo de Medio.

Un flujo de medio es un flujo de datos obtenidos de un archivo, obtenidos de la red o capturados de una cámara o micrófono. Los flujos de medios por lo regular contienen canales de datos llamados tracks. Por ejemplo, un archivo Quicktime puede contener tanto audio track y vídeo track. Los flujos de medio que tienen más de un track se consideran como flujos de medio complejos o multiplexados. Demultiplexar es el proceso de obtener cada track de manera individual de un flujo de medio complejo.

El tipo de track define el tipo de datos contenidos, como puede ser audio o vídeo. El formato define como es estructurado el dato de un track. Un flujo de medios puede ser identificado por su ubicación y protocolo usado para acceder. Por ejemplo un URL puede ser usado para describir la ubicación de un archivo AVI procedente un sistema de archivos remoto. Si el archivo es local, este puede ser accesado a través del protocolo FILE de Java. Un objeto “media locator” provee la manera de identificar la ubicación de la fuente o flujo de medios.

Para el presente proyecto Java define la clase RTP (Real-Time Transport Protocol) dentro de JMF. A continuación describiremos esta API.

### **Real-Time Transport Protocol.**

Para enviar o recibir medios en vivo de manera unicast, multicast o broadcast, así como el conducir una video conferencia, debemos tener la capacidad de recibir y enviar flujos de medios en tiempo real. Los siguientes párrafos describen los flujos de medios de manera profunda además de describir el RTP de JMF, usado para transmitir y recibir medios a través de la red. El termino flujo de medios es comúnmente usado para hacer referencia tanto a las técnicas de distribuir medios sobre una red de computadoras como para la reproducción de los medios. La transmisión de medios a través de la red en tiempo real requiere de un throughput de la red alto. Parece ser que es más fácil compensar la pérdida de datos que tener un retardo mayor en la espera de los datos.

Comencemos analizando un poco los protocolos de la familia TCP/IP (en particular TCP y UDP), http y FTP son protocolos basados en el TCP (Transfer Control Protocol), donde TCP es un protocolo de la capa de transporte diseñado para comunicaciones eficientes en anchos de banda pequeños, y cuando un paquete es dañado este se retransmite, el garantizar la transferencia de datos puede hacer que la red se congestione y exista un cuello de botella que no nos permita transferir más datos sobre la red. Por esta razón se hace presente el uso de otros protocolos que no se enfoquen tanto a la conexión. Uno de los protocolos más conocidos y usados para este propósito es el protocolo UDP (User Datagram Protocol), el cual es orientado a no conexión, por lo que no garantiza que cada paquete llegue a su destino en el mismo orden que fue enviado, así que es el transmisor quien se encarga de enviar paquetes y de acomodarlos en el orden que corresponde a cada paquete.

Al igual que TCP, UDP es un protocolo de la capa de transporte debajo del cual se encuentran otros protocolos para aplicaciones específicas. El estándar en Internet para transportar datos en tiempo real como audio y vídeo es el Real-Time Transport Protocol. RTP esta definido en el RFC 1889 del IETF (Internet Engineering Task Force), el cual es un producto del grupo de trabajo AVT (Audio Video Transport).

RTP provee una comunicación punto a punto a través de la red para la transmisión de servicios basados en tiempo real. RTP es un protocolo que se encuentra en las capas de red y transporte, pero que la mayoría de las veces se cree que trabaja sobre UDP. El siguiente diagrama explica la arquitectura básica de RTP.

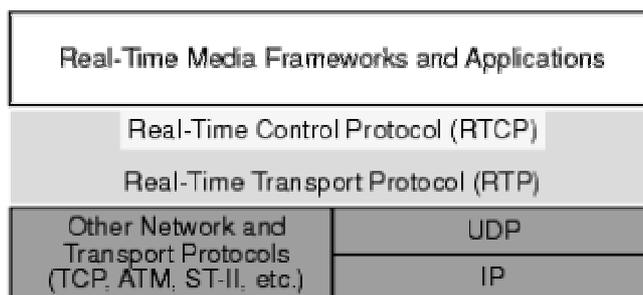


Figura 6.3. Arquitectura de RTP.

RTP puede ser usado para programar en la red mediante unicast, multicast y broadcast. Multicast es más eficiente para el tratamiento de medios sobre la red, como video conferencias, puesto que IP (Internet Protocol) soporta multicasting.

Los servicios que RTP proporciona nos permiten identificar el tipo de datos que van a ser transmitidos, determinar el orden de envío de los paquetes que deben de ser presentados y sincronizar los flujos de medios de las diferentes fuentes. RTP no garantiza que los paquetes van a llegar en el orden en el cual fueron enviados, inclusive no garantiza que los paquetes lleguen, aquí el receptor es el encargado de asegurarse que el orden de los paquetes sea el correcto usando la información contenida en el encabezado del paquete.

Puesto que RTP no provee de las garantías suficientes en cuanto a tiempo de respuesta o servicios, se tiene un protocolo de ayuda el cual es el RTCP (Real-Time Transport Control Protocol) que permite al programador monitorear la calidad de los datos. RTCP también provee los mecanismos para el control y la identificación para las transmisiones RTP.

