

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad en Ingeniería en Electricidad y Computación



“Implementación de un cliente de videoconferencia web basado en el protocolo SIP con la finalidad de facilitar la tutoría de clases a distancia”

INFORME DE MATERIA DE GRADUACIÓN

Previa la obtención del Título de:

Ingeniera en Telemática

Presentada por:

María José Argüello Vélez

Aurora Tatiana Cepeda Pozo

Guayaquil- Ecuador

Año

2012

AGRADECIMIENTO

*A DIOS nuestro Guía, que nos dio fortaleza,
perseverancia e inteligencia para alcanzar
satisfactoriamente esta meta .*

*A nuestros padres, por brindarnos su apoyo
incondicional, ustedes han sido el pilar
fundamental de nuestros logros.*

*A nuestros profesores, y a nuestro director
de tesis, por las enseñanzas impartidas
durante el camino de estudio.*

DEDICATORIA

*A Dios que es la luz que ilumina mi vida,
por el infinito amor y fortaleza que me ha
brindado para culminar esta importante
etapa en mi educación. A mis padres
que siempre creyeron en mí, por su confianza
apoyo y consejos.*

Aurora Tatiana Cepeda Pozo

*A Dios, quien es mi refugio y fortaleza, por darme
aliento y tomarme de la mano para seguir adelante en esta
etapa que culmina. A mis padres, quienes me forjaron en el
esfuerzo y el trabajo, por enseñarme a hacer las cosas con
amor, por su confianza y apoyo en todo momento. A mi
hermana, quien me hacía compañía en mis horas de estudio,
por enseñarme a reír siempre. A mis futuros hijos, quienes
son la inspiración en mi lucha diaria.*

María José Argüello Vélez

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este trabajo, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

María José Argüello Vélez

Aurora Tatiana Cepeda Pozo

TRIBUNAL DE SUSTENTACIÓN

Ing. Gabriel Astudillo

PROFESOR DE LA MATERIA DE GRADUACIÓN

Ing. Patricia Chávez

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

El proyecto consistió en la implementación de un cliente web de videoconferencia que utilizó el protocolo SIP para el establecimiento y la señalización de la sesión. La necesidad de utilizar una aplicación web de fácil uso y económica para realizar una videoconferencia es lo que ha impulsado el desarrollo de este proyecto.

Para mayor facilidad en el manejo del flujo de audio y video, se decidió utilizar el protocolo rtmp que fue liberado hace unos años por Adobe. Además se hizo uso de un servidor de VoIP llamado Asterisk y un servidor rtmp llamado Wowza Media Server, la conexión entre ambos servidores se realizó usando un puente de comunicación llamado Flashphoner, que es una aplicación java que se instaló en el servidor rtmp.

Por medio de la conexión de estas herramientas, se pudo realizar videoconferencias entre usuarios SIP y usuarios Flash desde una página web, lo que permite facilitar la tutoría de clases a distancia.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
DECLARACIÓN EXPRESA	iv
TRIBUNAL DE GRADUACIÓN	v
RESUMEN.....	vi
ÍNDICE GENERAL	vii
ABREVIATURAS	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS.....	xiv
INTRODUCCIÓN.....	xvi
CAPÍTULO 1	1
1.1. ANTECEDENTES.....	1
1.2. JUSTIFICACIÓN.....	2
1.3. DESCRIPCIÓN DEL PROYECTO	2
1.3.1. Objetivos Generales	2
1.3.2. Objetivos Específicos.....	3
1.4. METODOLOGÍA	3
1.5. PERFIL DE LA TESIS.....	3
CAPÍTULO 2	5
2.1. ASTERISK.....	5
2.1.1. Componentes y Características de Asterisk.....	7

2.1.2. Integración de Asterisk: Usando Interfaces	8
2.2. PROTOCOLOS NECESARIOS PARA SISTEMAS VOIP.....	9
2.2.1. Protocolo de Inicio de Sesión (SIP).....	10
2.2.2. Protocolo de Transporte en Tiempo Real (RTP)	13
2.2.3. RTMP	14
CAPÍTULO 3	15
3.1. INTRODUCCIÓN.....	15
3.2. HARDWARE.....	15
3.2.1. Servidor	16
3.2.2. Cliente	16
3.2.3. Cámara web y micrófono	16
3.3. SOFTWARE	16
3.3.1. Servidor	19
3.3.2. Cliente	18
3.3.3. Wowza Media Server.....	18
3.3.3.1. Características y Funcionalidades.....	19
3.3.4. Flashphoner.....	20
3.3.4.1. Versión Servidor	20
3.3.4.2. Versión Cliente.....	20
3.3.5. Licencias.....	21
3.3.5.1. Licencias para Wowza	21
3.3.5.2. Licencias para Flashphoner	26
3.4. INSTALACIÓN.....	26

3.4.1. Instalación de fuentes del kernel	26
3.4.2. Instalación de librerías	27
3.4.3. Instalación de Asterisk	27
3.4.4. Instalación JDK	29
3.4.5. Instalación del Servidor Web Apache.....	31
3.4.6. Instalación del Servidor Wowza Media.....	32
3.4.7. Instalación de Flashphoner	33
3.4.7.1. Instalación de Aplicación Servidor.....	33
3.4.7.2. Instalación de Aplicación Cliente.....	35
3.4.8. Instalación de Cliente Web	36
3.5. CONFIGURACIONES.....	36
3.5.1. Configuración de archivos de Asterisk	37
3.5.1.1. Configuración del archivo SIP.CONF	37
3.5.1.1.1. Sección General	38
3.5.1.1.2. Sección de extensiones	40
3.5.1.2. Configuración del archivo EXTENSIONS.CONF	41
3.5.1.2.1. Sección General	42
3.5.1.2.2. Contexto	42
3.5.2. Configuración de archivos Wowza Media Server	43
3.6. LÓGICA DEL SISTEMA DE VIDEOLLAMADA	47
3.6.1. Acción registrar	49
3.6.2. Acción llamar a usuario.....	51
CAPÍTULO 4	53

4.1. FUNCIONAMIENTO DEL SISTEMA	53
4.1.1. Iniciando los Servicios.....	53
4.1.2. Ingresando a la Aplicación Web.....	53
4.1.3. Configuración del Cliente SIP	56
4.1.4. Videollamada desde Cliente Web Flash.....	58
4.1.5. Videollamada desde Cliente SIP/ Eyebeam	61
4.2. PRUEBAS REALIZADAS.....	63
4.2.1. Pruebas con el Servidor de Asterisk detenido	63
4.2.2. Pruebas con el Servidor de Medios Wowza detenido.....	66
4.2.3. Pruebas con el Servidor de Medios Wowza y Servidor Asterisk habilitados ..	68
CONCLUSIONES Y RECOMENDACIONES	72
GLOSARIO.....	75
ANEXO.....	77
BIBLIOGRAFÍA.....	79

ABREVIATURAS

AGI	Interfaz de Salida de Asterisk.
AMI	Interfaz de Administración de Asterisk.
BSD	Sistema Operativo derivado de Linux.
GPL	Licencia Pública General.
HTTP	Protocolo de transferencia de hipertexto.
IAX	Protocolo nativo de la central telefónica Asterisk.
IVR	Respuesta de voz interactiva.
JDK	Paquete de desarrollo de herramientas JAVA.
LDAP	Protocolo de Acceso a Directorios Ligeros.
MGCP	Protocolo de control de dispositivos.
OS X	Sistema Operativo de MAC.
PBX	Sucursal de central telefónica privada.
PSTN	Red Telefónica pública conmutada.
RTCP	Protocolo de control de transporte en tiempo real.
RTMP	Protocolo de transporte multimedia en tiempo real.
RTP	Protocolo de transporte en tiempo real.
SCCP	Protocolo Propietario de Cisco de Control de Terminal.
SIP	Protocolo de inicio de sesión.
UDP	Protocolo de Datagrama de Usuario.
VoIP	Voz sobre el protocolo IP.
DRM	Gestión de derechos digitales.