### Arquitectura RTP.

Una sesión RTP es una asociación entre un conjunto de aplicaciones que se comunican con RTP. Una sesión es identificada por la dirección de la red y los puertos de comunicación. Donde un puerto es usado para los datos del medio y otro puerto es usado para el control de los datos (RTCP).

Un participante es un host, o una máquina así como un usuario. La participación en la sesión puede ser de tres tipos: recepción de datos (participación pasiva), o transmisión de datos (participación activa), o ambas.

Cada tipo de medio es transmitido en una diferente sesión. Por ejemplo, si queremos transmitir audio y video para una videoconferencia, una sesión debe de ser para la transmisión de audio y otra para la transmisión de vídeo. Esto permite a los participantes que tipo de medio quieren recibir, ya que si algún participante cuenta con un ancho de banda pequeño, la mejor solución sería la de recibir sólo el audio.

### Los paquetes de Medios.

Los datos de un medio son transmitidos a través de la red como una serie de paquetes. Esta serie de paquetes provenientes de una fuente en particular es conocida como RTP stream. Cada paquete RTP se compone de dos partes, un encabezado y los datos (como sucede con

cualquier otro protocolo dentro de la familia TCP/IP). A continuación mostramos la forma en la que esta construido un paquete RPT:

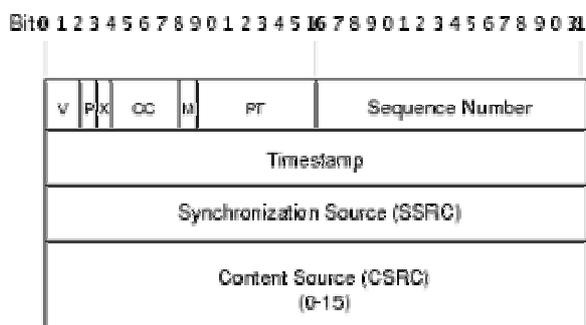


Figura 6.4. Paquete RTP.

A continuación describiremos cada uno de los campos de que se compone este paquete en el encabezado.

- Número de Versión de RTP (V): 2 bits. La versión definida en las especificaciones actuales es la 2.
- Padding (P): 1 bit. Si el bit de padding esta habilitado, existen uno o más bytes al final del paquete que no son de datos, el ultimo byte del paquete indica el número de bytes del padding. El padding es usado por algunos algoritmos de encriptación.
- Extensión (X): 1 bit. Si este bit esta habilitado el encabezado de mezclado es seguido de un encabezado de extensión. Este mecanismo de extensión permite agregar información al encabezado RTP.
- Contador CSRC (CC): 4 bits. El número de CSRC identifica que el encabezado de mezclado es el siguiente. Si el contador CSRC es cero, la fuente de sincronización es el origen de los datos contenidos.
- Marcador (M): 1 bit. Un bit de marcado definido para un perfil de medio particular.
- Tipo de Datos (PT): 7 bits. Es el indice dentro de un perfil de medio contenido en una tabla referente al formato de los datos. Los mapeos para los datos de audio y vídeo están especificados en el RFC 1890.
- Número de Secuencia: 16 bits. Un número de paquete unico que identifica la posición del paquete en el que se encuentra, el número de paquete es incrementado a uno cada vez que es enviado un paquete.
- Marca de Tiempo (Timestamp): 32 bits. Refleja el instante del muestreo del primer byte en los datos. Los paquetes pueden tener la misma marca de tiempo si estos son lógicamente generados en el mismo tiempo, por ejemplo si son parte del mismo cuadro de vídeo.
- SSRC: 32 bits. Sirve para identificar la fuente de sincronización. Si el CSRC es cero, los datos del paquete son los datos de sincronización, de lo contrario el SSRC identifica al mezclador.
- CSRC: 32 bits. Identifica la contribución de las fuentes para los datos. el número de las fuentes que contribuyen es indicado por el campo contador de CSRC; aquí puede haber más de 16 fuentes de contribución, entonces los bytes de datos contienen los datos originales mezclados provenientes de esas fuentes.

### **Paquetes de Control.**

Además de los datos provenientes de los medios para una sesión, los paquetes de datos de control (RTCP) son enviados periódicamente a todos los participantes de la sesión.

Los paquetes RTCP pueden contener información acerca de la calidad del servicio para los participantes de la sesión, información acerca de la fuente que transmite los datos y las estadísticas pertenecientes a los datos que han sido transmitidos.

Existen diferentes tipos de paquetes RTCP:

- Reporte de Transmisor.
- Reporte de Receptor.
- Descripción de la fuente.
- Fin (Bye).
- Especificación de la aplicación.

Los paquetes RTCP se pueden almacenar para un posterior análisis, estos son enviados como un conjunto de paquetes que contienen al menos dos paquetes, un paquete de reporte y una descripción de la fuente, por ejemplo.

Todos los participantes de una sesión envían paquetes de RTCP. Un participante que ha enviado solicita un reporte de transmisión. El paquete de reporte de transmisión contiene el número total de paquetes y bytes enviados así como información que puede ser usada para sincronizar los flujos de medios de diferentes sesiones.

Los participantes de la sesión suelen enviar paquetes de recepción a todas las fuentes de las cuales están obteniendo datos. Un paquete de reporte de recepción (RR) contiene información acerca del número de paquetes perdidos, el número más alto de secuencia recibido y el timestamp, que puede ser usado para estimar el tiempo que dura el enlace de comunicación entre un receptor y transmisor.

El primer paquete a través de RTCP debe de ser un paquete de reporte, aun si ningún dato ha sido enviado o recibido, en donde el paquete que se envía está vacío.

### **Aplicaciones RTP.**

Las aplicaciones RTP por lo general son divididas en aquellas que deben de poder recibir datos a través de la red (RTP Clients) y aquellas que debe de poder transmitir datos a través de la red (RTP Servers). Algunas aplicaciones hacen las dos. Por ejemplo, aplicaciones de conferencia capturan y transmiten datos al mismo tiempo que reciben datos de la red.

Recepción de Flujo de Medios de la Red.

El tener la capacidad de recibir datos a través de RTP se hace necesaria para el uso de diferentes aplicaciones, por ejemplo:

- Aplicaciones de conferencia necesitan tener la capacidad de recibir flujos de datos por medio de una sesión RTP así como desplegarlos en la consola.
- Una máquina contestadora telefónica necesita tener la capacidad de recibir datos de un archivo que contenga un flujo de medios almacenado.

- Una aplicación que tenga que grabar algún tipo de conversación o conferencia debe de tener la capacidad de recibir un flujo de medios a través de una sesión RTP.

Transmisión de flujos de medios a través de la Red.

Aplicaciones del servidor RTP capturan o almacenan flujos de medios a través de la red. Por ejemplo, en una aplicación de conferencia, un flujo de medios puede ser la captura de imágenes por medio de una cámara de vídeo y ser enviada a una o más sesiones RTP. Los flujos de medios deben de ser codificados en múltiples formatos y ser enviados en diferentes sesiones de RTP para hacer la conferencia con receptores heterogéneos. Una conferencia multiparticipante puede ser implementada sin un direccionamiento IP multicast usando solamente sesiones RTP.

Para concluir nuestra explicación debemos de hacer mención de que las clases de JMF que se basan en RTP realizan de manera automática la detección del Hardware correspondiente a la tarjeta de vídeo y la tarjeta de audio. La PC al contener este tipo de tarjetas contiene implementadas las etapas antes descritas para la digitalización de una señal analógica, así la tarjeta de sonido tiene integrados los filtros, cuantificadores y codificadores respectivos al tratamiento de sonido. De igual forma la webcam contiene los chips necesarios para la digitalización de la señal, he aquí las ventajas de la tecnología y como podemos aprovechar la facilidad de comunicación con los dispositivos periféricos que interactúan con nuestro sistema, entonces el problema a resolver es referente a la programación de estos dispositivos, permitiendo al programador concentrarse en la solución lógica y no física.

Para una mayor referencia en el Anexo B, se tiene el código utilizado para llevar a cabo la implementación de la programación.



## CONCLUSIONES.

---

*“Dar ejemplo no es la principal manera de influir sobre los demás; es la única”.*

La implementación de soluciones utilizando nuevas tecnologías esta teniendo un crecimiento acelerado dentro de la industria en la actualidad, es importante recalcar que la formación y el desarrollo dentro de los centros de investigación debe tener una mayor vinculación con la industria si queremos que el desarrollo tanto regional como global continúe.

La experiencia de este trabajo deja claro que las soluciones en cuanto a desarrollo e implementación tecnológica pueden permitirnos economizar y no con eso comprometer la calidad de la solución. Una de las principales ventajas es la utilización de tecnologías de libre acceso, con esto obtenemos herramientas estandarizadas que nos permiten trabajar con parámetros internacionales y sin error a trabajar basándonos en la interconexión de sistemas abiertos, además es importante la retroalimentación tanto nacional como internacional en el desarrollo de las soluciones ya que existe gente que tiene en mente la solución de problemas con profundas similitudes.

Cabe concluir que este trabajo ha rendido los frutos planteados desde un principio, mi finalidad última en este proyecto es la de dar la pauta para que más personas se interesen en el desarrollo y optimización de módulos en particular del sistema de comunicaciones aquí desarrollado.



## ANEXO.

---

Para el programa cliente que estará corriendo en la móvil esta este programa en Java.

```
import java.io.*;
import java.lang.*;
import java.net.*;
import descriptar;

public class A {
    public int recibe_dato_enscriptado(InputStream entrada){
        int c,i,dato_e=0;
        try {
            for(i=0;i<2;i++) {
                c=entrada.read();
                if(i==0)
                    dato_e=c<<8;
                else
                    dato_e=dato_e+c; } }
            catch(IOException e) {
                e.printStackTrace(); }
            return dato_e; }
    public void runClient() {
        Socket client;
        InputStream input;
        try {
            client=new Socket("148.204.221.245",6500);
            System.out.println("Socket creado\n");
            input=client.getInputStream();
            System.out.println("Flujo de entrada\n");
            String s = new String("\n");
            descriptar descriptador=new descriptar();
            System.out.println(descriptador.rundescriptar
```

```
        (recibe_dato_encrypted(input,255));
        System.out.println("\n");
        client.close();
        System.out.println("Socket cerrado\n"); }
    catch(IOException e) {
        e.printStackTrace(); }}
public static void main(String args[]) {
    A c=new A();
    c.runClient(); } }
```

Como podemos ver el puerto que se esta utilizando es el 6500 y la estación remota tiene la dirección IP 148.204.221.225.