ÍNDICE DE FIGURAS

Figura 2- 1. Conectividad entre centrales analógicas y digitales.	5
Figura 2- 2. Puertos FXS y FXO.	6
Figura 2- 3. Esquema de Comunicación entre dos terminales IP usando SIP.....	12
Figura 3- 1. Instalación JDK.....	31
Figura 3- 2. Salida de comando java –versión.	31
Figura 3- 3. Formulario para obtener licencia de Wowza.	33
Figura 3- 4. Ingresar Número de Licencia de WowzaMediaServer.....	33
Figura 3- 5. Formulario para Licencia Flashphoner.....	35
Figura 3- 6. Diagrama Esquemático de cliente Flash.....	47
Figura 3- 7. Diagrama Esquemático del funcionamiento de Wowza.....	48
Figura 3- 8. Diagrama esquemático de Servidor Asterisk.	49
Figura 3- 9. Diagrama de Flujo del registro de un usuario.....	50
Figura 3- 10. Diagrama de Flujo de la llamada entre dos usuarios.....	51
Figura 3- 11. Diagrama de Flujo del Sistema de Videollamada.....	513
Figura 4- 1. Iniciando los Servicios.	54
Figura 4- 2. Página principal de aplicación web.	55
Figura 4- 3. Configuración de usuario en servidor asterisk.	55
Figura 4- 4. Registro de usuario en servidor asterisk.	56
Figura 4- 5. Configuración de Eyebeam.....	57
Figura 4- 6. Usuario registrado en Eyebeam.....	58
Figura 4- 7. Marcación desde cliente Web.....	58
Figura 4- 8. Iniciando petición de llamada desde cliente Web.....	59

Figura 4- 9. Llamada entrante desde Cliente Web Flash a Cliente SIP Eyebeam. ...	60
Figura 4- 10. Llamada entrante de Cliente Web Flash a Cliente Web Flash.	60
Figura 4- 11. Marcación desde Cliente SIP Eyebeam.	61
Figura 4- 12. Iniciando petición de llamada entrante desde Cliente SIP Eyebeam..	62
Figura 4- 13. Llamada entrante de Cliente SIP Eyebeam a Cliente Web Flash.	62
Figura 4- 14. Comando para detener el servicio de Asterisk.	63
Figura 4- 15. Comando para reiniciar el servidor RTMP.	63
Figura 4- 16. Consola de Asterisk deshabilitada.	64
Figura 4- 17. Esperando la respuesta de registro del Cliente web en el servidor Asterisk.	64
Figura 4- 18. Error de registro en cliente web.	65
Figura 4- 19. Error de registro de cliente SIP debido a asterisk detenido.	66
Figura 4- 20. Comandos para iniciar asterisk y detener el servidor RTMP.	66
Figura 4- 21. Consola de Asterisk habilitada.	67
Figura 4- 22. Error de Conexión Fallida.	67
Figura 4- 23. Registro Exitoso de Cliente SIP Eyebeam.	68
Figura 4- 24. Comandos para reiniciar Servidores asterisk y RTMP.	68
Figura 4- 25. Registro exitoso de peers en Asterisk.	69
Figura 4- 26. Registro exitoso de peers en asterisk.	70
Figura 4- 27. Aceptación y Establecimiento de la Llamada.	70
Figura 4- 28. Videollamada del lado del cliente Web Flash.	71
Figura 4- 29. Videollamada del lado del cliente SIP Eyebeam.	71

ÍNDICE DE TABLAS

Tabla 3- 1. Características del servidor.....	16
Tabla 3- 2. Características del cliente.	16
Tabla 3- 3. Componentes de audio y video.....	167
Tabla 3- 4. Software instalado en el Servidor.....	17
Tabla 3- 5. Aplicaciones usadas como clientes.....	18
Tabla 3- 6. Precios Base para licencias Wowza Media Server.....	22-23
Tabla 3- 7. Precios de canales e instancias de servidor y complementos usando una licencia mensual.	24
Tabla 3- 8. Precios de canales e instancias de servidor y complementos usando una licencia por día.....	25
Tabla 3- 9. Precios del servidor y complementos usando una licencia Indefinida... .	25
Tabla 3- 10. Precios de Licencias para Flashphoner.....	26
Tabla 3- 11. Direcciones IP de Servidor (Máquina Virtual).....	37
Tabla 3- 12. Direcciones IP de Cliente 1 (Máquina Virtual)	37
Tabla 3- 13. Direcciones IP de Cliente 2 (Laptop).....	38
Tabla 3- 14. Configuración de Sección General del archivo sip.conf.....	38
Tabla 3- 15. Listado de parámetros configurados en la sección general del archivo sip.conf.....	40
Tabla 3- 16. Configuración de extensiones del archivo sip.conf.....	40
Tabla 3- 17. Listado de parámetros configurados en una extensión del archivo sip.conf.....	41
Tabla 3- 18. Configuración de Sección General del archivo extensions.conf.	42

Tabla 3- 19. Listado de parámetros configurados en Sección General del archivo extensions.conf.....	42
Tabla 3- 20. Configuración de contexto internos del archivo extensions.conf.	42
Tabla 3- 21. Configuración de contexto internos del archivo extensions.conf.	43
Tabla 3- 22. Configuración de archivo flashphoner.properties.	45
Tabla 3- 23. Configuración de archivo flashphoner.xml.....	47

INTRODUCCIÓN

El considerable aumento de las redes de comunicación y el diverso mundo de las telecomunicaciones, han impulsado el desarrollo de nuevas tecnologías como: la mensajería instantánea, las llamadas VoIP, las videollamadas, etc.; logrando cumplir un objetivo esencial que es acortar las distancias y mantenernos cerca de nuestros familiares, amigos y estaciones de trabajo. La videoconferencia es una forma de comunicación que tiene muchos beneficios y utilidad para quienes necesitan visualizar al comunicador o receptor del mensaje de voz. Por ejemplo en la relación profesor–alumno para recibir o brindar tutoría.

En la actualidad existen varios software de escritorio y muy pocas aplicaciones web que realizan esta comunicación, pero no son idóneos para los usuarios comunes debido a la complejidad de la aplicación, los privilegios de administrador que se necesitan para la instalación cuando el programa es de escritorio, o por su costo elevado.

La solución a estos problemas se resume en el uso de un poderoso servidor VoIP de Código Abierto como Asterisk que posee un canal llamado `chan_sip`, de uso exclusivo para el protocolo de señalización SIP, el cuál combinado con otro servidor Libre RTMP será la combinación perfecta para la implementación de una aplicación web de videoconferencia eficaz y económica.

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN

1.1. ANTECEDENTES

El desarrollo de las telecomunicaciones ha causado un gran impacto en la sociedad, obligándonos a involucrarnos en las nuevas tendencias tecnológicas para ser más eficientes y competitivos en el mundo laboral que cada vez exige más.

La videoconferencia es una forma de comunicación necesaria en el ámbito educacional y empresarial, pero que aún no se ha masificado debido a ciertos factores como dificultades de instalación o costos elevados. Por esta razón surge la necesidad de implementar un cliente de videoconferencia web que sea de fácil acceso, sin previa instalación y sobretodo que sea económico para satisfacer las necesidades de la mayoría de los usuarios.

Hace poco más de una década apareció un software para PBX llamado Asterisk de arquitectura de Código Abierto que ha tenido una excelente acogida y un continuo avance. En la actualidad cuenta con características de funcionabilidad, escalabilidad, flexibilidad e interoperabilidad que han permitido desarrollar poderosas soluciones VoIP a precios muy reducidos, convirtiéndolo en el programa de mayor avance tecnológico en la última década.

Uno de los beneficios que ha tenido Asterisk por ser de código abierto, es el perfecto entendimiento con diferentes librerías que varios desarrolladores han creado como JainSIP, que permite a las aplicaciones realizadas en lenguajes JAVA y JavaScript interactuar con la PBX usando el protocolo SIP, el cual está

representado en Asterisk con el modulo chan_sip. Se han tomado las funciones principales de éste módulo, para facilitar el registro y la ubicación del cliente web, además de permitir el establecimiento de la sesión durante la realización de la llamada, logrando una comunicación entre el cliente origen y destino.

1.2. JUSTIFICACIÓN

Debido a los requisitos que existen en las entidades educacionales con respecto a la educación continua y cada vez más completa de sus docentes, los maestros tienen la necesidad de complementar su conocimiento con estudios de postgrado, desarrollados generalmente fuera de su localidad de trabajo, lo cual les impide el dictado de clases de forma presencial.

Para este problema existen alternativas, como el uso de aplicaciones o programas de mensajería instantánea que incluyen la opción de videoconferencias, pero estas herramientas no son especializadas para el dictado de clases y además necesitan un proceso de instalación, que en ocasiones resulta tedioso para el usuario final.

1.3. DESCRIPCIÓN DEL PROYECTO

El presente proyecto consiste en la implementación de un cliente de videoconferencia web que utiliza el protocolo SIP para el establecimiento de la sesión y un componente Flash para el manejo del video, usando un servidor de PBX llamado Asterisk para su respectivo desarrollo.

1.3.1. Objetivos Generales

- Implementar un cliente para videoconferencia web basado en el protocolo SIP que facilite la tutoría de clases a distancia de una manera eficiente y a bajo costo.

1.3.2. Objetivos Específicos

- Usar los conocimientos adquiridos previamente en las materias de Desarrollo de Aplicaciones Web, Fundamentos de Linux y Fundamentos de redes de datos.
- Utilizar y gestionar el modulo chan_sip de Asterisk para el uso del protocolo SIP.
- Utilizar herramientas Flash que usan el protocolo RTMP para manejar el flujo multimedia.
- Utilizar un puente de comunicación entre RTMP y SIP para comunicar los servidores Wowza Media y Asterisk.

1.4. METODOLOGÍA

Con el fin de alcanzar los objetivos planteados, se instalarán los servidores Asterisk y Wowza Media sobre el sistema operativo Centos 5. En el lado del cliente web se utilizará un navegador libre con un componente de Adobe Flash. Y las pruebas correspondientes se realizarán con dos webcams, dos micrófonos y un cliente SIP de escritorio.

1.5. PERFIL DE LA TESIS

La finalidad de esta tesis es facilitar la videoconferencia web de manera eficiente usando Asterisk como una central VoIP y Wowza como servidor RTMP.

En el capítulo 2, se puntualizarán los principios teóricos, las características básicas y aplicaciones de Asterisk para soluciones de Voz sobre IP; además se explicará de manera general los protocolos usados por las herramientas que escogimos para el desarrollo del proyecto.

En el capítulo 3, se especificarán los detalles del análisis, diseño, instalación de componentes y librerías necesarias para la implementación del proyecto.

Por último en el capítulo 4, se describirá el funcionamiento y los pasos a seguir para utilizar el cliente de videoconferencia web que se ha implementado, así como las pruebas de conexión que se han realizado.

CAPÍTULO 2

2. ASTERISK, PROTOCOLOS PARA VOIP

2.1. ASTERISK

Asterisk es un software que permite controlar y gestionar varios tipos de comunicaciones ya sean analógicas, digitales o VoIP mediante todos los protocolos que implementa. Esta aplicación fue creada en 1999 por Mark Spencer de la empresa Digium y fue donada a la comunidad con licencia libre, quienes han colaborado mucho en sus mejoras.

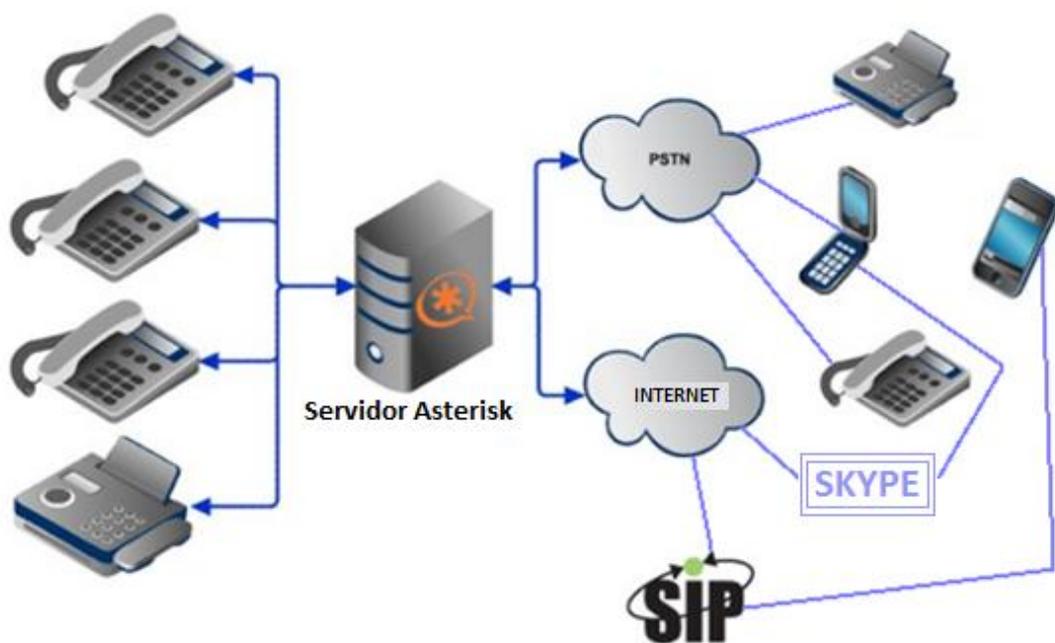


Figura 2- 1. Conectividad entre centrales analógicas y digitales[17].

Se ejecuta sobre Linux, BSD, Windows (emulado), OS X y proporciona todas las características de una central telefónica PBX, que funciona como una red telefónica interna la cual se comunica con la central telefónica tradicional (analógica) logrando

así que centralitas analógicas y digitales se integren con la tecnología más actual: Voz sobre IP[4].