## ANEXO.

---

Para la parte de Transmisión de Audio y Vídeo Tenemos:

```
// Clases necesarias para el programa.
import java.awt.*;
import java.io.*;
import java.net.InetAddress;
import javax.media.*;
import javax.media.protocol.*;
import javax.media.protocol.DataSource;
import javax.media.format.*;
import javax.media.control.TrackControl;
import javax.media.control.QualityControl;
import javax.media.rtp.*;
import javax.media.rtp.rtcp.*;
import com.sun.media.rtp.*;

public class transmitir_medios {
    // Colocar la clase MediaLocator
    // Puede ser un archivo, un http o una fuente de captura
    private MediaLocator locator;
    private String ipAddress;
    private int portBase;
    private Processor processor = null;
    private RTPManager rtpMgrs[];
    private DataSource dataOutput = null;
    public transmitir_medios(MediaLocator locator,
                             String ipAddress,
                             String pb,
                             Format format) {
        this.locator = locator;
        this.ipAddress = ipAddress;
        Integer integer = Integer.valueOf(pb);
        if (integer != null)
            this.portBase = integer.intValue();
    }
}
/**
 * Comienza la transmision. Devuelve null en exito.
```

```
* De lo contrario devuelve una cadena con el error correspondiente.
*/
public synchronized String start() {
    String result;
    // Crea un procesador para el correspondiente media locator
    // y programa la salida JPEG/RTP
    result = createProcessor();
    if (result != null)
        return result;
    // Crea una session RTP para transmitir la salida al
    // procesador especificado por la dirección IP y puerto.
    result = createTransmitter();
    if (result != null) {
        processor.close();
        processor = null;
        return result;
    }
    // Comienza la transmission
    processor.start();
    return null;
}
/**
 * Detenemos la transmission si esta ya habia comenzado
 */
public void stop() {
    synchronized (this) {
        if (processor != null) {
            processor.stop();
            processor.close();
            processor = null;
            for (int i = 0; i < rtpMgrs.length; i++) {
                rtpMgrs[i].removeTargets( "Sesion finalizada.");
                rtpMgrs[i].dispose();} } }
}
private String createProcessor() {
    if (locator == null)
        return "Locator is null";
    DataSource ds;
    DataSource clone;
    try {
        ds = javax.media.Manager.createDataSource(locator);
    } catch (Exception e) {
        return "No se pudo crear DataSource";
    }
    // tratamos de crear al procesador para capturar el media locator
    try {
        processor = javax.media.Manager.createProcessor(ds);
    } catch (NoProcessorException npe) {
        return "Couldn't create processor";
    } catch (IOException ioe) {
        return "IOException creando procesador";
    }
    // Esperamos para la configuración
    boolean result = waitForState(processor, Processor.Configured);
    if (result == false)
        return "Couldn't configure processor";
    // Obtenemos los tracks del procesador
    TrackControl [] tracks = processor.getTrackControls();
    // Vemos si al menos tenemos un track.
```

```

if (tracks == null || tracks.length < 1)
    return "Couldn't find tracks in processor";
// Colocamos la salida para el descriptor tipo RAW_RTP
// Esto limitara los formatos soportados por
// Track.getSupportedFormats para permitir formatos RTP validos.
ContentDescriptor cd = new ContentDescriptor(ContentDescriptor.RAW_RTP);
processor.setContentDescriptor(cd);
Format supported[];
Format chosen;
boolean atLeastOneTrack = false;
// Programamos los tracks.
for (int i = 0; i < tracks.length; i++) {
    Format format = tracks[i].getFormat();
    if (tracks[i].isEnabled()) {
        supported = tracks[i].getSupportedFormats();
        // Hemos colocado la salida RAW_RTP.
        // Por lo tanto todos los formatos son soportados por RTP.
        // Ahora tomaremos el primero.
        if (supported.length > 0) {
            if (supported[0] instanceof VideoFormat) {
                // Para formatos de video, se debe checar doble
                // no todos los tamaños pueden trabajar.
                chosen = checkForVideoSizes(tracks[i].getFormat(),
                    supported[0]);
            } else
                chosen = supported[0];
            tracks[i].setFormat(chosen);
            System.err.println("Track " + i + " esta listo para transmitir como:");
            System.err.println(" " + chosen);
            atLeastOneTrack = true;
        } else
            tracks[i].setEnabled(false);
    } else
        tracks[i].setEnabled(false);}
if (!atLeastOneTrack)
    return "Couldn't set any of the tracks to a valid RTP format";
// Realiza el procesador. Esto creara internamente un flujo
// grafico para intentar crear una salida del tipo JPEG/RTP
// en marcos de audio.
result = waitForState(processor, Controller.Realized);
if (result == false)
    return "Couldn't realize processor";
// Colocamos la calidad JPEG .5.
setJPEGQuality(processor, 0.5f);
// obtenemos la fuente de salida del procesador.
dataOutput = processor.getDataOutput();
return null;
}
/**
 * Usamos la API RTPManager para crear una session de cada medio
 * en el procesador.
 */
private String createTransmitter() {
    // Debemos de checar el tipo.
    PushBufferDataSource pbds = (PushBufferDataSource)dataOutput;
    PushBufferStream pbss[] = pbds.getStreams();