Este software puede también interoperar con casi todos los equipos de telefonía basados en estándares usando hardware relativamente barato, permitiendo comunicar perfectamente las redes de telecomunicaciones conmutadas por circuito con redes de datos conmutadas por paquetes mediante el uso de cualquier tarjeta compatible con las interfaces. La selección de una tarjeta de telefonía de código abierto era limitado, pero debido al interés en Asterisk la demanda en el mercado de tarjetas ha crecido, dando paso a nuevos fabricantes. Una de las maneras más populares y rentables para conectarse a la PSTN es usar las tarjetas de interfaz analógica y digital que se desarrollaron en el proyecto de Telefonía Zapata, cada una de estas tarjetas requieren puertos FXO y FXS respectivamente.

El puerto FXS es la interfaz externa del abonado que provee el tono de marcado y el voltaje del timbre que alertan al usuario sobre la existencia de una llamada entrante y el puerto FXO es la interfaz externa de la central que recibe la línea analógica y envía una indicación de colgado o descolgado, esta interfaz está siempre adjunta a un dispositivo, ya sea teléfono o fax, llamado generalmente “dispositivo FXO”[5].

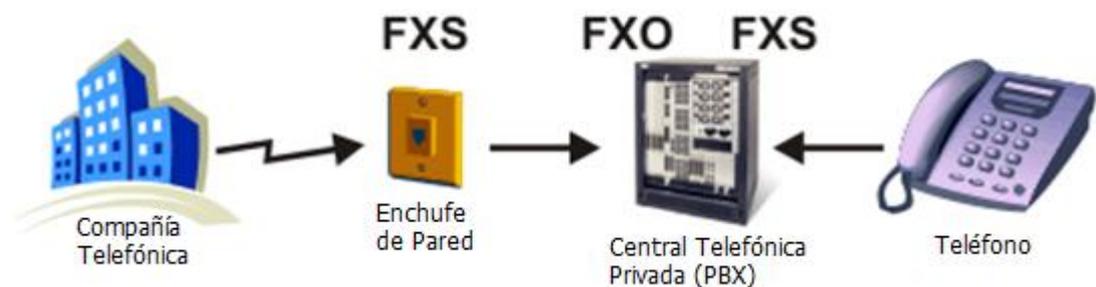


Figura 2- 2. Puertos FXS y FXO [18].

Su mayor ventaja es tener licencia GPL considerándose una aplicación de código abierto, lo cual la hace muy codiciada y accesible por desarrolladores, estudiantes y empresas que desean crear sus propios sistemas VoIP. Asterisk está integrado con varios módulos como el canal de sesión SIP, el cual nos permitirá comunicarnos con teléfonos y puertos de enlaces, facilitándonos el manejo de la señalización en la comunicación. Además es programable y se adapta a varios lenguajes de programación (php, perl, java, etc.) con la ayuda de sus interfaces AMI y AGI.

2.1.1. Componentes y Características de Asterisk.

Asterisk incluye una variedad de componentes necesarios para crear un sistema PBX, un sistema IVR o cualquier tipo de solución de comunicaciones.

Entre los componentes que incluye, tenemos[7]:

- Controladores para varios protocolos VoIP.
- Controladores para las tarjetas de interfaz y dispositivos PSTN.
- Enrutamiento y gestión de llamadas para las llamadas entrantes.
- Generación y enrutamiento de llamadas salientes.
- Funciones de Gestión de Medios (grabar, reproducir, generar el tono, etc.)
- Detalle de Grabación de Llamadas para contabilidad y facturación.
- Transcodificación (Conversión de un formato a otro)
- Conversión de Protocolos
- Integración de Base de Datos para acceso a la información por bases de datos relacionales.
- Integración de Servicios Web para acceso a los datos usando protocolos estándares de internet.

- Integración de LDAP para acceder a sistemas de directorio corporativos.
- Transición de Llamada Única y Múltiple.
- Grabación de Llamadas y Funciones de Monitoreo.
- Lenguaje de escritura del plan de marcado integrado para procesamiento de llamadas.
- Gestión de llamadas externas en cualquier programación o lenguaje de escritura a través de AGI.
- Síntesis de Voz (conocido como “text-to-speech”) en varios idiomas y dialectos utilizando motores de terceros.
- Reconocimiento de Voz en varios idiomas utilizando motores de reconocimiento de terceros.

Además, Asterisk provee un gran número de características como: servicios de Correo de Voz con Directorio, Conferencias de llamadas, Respuesta Interactiva de Voz, Colas de Llamadas. Soporta Llamadas tripartitas, servicios de Identificación de Llamadas (Caller ID), SIP, H.323, MGCP (solo gestor de llamadas), y SCCP/Skinny.

2.1.2. Integración de Asterisk: Usando Interfaces

Muchas veces es necesario usar aplicaciones externas junto con Asterisk, para agregar funcionalidad e integrar nuestra central telefónica con nuevas tecnologías. Para ello, Asterisk nos facilita las interfaces AMI y AGI.

La Interfaz para el Manejo de Asterisk (AMI) permite a un programa cliente conectarse a una instancia de Asterisk y ejecutar comandos o leer eventos a través de un flujo TCP/IP, es decir, permite a programas externos controlar y

monitorear un sistema creado sobre Asterisk. Los integradores (quienes añaden componentes al programa) encuentran esto muy útil cuando desean realizar un seguimiento al estado de los clientes de telefonía dentro de Asterisk, y direccionar a este cliente basados en reglas personalizadas y posiblemente dinámicas.

Por otro lado, existe la Interfaz de Puerta de Enlace (AGI) que añade funcionalidad a Asterisk, permitiendo controlar el plan de marcado (Dialplan) mediante programas externos desarrollados en diferentes lenguajes de programación (Perl, PHP, Python, C, Java, etc). Son usualmente utilizados para una lógica avanzada. Existen cinco tipos de AGIs[1]:

- AGI Normal, puede controlar el plan de marcado, que es llamado en el archivo `extensions.conf`.
- Async AGI, permite scripting de AGI asíncrono.
- EAGI, permite a la aplicación acceder y controlar el canal de sonido, además de la interacción con el plan marcado.
- FastAGI, puede utilizarse para hacer el procesamiento en una máquina remota a través de una conexión a la red.
- DeadAGI, da acceso a un canal muerto después de terminada la llamada. Está obsoleta a partir de Asterisk 1.6.

2.2. PROTOCOLOS NECESARIOS PARA SISTEMAS VOIP

Existen varios protocolos gracias a los cuales es posible la transmisión y recepción de voz y video a través de un sistema de Voz sobre IP. Cada uno de estos protocolos está encargado de gestionar una parte de la comunicación, ya sea el

establecimiento, inicio y finalización de la llamada o la transmisión en sí de datos multimedia.

Asterisk soporta estos protocolos de VoIP, que específicamente realizan el trabajo de señalización y establecimiento de la comunicación: IAX, SIP, H.323, MGCP, SCCP/Skinny. De los cuales el más relevante para nosotros es el protocolo de inicio de sesión SIP.