```

```

rtpMgrs = new RTPManager[pbss.length];
SessionAddress localAddr, destAddr;
InetAddress ipAddr;
SendStream sendStream;
int port;
SourceDescription srcDesList[];
for (int i = 0; i < pbss.length; i++) {
    try {
        rtpMgrs[i] = RTPManager.newInstance();
        // La direccion de la session local sera creada en
        // el mismo Puerto. Esto es necesario para clase
        // en conjunto con JMStudio.
        // JMStudio asume - in a session unicast - que el
        // transmisor transmite por el mismo Puerto que recibe
        // y que envia RTCP de regreso por el puerto
        // que transmite el host.
        port = portBase + 2*i;
        ipAddr = InetAddress.getByAddress(ipAddress);
        localAddr = new SessionAddress( InetAddress.getLocalHost(),
                                       port);
        destAddr = new SessionAddress( ipAddr, port);
        rtpMgrs[i].initialize( localAddr);
        rtpMgrs[i].addTarget( destAddr);
        System.err.println( "Creada session RTP: " + ipAddress + " " + port + " " + i);

        sendStream = rtpMgrs[i].createSendStream(dataOutput, i);
        sendStream.start();
    } catch (Exception e) {
        return e.getMessage();}
    return null;}
/**
 * Para JPEG y H263, nosotros sabemos que trabajan para tamaños
 * particulares. Asi que debemos ver que trabajen a
 * los tamaños correctos.
 */
Format checkForVideoSizes(Format original, Format supported) {
    int width, height;
    Dimension size = ((VideoFormat)original).getSize();
    Format jpegFmt = new Format(VideoFormat.JPEG_RTP);
    Format h263Fmt = new Format(VideoFormat.H263_RTP);
    if (supported.matches(jpegFmt)) {
        // Para JPEG, nos aseguramos que el tamaño sea divisible por 8.
        width = (size.width % 8 == 0 ? size.width :
                (int)(size.width / 8) * 8);
        height = (size.height % 8 == 0 ? size.height :
                (int)(size.height / 8) * 8);
    } else if (supported.matches(h263Fmt)) {
        // Para H.263, solo se soportan algunos tamaños.
        if (size.width < 128) {
            width = 128;
            height = 96;
        } else if (size.width < 176) {
            width = 176;
            height = 144;
        } else {
            width = 352;

```

```

        height = 288; }
    } else {
        // No se conoce este formato en particular. Solo lo
        // dejamos aparte.
        return supported; }
    return (new VideoFormat(null,
                            new Dimension(width, height),
                            Format.NOT_SPECIFIED,
                            null,
                            Format.NOT_SPECIFIED)).intersects(supported);}

/**
 * Asignando la calidad adecuada para el codificador JPEG.
 * 0.5 es aceptable.
 */
void setJPEGQuality(Player p, float val) {
    Control cs[] = p.getControls();
    QualityControl qc = null;
    VideoFormat jpegFmt = new VideoFormat(VideoFormat.JPEG);
    // Hacemos un ciclo en los controles para encontrar la calidad
    // del codificador JPEG.
    for (int i = 0; i < cs.length; i++) {
        if (cs[i] instanceof QualityControl &&
            cs[i] instanceof Owned) {
            Object owner = ((Owned)cs[i]).getOwner();
            // Vemos si el propietario es un Codec.
            // Entonces checamos la salida.
            if (owner instanceof Codec) {
                Format fmts[] = ((Codec)owner).getSupportedOutputFormats(null);
                for (int j = 0; j < fmts.length; j++) {
                    if (fmts[j].matches(jpegFmt)) {
                        qc = (QualityControl)cs[i];
                        qc.setQuality(val);
                        System.err.println("- Asignando calidad " +
                            val + " on " + qc);
                        break; } }
                if (qc != null)
                    break; } }
    }

    /**
     * Metodos convenientes para obtener los cambios del procesador.
     */
    private Integer stateLock = new Integer(0);
    private boolean failed = false;
    Integer getStateLock() {
        return stateLock; }
    void setFailed() {
        failed = true; }
    private synchronized boolean waitForState(Processor p, int state) {
        p.addControllerListener(new StateListener());
        failed = false;
        // Llamamos al metodo requerido por el procesador.
        if (state == Processor.Configured) {
            p.configure();
        } else if (state == Processor.Realized) {
            p.realize(); }
        // Esperamos que un evento confirme
        // el exito del metodo, un evento de fallo.