Además de estos protocolos son necesarios también los códecs, que son el medio por el cual puede ser convertida la voz analógica en una señal digital para ser transportada a través de la red. Los códecs se diferencian unos de otros por sus requisitos de ancho de banda y calidad, entre los que soporta Asterisk están el G.711, G.726, G.729A, GSM, iLBC, Speex, MP3. de acuerdo a la información proporcionada en [1].

Pero para que pueda implementarse un sistema de voz sobre IP deben utilizarse dos canales, uno para señalización y otro para transmisión multimedia, es decir, que para completar el proceso de transmisión y recepción de información multimedia se usan también protocolos de transporte que sirven a la capa de aplicación como lo son los protocolos RTMP y RTP, que serán explicados más adelante.

2.2.1. Protocolo de Inicio de Sesión (SIP)

El protocolo de inicio de sesión SIP trabaja en la capa de aplicación y permite crear, modificar y terminar una conexión, es decir, inicia y dirige las sesiones o conexiones entre dos o más participantes. Se encarga solo de establecer y señalar la llamada, no entrega flujos de información multimedia ni controla la entrega de estos flujos[16]. Está basado en arquitectura cliente servidor y utiliza

para sus cabeceras el mismo formato de los mensajes HTTP, usa el puerto bien conocido 5060 para sus comunicaciones.

SIP soporta cinco facetas de establecimiento y terminación de comunicaciones multimedia. Ubicación de usuario, disponibilidad de usuario, capacidades de usuario, configuración de la sesión y administración de sesiones. Como podemos observar en la figura 2.3, SIP utiliza dos tipos de mensajes: las solicitudes y las respuestas. Las solicitudes son basadas en texto, generalmente enviadas por los clientes y pueden ser de diferentes tipos: "invite", "ack", "bye", "cancel", "options" o "register". Cada tipo o método cumple con una función específica dentro del proceso de establecimiento y descripción de la sesión que realiza SIP. Por ejemplo, "invite" es usado para invitar a un usuario a una sesión, "ack" permite reconocer que el intercambio de mensajes entre cliente y servidor es confiable, "bye" se utiliza para informar al servidor que se quiere terminar una llamada, "cancel" permite cancelar una solicitud ya realizada al servidor, "options" se usa para solicitar información sobre las capacidades del usuario llamado, y "register" se usa para registrar la dirección de la parte llamada con el servidor según los datos encontrados en [2].

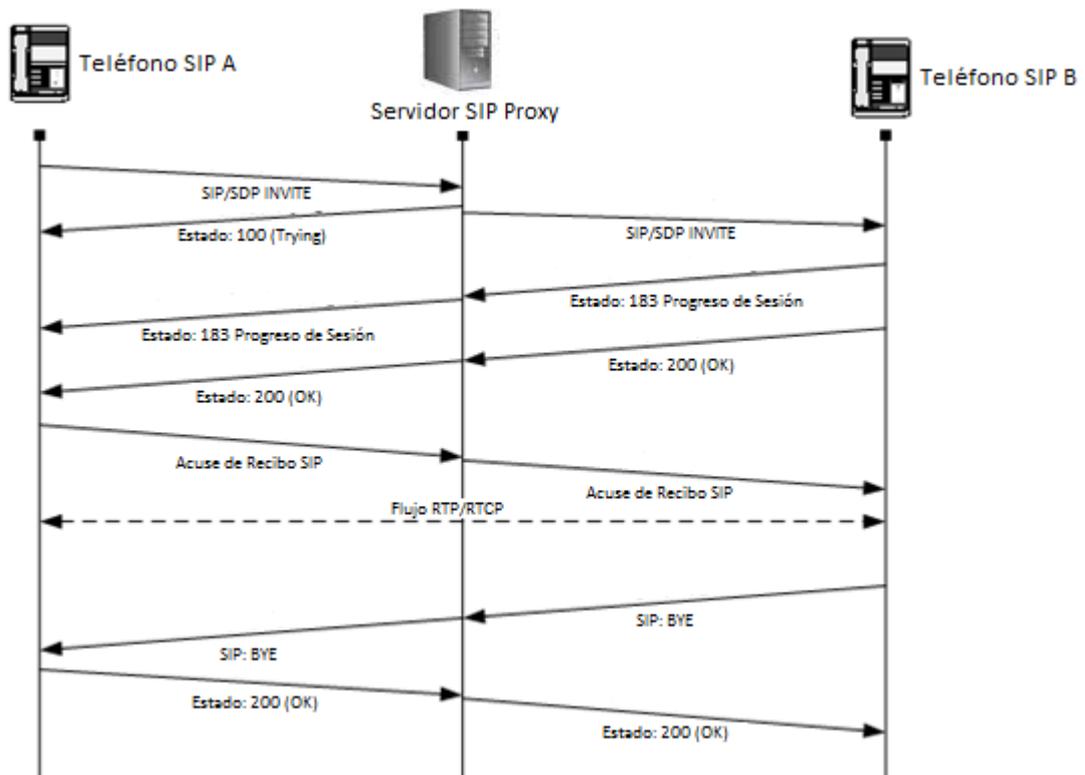


Figura 2- 3. Esquema de Comunicación entre dos terminales IP usando SIP[19].

Por otro lado, las respuestas a estas solicitudes son emitidas por los servidores y consisten de códigos numéricos que informan a los clientes sobre el estado de las solicitudes realizadas, estos códigos son muy similares a los utilizados en el protocolo HTTP, como por ejemplo, códigos 1xx para peticiones recibidas que están en proceso de respuesta, códigos 2xx para peticiones exitosamente recibidas y aceptadas, códigos 3xx enviados por servidores proxy para agentes que han cambiado de dirección, códigos 4xx para respuestas no encontradas o que no pueden ser procesadas por el servidor, códigos 5xx para errores en el servidor y códigos 6xx para fallas globales del sistema.

Además, las llamadas procesadas por este protocolo inicialmente son establecidas por un SIP proxy, que hace las veces de intermediario entre

terminales cliente y servidor y luego que ya está establecida la conexión, los agentes de usuario se comunican directamente entre sí, lo que es conocido como una comunicación punto a punto.

Como se ha explicado, el protocolo SIP establece la conexión entre dos clientes, es decir, se encarga del inicio de la comunicación pero no trabaja solo, para hacer posible la comunicación multimedia debe trabajar acompañado de otro protocolo que permita el transporte en sí de contenido multimedia, este protocolo es conocido como RTP.

2.2.2. Protocolo de Transporte en Tiempo Real (RTP)

RTP es un protocolo de transporte para aplicaciones en tiempo real, es decir, que realiza la entrega punto a punto de información que es transmitida en tiempo real como por ejemplo, audio y video interactivo. Este protocolo se encarga del manejo de aspectos relativos a la temporización, marcando los paquetes UDP con la información necesaria para la correcta entrega de los mismos en recepción, la identificación del tipo de carga, el número de secuencia y el monitoreo de la entrega.

Es así que RTP garantiza la entrega de este tipo de paquetes y su orden de llegada correcto como lo encontramos en [15].

Para asegurar la calidad del servicio, RTP no trabaja solo sino que se vale de otro protocolo llamado RTCP, que se encarga de realizar la parte del control de los medios de transmisión, es decir, que permite detectar situaciones de congestión en la red y realizar correcciones si fuese necesario, usa otro puerto adicional de numeración seguida al puerto que usa RTP.

Pero específicamente con el protocolo SIP trabaja en conjunto, encargándose solo del transporte de los datos de voz en tiempo real.