```

```

// Ver la clase interna StateListener
while (p.getState() < state && !failed) {
    synchronized (getStateLock()) {
        try {
            getStateLock().wait();
        } catch (InterruptedException ie) {
            return false;}}
    if (failed)
        return false;
    else
        return true; }
/*****
* Causas internas.
*****/
class StateListener implements ControllerListener {
    public void controllerUpdate(ControllerEvent ce) {
        // Si hay un error en la configuración o
        // éxito, el procesador sera cerrado.
        if (ce instanceof ControllerClosedEvent)
            setFailed();
        // Todos los eventos de control, envian una notificación
        // para esperar el hilo en el metodo waitForState.
        if (ce instanceof ControllerEvent) {
            synchronized (getStateLock()) {
                getStateLock().notifyAll();}}}}
/*****
* Función principal de uso
*****/
public static void transmite_audio_video(String [] args) {
    // Son necesarios tres parametros para comenzar a transmitir.
    // Por ejemplo,
    // transmite_audio_video(file:/C:/media/test.mov,
    // 129.130.131.132,42050);
    if (args.length < 3) {
        prUsage();
        Format fmt = null;
        int i = 0;
        // Creamos un objeto de transmission de audio.
        Transmitir_medios at = new transmitir_medios(new MediaLocator(args[i]), args[i+1],
args[i+2], fmt);
        // Comenzamos la transmisión
        String result = at.start();
        // result sera diferente de null si hay error. El valor regresado
        // es una cadena que describe el error. Se imprime.
        if (result != null) {
            System.err.println("Error : " + result);
            System.exit(0);}
        System.err.println("Comienza la transmisión...");
        // transmitimos indefinidamente hasta que cambia la variable
        // esto lo hacemos ya que el procesador debe de transmitir
        // audio y video durante todo el proceso de movimiento
        // del móvil.
        try {
            while(1){
                Thread.currentThread();
            } catch (InterruptedException ie) { }

```

```

        // Detenemos la transmisión
        at.stop();
        System.err.println("...transmission ended.");
        System.exit(0);}
    static void prUsage() {
        System.err.println("Uso: transmitir_medios <sourceURL> <destIP> <destPortBase>");
        System.err.println("  <sourceURL>: ingrese URL o nombre de archivo");
        System.err.println("  <destIP>: multicast, broadcast o unicast IP de transmisión");
        System.err.println("  <destPortBase>: Puerto de red para la transmisión.");
        System.err.println("                    El primer track usara el Puerto de destino base.");
        System.err.println("                    El proximo track usara el puerto base + 2 y así
sucesivamente.\n");
        System.exit(0);} }

```

Ahora presentaremos el código correspondiente para la recepción de los datos de audio y vídeo.

```

// Clases a utilizar.
import java.io.*;
import java.awt.*;
import java.net.*;
import java.awt.event.*;
import java.util.Vector;
import javax.media.*;
import javax.media.rtp.*;
import javax.media.rtp.event.*;
import javax.media.rtp.rtcp.*;
import javax.media.protocol.*;
import javax.media.protocol.DataSource;
import javax.media.format.AudioFormat;
import javax.media.format.VideoFormat;
import javax.media.Format;
import javax.media.format.FormatChangeEvent;
import javax.media.control.BufferControl;
/**
 * recibir_medios recibe transmisiones RTP usando la nueva API RTP.
 */
public class recibir_medios implements ReceiveStreamListener, SessionListener,
    ControllerListener{
    String sessions[] = null;
    RTPManager mgrs[] = null;
    Vector playerWindows = null;
    boolean dataReceived = false;
    Object dataSync = new Object();
    public AVReceive2(String sessions[]) {
        this.sessions = sessions; }
    protected boolean initialize() {
        try {
            InetAddress ipAddr;
            SessionAddress localAddr = new SessionAddress();
            SessionAddress destAddr;
            mgrs = new RTPManager[sessions.length];
            playerWindows = new Vector();
            SessionLabel session;
            // Abrir sesiones RTP.
            for (int i = 0; i < sessions.length; i++) {
                // Probar las direcciones de sesión.
                try {

```

```

        session = new SessionLabel(sessions[i]);
    } catch (IllegalArgumentException e) {
        System.err.println("error para levantar sesion: " + sessions[i]);
        return false;
    }
    System.err.println(" - Sesion abierta RTP por: IP: " + session.addr + " puerto: " +
session.port + " ttl: " + session.ttl);
    mgrs[i] = (RTPManager) RTPManager.newInstance();
    mgrs[i].addSessionListener(this);
    mgrs[i].addReceiveStreamListener(this);
    ipAddr = InetAddress.getByName(session.addr);
    if( ipAddr.isMulticastAddress() ) {
        // la dirección origen destino es identica:
        localAddr= new SessionAddress( ipAddr,
                                     session.port,
                                     session.ttl);
        destAddr = new SessionAddress( ipAddr,
                                     session.port,
                                     session.ttl);
    } else {
        localAddr= new SessionAddress( InetAddress.getLocalHost(),
                                     session.port);
        destAddr = new SessionAddress( ipAddr, session.port);
    }
    mgrs[i].initialize( localAddr);
    // Se pueden probar otros tamaños de buffer para ver
    // si se puede tener mayor suavidad.
    BufferControl bc =
(BufferControl)mgrs[i].getControl("javax.media.control.BufferControl");
    if (bc != null)
        bc.setBufferLength(350);
    mgrs[i].addTarget(destAddr);}
} catch (Exception e){
    System.err.println("No se puede crear sesion RTP: " + e.getMessage());
    return false;}
// Esperar la recepción de datos.
long then = System.currentTimeMillis();
long waitingPeriod = 30000; // Esperar un maximo de 30 segs.
try{
    synchronized (dataSync) {
        while (!dataReceived &&
              System.currentTimeMillis() - then < waitingPeriod) {
            if (!dataReceived)
                System.err.println(" - Esperando datos RTP...");
            dataSync.wait(1000);}}
    catch (Exception e) {}
    if (!dataReceived) {
        System.err.println("No RTP recibido.");
        close();
        return false;}
    return true;}
public boolean isDone() {
    return playerWindows.size() == 0;}
/**
 * Cerrar las sesiones y los administradores.
 */
protected void close() {
    for (int i = 0; i < playerWindows.size(); i++) {