2.2.3. RTMP

El protocolo de mensajería en tiempo real (RTMP) fue desarrollado por Macromedia para transmitir audio, video y datos sobre la web, entre un reproductor flash y un servidor, inicialmente era un protocolo propietario que luego fue liberado por Adobe para su uso público.

En nuestra aplicación este protocolo es usado por el servidor de medios Wowza que permite que el video y el audio sean reproducidos en el cliente web. El protocolo RTMP tiene múltiples variantes, una de ellas, es RTMPT una forma del protocolo RTMP pero encapsulado a través de un túnel HTTP, es decir, usa peticiones HTTP para atravesar los cortafuegos y poder transmitir correctamente la información multimedia.

CAPÍTULO 3

3. IMPLEMENTACIÓN

3.1. INTRODUCCIÓN

La presente solución tecnológica toma las bondades del software libre para realizar un sistema sólido, eficaz y económico que podrá ser utilizado por muchos usuarios que necesitan realizar tutorías a distancia y no cuentan con el software adecuado.

Su principal funcionamiento se basa en el servidor de centrales telefónicas Asterisk que nos permite implementar poderosas soluciones VoIP de forma gratuita, y que se pueden fácilmente integrar con otros servidores de código abierto con un perfecto funcionamiento.

3.2. HARDWARE

El proceso de selección del hardware siempre ha sido de mucha importancia para el rendimiento y desempeño de su sistema. Cabe recalcar que este proyecto requiere utilizar 3 máquinas para su implementación, dos que actúan como clientes y una que actúa como servidor, pero por la falta de recursos económicos solo se utilizaron 2 computadoras, la primera que funciona como cliente y servidor, donde el servidor se encuentra instalado en forma virtual, y la segunda que solo actúa como cliente.

Para el caso en particular de la máquina que actúa como cliente y servidor se ha tomado una computadora de arquitectura x_86 como servidor principal, con las siguientes características:

3.2.1. Servidor

Se utilizó como servidor una máquina virtual, que está dentro de un computador de escritorio que tiene las características mostradas en la tabla 3.1.

Procesador	Intel Core Duo de 2.8 GHZ
Tarjeta de Red	10/100 Mbps
Memoria Ram	2 GB
Disco Duro	300 GB

Tabla 3- 1. Características del servidor.

3.2.2. Cliente

Se utilizaron dos clientes para realizar la videoconferencia. El primer cliente se describió anteriormente en la sección 3.2.1. El segundo cliente es una laptop HP modelo G42 de arquitectura x_64, con las características expuestas en la tabla 3.2.

Procesador	Intel Core i 3 2.8 GHZ
Tarjeta de Red	10/100 Mbps
Memoria Ram	3 GB
Disco Duro	500 GB

Tabla 3- 2. Características del cliente.

3.2.3. Cámara web y micrófono

Para la captura del audio y video utilizamos los componentes de hardware descritos en la tabla 3.3.

Cámara web USB	Vimicro
Micrófono	Digium
Cámara web incorporada laptop	HPg42
Micrófono incorporado en laptop	HPg42

Tabla 3- 3: Componentes de audio y video

3.3. SOFTWARE

3.3.1. Servidor

En esta sección se aclara lo siguiente: el servidor que se ha utilizado para este proyecto está instalado en una forma virtual con un sistema operativo de Linux, sobre un sistema operativo Windows XP arquitectura X86. Esto se realizó debido a la falta de recursos económicos para conseguir otra computadora. En la tabla3.4 observaremos los programas de código abierto que es necesario instalar en el servidor.

Software de Sistema Operativo	Red Hat Enterprise Linux 5.4 Arquitectura: X86
Software VoIP	Asterisk 1.6
Java Development kit	Versión 6 update 29
Servidor web	Apache
Software RTMP	Wowza Media Server 3
Aplicación bridge entre VoIP y Rtmp	Flasphoner Server 1.0.5.1057
Aplicación web	Flash client 1.0.5.124

Tabla 3- 4: Software instalado en el Servidor.

Aparte de estos programas es necesario instalar las siguientes librerías para que nuestro software Asterisk trabaje correctamente.

3.3.2. Cliente

Se utilizaron dos clientes: un cliente web y un cliente sip que contienen las características descritas en la tabla 3.5.

Cliente web	Firefox 3.6.24 Plugin Adobe Flash Player 10
Cliente SIP	Eyebeam

Tabla 3- 5: Aplicaciones usadas como clientes.

3.3.3. Wowza Media Server

Wowza Media Server es un software de servidor unificado desarrollado por Wowza Media Systems. El servidor se utiliza para la transmisión de vídeo en directo y bajo demanda en formato Flash FLV y H.264, audio y aplicaciones RIA (Rich Internet Applications) sobre redes IP públicas y privadas hacia PCs de escritorio, portátiles y Tablet PC, dispositivos móviles, IPTV set-top boxes, sets de televisión conectados a internet y otros dispositivos conectados a la red.

El servidor es una aplicación Java compatible con los siguientes sistemas operativos: Linux, Mac OS X, Solaris, Unix y Windows. Wowza Media Server puede transmitir a múltiples tipos de clientes y dispositivos de reproducción en forma simultánea, incluyendo el reproductor de Adobe Flash, Microsoft Silverlight player, Apple QuickTime Player y dispositivos iOS (iPad, iPhone, iPod Touch), teléfonos móviles 3GPP (Android, BlackBerry OS, Symbian , etc), IPTV

set-top boxes (Amino, Enseio, Roku y otros), y consolas de juegos como Wii y PS3.

Adicionalmente se puede utilizar el Wowza Media Server para poder proporcionar archivos de audio MP3/M4A que podrán ser escuchados - pero no grabados- a través de un reproductor Flash en la página web.

Wowza apoya todas las variantes del protocolo Real Time Messaging: RTMP, RTMPT, RTMPS, RTMPE y RTMPTE como también los protocolos Real Time RTP, RTSP y MPEG-TS.

3.3.3.1. Características y Funcionalidades

Este servidor de medios que tiene poco tiempo de haber sido lanzado al mercado pero que ha sido una herramienta muy poderosa para mejorar muchos de los sistemas de medios implementados en software posee características muy relevantes que le dan fortaleza a su funcionamiento. Según la información de [11], entre estas características tenemos:

- Alto rendimiento.- Wowza Media Server es un servidor multi-hilo de java de 64 bits que ofrece hasta 10Gbps de rendimiento por servidor de flujo continuo para video bajo demanda y en directo en hardware estándar.
- Unificado.- Este servidor unifica transmisiones multi-protocolo y multi-usuario H.264 en un único flujo de trabajo que permite transmitir a cualquier plataforma de medios y a cualquier pantalla simultáneamente.
- Escalable.- Permite realizar soluciones con escalabilidad multiservidor de manera más sencilla y rentable. El balanceo de carga robusto basado en demanda del servidor Wowza asegura que los flujos directos y bajo

demanda son distribuidos eficientemente a través de múltiples servidores sin sobrecargar la infraestructura usada.

- Extensible.- Sus funcionalidades pueden ser ampliadas utilizando un conjunto robusto de APIs que permite la personalización con diferentes aplicaciones y ambientes.

3.3.4. Flashphoner

Es un software de código abierto que permite establecer la comunicación entre aplicaciones Flash Player y un servidor VoIP, es decir, sirve como puerta de enlace entre los protocolos RTMP y SIP.

Es así, que Flashphoner sirve para establecer comunicaciones Flash-VoIP o para desarrollar teléfonos Flash completamente funcionales. Para ello, tenemos a disposición tanto Servidor Flashphoner como Cliente Flashphoner con licencias de código abierto, las que podemos encontrar en [10].

3.3.4.1. Versión Servidor

Es una aplicación en Java que conecta los servidores RTMP y VoIP a través del protocolo SIP. Usa el protocolo RTMP para comunicarse con el cliente Flash y el protocolo SIP para la comunicación con el servidor Asterisk.

Soporta el sistema de marcación por tonos DTMF (Dual-Tone Multi-Frequency), el mensaje instantáneo vía SIP MESSAGE y los siguientes códecs; para **audio**: Speex, G.711, G.729 y para **video**: H.263, H.264.

3.3.4.2. Versión Cliente

Es una aplicación web desarrollada en javascript y Flex/ActionScript 3 que usa el protocolo RTMP para señalización y transmisión multimedia. Soporta códecs NellyMoser, Speex y Sorenson spark, que son los usados por Flash.

3.3.5. Licencias

3.3.5.1. Licencias para Wowza

Este software ofrece varias versiones de licencias, una para cada conjunto de actividades que realice la empresa, según las necesidades que tenga el usuario del programa. Estas licencias facilitan el funcionamiento de Sitios de internet de medios y redes sociales; Proveedores de Servicio de Flujo Continuo y Alojamiento WEB; Empresas; OEMs, Desarrolladores y Otros que ofrecen características que hacen de este software la base para aplicaciones de flujo continuo de medios de comunicación y empresas de entretenimiento. Soporta Flash, iPhone / iPad, teléfonos inteligentes, tabletas, decodificadores IPTV, etc; transmisión en vivo y bajo demanda, interactivos de audio, video y chat de texto y grabación remota.

Dependiendo del tiempo de validez, existen licencias mensuales, diarias y perpetuas. Las mensuales y las diarias proveen solo una clave de licencia para muchos servidores y las actualizaciones son gratuitas. A diferencia de la licencia perpetua que provee claves de licencia separadas para cada servidor y complemento y que se paga una sola vez y se usa para siempre, además las actualizaciones menores son gratuitas. Para todas las licencias se ofrece soporte gratis en línea y por correo electrónico.

En la tabla 3.6 se muestran los precios bases de las licencias para el servidor de medios Wowza según el tiempo de uso.

	Mensual	Diario	Indefinido	30 días de Prueba
Precio Base	\$55	\$5	\$995	Gratis
Ideal para	Cualquier cliente, cualquier proyecto.	Solo pruebas por días	Cualquier cliente que no sea proveedor de algún tipo de servicios de uso en Internet.	Cualquier cliente, cualquier proyecto en un ambiente de prueba por 30 días.
Licencia	Una clave de licencia para cualquier número de servidores y Complementos.	Una clave de licencia para cualquier número de servidores y Complementos.	Claves de licencia independientes, una para cada servidor y Complemento.	Una clave de licencia solo sirve para una instancia de servidor.
Descuentos	Automático para el uso de múltiples servidores y	Tarifas diarias sin descuento.	Inmediata para las transacciones de múltiples	No disponible

	Complementos.		licencias.	
Actualizaciones	Todas GRATIS.	Todas GRATIS.	Menor: GRATIS. Total: Se aplican tarifas.	No disponible
Beneficios	No existe el riesgo de cancelarse en cualquier momento.	No existe el riesgo de cancelarse en cualquier momento.	Pague una vez y lo utiliza para siempre.	Existe el riesgo de cancelarse después de un mes.
Soporte	GRATIS en línea y correo electrónico	GRATIS en línea y correo electrónico	GRATIS en línea y correo electrónico	GRATIS en línea y correo electrónico

Tabla 3- 6: Precios Base para licencias Wowza Media Server

Los precios están en dólares de EE.UU. y no los impuestos aplicables. Además estas licencias sólo se pueden utilizar en un servidor que está conectado a Internet.

También existen descuentos para las organizaciones educativas y sin fines de lucro, la información de estos descuentos es en forma interna comunicándose con el contacto edu@wowza.com.

Adicional a estos precios se añade el cobro en función del número máximo de instancias y canales en uso en un día del servidor y los complementos. Los precios varían según el tipo de licencia escogida.

La tabla 3.7 muestra el precio de los canales e instancias del servidor y complementos según la licencia mensual.

	Servidor	Transcoder	nDVR	DRM
Instancia / canales	Instancia por mes	Canal por mes	Instancia por mes	Instancia por mes
Primeros 4 (1 a 4)	\$55	\$30	\$20	\$55
Siguientes 6 (5 a 10)	\$50	\$27	\$18	\$50
Siguientes 10 (11 a 20)	\$45	\$24	\$16	\$45
Siguientes 30 (21 a 50)	\$40	\$21	\$14	\$40
Siguientes 50 (51 a 100)	\$35	\$18	\$12	\$35

Tabla 3- 7: Precios de canales e instancias de servidor y complementos usando una licencia mensual.

Según las licencias diarias, los precios de los canales e instancias del servidor y complementos son los mostrados en la tabla 3.8.

	Servidor	Transcoder	nDVR	DRM
Instancia/canales	Instancia por día	Canal por día	Instancia por día	Instancia por día
cualquier cantidad	\$5	\$3	\$2	\$5

Tabla 3- 8: Precios de canales e instancias de servidor y complementos usando una licencia por día.

En el caso de la licencia indefinida, los precios son individuales por cada servidor y por cada complemento usado, aquí se aplican descuentos por volumen por la compra de varias licencias en la misma transacción de compra. Con estas licencias obtenemos instancias y canales con conexión ilimitados, como se muestra en la tabla 3.9.

	Servidor	Transcoder	nDVR	DRM
Licencia única	\$995	\$1495	\$495	\$995
2 a 4 licencias	\$895	\$1345	\$445	\$895
5 o más licencias	\$799	\$1199	\$399	\$799

Tabla 3- 9: Precios del servidor y complementos usando una licencia Indefinida.

También existe la licencia de prueba gratis por 30 días, la cual usamos en este proyecto para evaluar el total funcionamiento del software. Contiene todos los complementos incluyendo el Transcoder, nDVR y DRM, pero solo posee una instancia de servidor ya sea física o virtual.

Wowza ofrece además una licencia para desarrolladores que dura 180 días, provee una sola clave de licencia para una sola instancia de servidor, y tiene limitaciones en cuanto a la cantidad de conexiones y complementos, pero que son suficientes para realizar pruebas e implementaciones de nuevos sistemas. La información presentada en esta sección la encontramos en [11].

3.3.5.2. Licencias para Flashphoner

La aplicación Servidor de Flashphoner posee también varios tipos de licencias. Para la implementación de nuestro proyecto usamos la licencia para desarrolladores que nos permite tener un máximo de 10 conexiones, si se desea implementar el sistema para más número de conexiones se deben tener en cuenta los costos de la tabla 3.10 extraídos de [10].