```

```

        try {
            ((PlayerWindow)playerWindows.elementAt(i)).close();
        } catch (Exception e) {}
    }
    playerWindows.removeAllElements();
    // Cerrar la sesión RTP.
    for (int i = 0; i < mgrs.length; i++) {
        if (mgrs[i] != null) {
            mgrs[i].removeTargets( "Cerrar session recibir_medios");
            mgrs[i].dispose();
            mgrs[i] = null; } }
    PlayerWindow find(Player p) {
        for (int i = 0; i < playerWindows.size(); i++) {
            PlayerWindow pw = (PlayerWindow)playerWindows.elementAt(i);
            if (pw.player == p)
                return pw; }
        return null; }
    PlayerWindow find(ReceiveStream strm) {
        for (int i = 0; i < playerWindows.size(); i++) {
            PlayerWindow pw = (PlayerWindow)playerWindows.elementAt(i);
            if (pw.stream == strm)
                return pw; }
        return null; }
    /**
     * Escuchar una sesión.
     */
    public synchronized void update(SessionEvent evt) {
        if (evt instanceof NewParticipantEvent) {
            Participant p = ((NewParticipantEvent)evt).getParticipant();
            System.err.println(" - Un nuevo participante: " + p.getCNAME()); } }
    /**
     * Recibir un flujo de datos.
     */
    public synchronized void update( ReceiveStreamEvent evt) {
        RTPManager mgr = (RTPManager)evt.getSource();
        Participant participant = evt.getParticipant(); // Puede ser null.
        ReceiveStream stream = evt.getReceiveStream(); // Puede ser null.
        if (evt instanceof RemotePayloadChangeEvent) {
            System.err.println(" - Received an RTP PayloadChangeEvent.");
            System.err.println("Perdon, No se puede cambiar los datos.");
            System.exit(0); }
        else if (evt instanceof NewReceiveStreamEvent) {
            try {
                stream = ((NewReceiveStreamEvent)evt).getReceiveStream();
                DataSource ds = stream.getDataSource();
                // Encontrar los formatos.
                RTPControl ctl = (RTPControl)ds.getControl("javax.media.rtp.RTPControl");
                if (ctl != null){
                    System.err.println(" - Recibir Nuevo RTP: " + ctl.getFormat());
                } else
                    System.err.println(" - Recibir un flujo RTP nuevo");
                if (participant == null)
                    System.err.println(" Flujo origen no identificado.");
                else {
                    System.err.println(" El flujo viene: " + participant.getCNAME()); }
                // Crea un reproductor enviandolo al Media Manager
                Player p = javax.media.Manager.createPlayer(ds);

```

```

        if (p == null)
            return;
        p.addControllerListener(this);
        p.realize();
        PlayerWindow pw = new PlayerWindow(p, stream);
        playerWindows.addElement(pw);
        // Notificar a initialize() que un Nuevo flujo ha llegado.
        synchronized (dataSync) {
            dataReceived = true;
            dataSync.notifyAll();}
    } catch (Exception e) {
        System.err.println("NewReceiveStreamEvent exception " + e.getMessage());
        return;}
    }
else if (evt instanceof StreamMappedEvent) {
    if (stream != null && stream.getDataSource() != null) {
        DataSource ds = stream.getDataSource();
        // Encontrar los formatos.
        RTPControl ctl = (RTPControl)ds.getControl("javax.media.rtp.RTPControl");
        System.err.println(" - Flujo previo ");
        if (ctl != null)
            System.err.println("    " + ctl.getFormat());
        System.err.println("    se ha identificado trnasmisor: " +
participant.getCNAME()); } }
else if (evt instanceof ByeEvent) {
    System.err.println(" - Fin \"bye\" de: " + participant.getCNAME());
    PlayerWindow pw = find(stream);
    if (pw != null) {
        pw.close();
        playerWindows.removeElement(pw); } } }
/**
 * EscucharControles para los reproductores.
 */
public synchronized void controllerUpdate(ControllerEvent ce) {
    Player p = (Player)ce.getSourceController();
    if (p == null)
        return;
    // Hacer esto cuando los reproductores internos estén listos.
    if (ce instanceof RealizeCompleteEvent) {
        PlayerWindow pw = find(p);
        if (pw == null) {
            // Algo salio mal.
            System.err.println("Error interno!");
            System.exit(-1);}
        pw.initialize();
        pw.setVisible(true);
        p.start();}
    if (ce instanceof ControllerErrorEvent) {
        p.removeControllerListener(this);
        PlayerWindow pw = find(p);
        if (pw != null) {
            pw.close();
            playerWindows.removeElement(pw); }
        System.err.println("recibir_medios internal error: " + ce);} }
/**
 * Una clase util para resolver las direcciones.
 */

```

```

class SessionLabel {
    public String addr = null;
    public int port;
    public int ttl = 1;
    SessionLabel(String session) throws IllegalArgumentException {
        int off;
        String portStr = null, ttlStr = null;
        if (session != null && session.length() > 0) {
            while (session.length() > 1 && session.charAt(0) == '/')
                session = session.substring(1);
            // Vemos si hay alguna dirección especificada.
            off = session.indexOf('/');
            if (off == -1) {
                if (!session.equals(""))
                    addr = session;
            } else {
                addr = session.substring(0, off);
                session = session.substring(off + 1);
                // Vemos si hay un Puerto especificado
                off = session.indexOf('/');
                if (off == -1) {
                    if (!session.equals(""))
                        portStr = session;
                } else {
                    portStr = session.substring(0, off);
                    session = session.substring(off + 1);
                    // Vemos si hay un ttl especificado
                    off = session.indexOf('/');
                    if (off == -1) {
                        if (!session.equals(""))
                            ttlStr = session;
                    } else {
                        ttlStr = session.substring(0, off);}}}
        if (addr == null)
            throw new IllegalArgumentException();
        if (portStr != null) {
            try {
                Integer integer = Integer.valueOf(portStr);
                if (integer != null)
                    port = integer.intValue();
            } catch (Throwable t) {
                throw new IllegalArgumentException();}
        } else
            throw new IllegalArgumentException();
        if (ttlStr != null) {
            try {
                Integer integer = Integer.valueOf(ttlStr);
                if (integer != null)
                    ttl = integer.intValue();
            } catch (Throwable t) {
                throw new IllegalArgumentException();}}}
}
/**
 * Clases graficas para el reproductor.
 */
class PlayerWindow extends Frame {
    Player player;

```

```
ReceiveStream stream;
PlayerWindow(Player p, ReceiveStream strm) {
    player = p;
    stream = strm;}
public void initialize() {
    add(new PlayerPanel(player));}
public void close() {
    player.close();
    setVisible(false);
    dispose();}
public void addNotify() {
    super.addNotify();
    pack();} }

/**
 * Clases graficas para el reproductor.
 */
class PlayerPanel extends Panel {
    Component vc, cc;
    PlayerPanel(Player p) {
        setLayout(new BorderLayout());
        if ((vc = p.getVisualComponent()) != null)
            add("Center", vc);
        if ((cc = p.getControlPanelComponent()) != null)
            add("South", cc); }
    public Dimension getPreferredSize() {
        int w = 0, h = 0;
        if (vc != null) {
            Dimension size = vc.getPreferredSize();
            w = size.width;
            h = size.height; }
        if (cc != null) {
            Dimension size = cc.getPreferredSize();
            if (w == 0)
                w = size.width;
            h += size.height; }
        if (w < 160)
            w = 160;
        return new Dimension(w, h); } }
public static void main(String argv[]) {
    if (argv.length == 0)
        prUsage();
    recibir_medios avReceive = new recibir_medios(argv);
    if (!avReceive.initialize()) {
        System.err.println("Error para inicializar.");
        System.exit(-1);}
    // Ver si recibir_medios fallo.
    try {
        while (!avReceive.isDone())
            Thread.sleep(1000);
    } catch (Exception e) {}
    System.err.println("Exiting recibir_medios");}
static void prUsage() {
    System.err.println("Uso: recibir_medios <session> <session> ...");
    System.err.println("    <session>: <address>/<port>/<ttd>");
    System.exit(0);}
} // fin de recibir_medios
```

## BIBLIOGRAFÍA.

---

- [1] Harvey. M. Deitel. “Introducción a los Sistemas Operativos”. Editorial Addison-Wessley. Segunda Edición. Año 2005. E.U.A.
- [2] Tanenbaum Andrew. S. “Redes de Computadoras”. Editorial Prentice Hall. Tercera Edición. Año 2005. México.
- [3] Tanenbaum Andrew. S. “Sistemas Operativos Modernos”. Editorial Prentice Hall. Primera Edición. Año 2005. México.
- [4] Abraham Silberschatz y Peter B. Galvin. “Operating System Concepts”. 2004 Addison-Wesley.
- [5] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer. Ed. Pearson-Prentice Hall. 2004.
- [6] Tratamiento Digital de las Señales. J. G. Proakis & D. G. Manolakis. 2003.
- [7] Data Structures Algorithms an Applications in Java. Sahani. Ed. Mc Graw Hill.
- [8] <http://www.mpeg.org/>
- [9] <http://www.mp3.org/>
- [10] <http://www.ietf.org/rfc1889.txt>