	Licencia de Desarrolladores	Licencia de proveedor de soluciones	Licencia de proveedor de servicios
Características	GRATIS	\$2000	\$500
Conexiones/líneas	máximo 10	ilimitadas	máximo 20

Tabla 3- 10: Precios de Licencias para Flashphoner.

3.4. INSTALACIÓN

La información presentada en las secciones 3.4.1, 3.4.2 y 3.4.3 fue tomada de [1].

3.4.1. Instalación de fuentes del kernel.

Después de haber instalado el sistema operativo, debemos estar seguros que tenemos la misma versión del kernel y de las fuentes del kernel. Los comandos para verificar se muestran a continuación:

```
rpm -q kernel
```

```
rpm -q kernel-devel
```

En caso de que no sean iguales debemos escribir lo siguiente:

```
yum install kernel
```

```
yum install kernel-devel
```

3.4.2. Instalación de librerías

Es necesario instalar un conjunto de librerías que ayudan al correcto funcionamiento del software de VoIP. Para lograr la instalación se debe ejecutar el comando `yum install` seguido del nombre de cada una de las siguientes librerías a instalar; bison-devel, ncurses, ncurses-devel, libtermcap, libtermcap-devel, zlib, zlib-devel, openssl, openssl-devel, gnutls-devel, gcc, gcc-c++, newt, newt-devel y libtool.

3.4.3. Instalación de Asterisk.

Primero, se debe ubicar el directorio `/usr/src/` ejecutando

```
cd /usr/src/
```

Después, se descargan los paquetes `asterisk- 1.6.2.20.tar.gz`, `asterisk-addons-1.6.2.3.tar.gz`, `dahdi-linux-complete-2.4.1.2+2.4.1.tar.gz` y `libpri-1.4.11.5.tar.gz`, los mismos que se encuentran disponibles en los enlaces oficiales de asterisk.

A continuación se descomprimen los paquetes con el siguiente comando `tar xzvf` seguido del nombre de cada archivo, en esta caso, `asterisk- 1.6.2.20.tar.gz`, `asterisk-addons-1.6.2.3.tar.gz`, `dahdi-linux-complete-2.4.1.2+2.4.1.tar.gz` y `libpri-1.4.11.5.tar.gz`.

En este momento, se instalan los paquetes respetando el siguiente orden para asegurar que las dependencias requeridas por los scripts de configuración de libpri, dahdi o asterisk sean previamente instaladas.

- libpri-1.4.11.5
- dahdi-linux-complete -2.4.1.2+2.4.1
- asterisk- 1.6.2.20
- asterisk-addons-1.6.2.3

Respetando el orden establecido se debe acceder al directorio libpri-1.4.11.5 para instalar paquete ejecutando los comandos que se muestran a continuación:

```
cd libpri-1.4.11.5  
  
make clean  
  
make  
  
make install
```

Después, debemos ubicar el directorio dahdi-linux-complete -2.4.1.2+2.4.1 y ejecutar las siguientes sentencias:

```
cd dahdi-linux-complete -2.4.1.2+2.4.1  
  
make clean  
  
make  
  
make install  
  
make config
```

Ahora se debe compilar e instalar el software asterisk, pero antes se deben juntar todas las librerías que se encuentran en el servidor y que son requeridas por el software para lograr una instalación exitosa.

```
cd asterisk- 1.6.2.20  
make clean  
./configure  
make  
make install  
make samples  
make config
```

De manera opcional se pueden crear configuraciones de ejemplo que tiene el servidor VoIP escribiendo la siguiente sentencia

```
make samples
```

Finalmente se instala asterisk-addons-1.6.2.3 de la siguiente manera

```
cd asterisk-addons-1.6.2.3  
make clean  
./configure  
make install  
make samples
```

3.4.4. Instalación JDK.

Para instalar el kit de desarrollo de java primero se necesita establecer el directorio de descarga.

```
cd /opt/
```

Ahora se descarga la aplicación para Linux en el siguiente enlace:

<http://download.oracle.com/otn-pub/java/jdk/6u30-b12/jdk-6u30-linux-i586.bin>

Antes de instalarlo se le debe dar permiso de ejecución

```
chmod +x jdk-6u29-linux-i586.bin
```

A continuación se instala la aplicación.

```
./jdk-6u29-linux-i586.bin
```

Enseguida se creará un directorio de nombre `jdk1.6.0_29`, el mismo que se necesitará para configurar un enlace simbólico llamado `java`:

```
ln -s jdk1.6.0_29 java
```

Luego se debe modificar el archivo `.bash_profile` del usuario que se está utilizando, en este caso el `root`.

```
cd /root/  
vi .bash_profile
```

Después se debe abrir el archivo `bash_profile` y buscar la variable `$exportPATH`, se modifica el archivo de esta manera:

```
$export PATH = $PATH :/opt/java/bin
```

Ahora se configuran algunas alternativas que tiene la aplicación.

```
Alternatives --install /usr/bin/java java  
/opt/jdk1.6.0_29/bin/java 2  
alternatives --config java
```

Se presentará la siguiente pantalla:

Selection	Command
*+ 1	/usr/lib/jvm/jre-1.4.2-gcj/bin/java
2	/opt/jdk1.6.0_07/bin/java

Figura 3- 1. Instalación JDK.

Aquí se debe escoger el directorio donde se instaló el JDK. En este caso el marcado como 2.

Finalmente se deben configurar las variables JAVA_HOME y PATH mediante las siguientes sentencias

```
PATH=$PATH:/opt/java/bin
export PATH
```

Se reinicia el servidor

```
reboot
```

Luego se comprueba que se instaló con éxito usando el comando

```
java -versión
```

Se debe presentar la siguiente pantalla:

```
[root@localhost ~]# java -version
java version "1.6.0_20"
OpenJDK Runtime Environment (IcedTea6 1.9.10) (rhel-1.23.1.9.10.el5_7-i386)
OpenJDK Client VM (build 19.0-b09, mixed mode)
```

Figura 3- 2. Salida de comando java –versión.

3.4.5. Instalación del Servidor Web Apache.

Primero se recomienda revisar si este servidor ya se encuentra instalado en la distribución de Linux instalada, escribiendo el comando:

```
rpm -q httpd
```

En caso de no estar presente, se procede a instalar el paquete httpd en el sistema operativo ejecutando el siguiente comando:

```
yum install httpd
```

Finalmente se inicia el servicio Web:

```
service httpd start
```

3.4.6. Instalación del Servidor Wowza Media.

Antes de empezar la instalación se debe obtener el software WowzaMediaServer-3.0.3.rpm.bin y ubicarlo en el directorio usr/src/

```
cd /usr/src/
```

Se realiza la descarga del software desde el enlace ubicado en la página oficial de Wowza, escogiendo la opción de instalación Linux RPM.

Se procede con la instalación ejecutando los siguientes comandos:

```
chmod +x WowzaMediaServer-3.0.3.rpm.bin
```

```
./WowzaMediaServer-3.0.3.rpm.bin
```

Después de la instalación se necesita obtener una licencia válida para usar el software. Es preferible usar una licencia para desarrolladores, que se descarga desde <http://www.wowza.com/pro10s.php>.

Se llena el formulario con los campos requeridos e inmediatamente envían la licencia al correo electrónico que se registra en el formulario.

Wowza Media Server 3 Free Developer License

Fill in the fields below to obtain a Wowza Media Server 3 Free Developer License. The license key will be sent to the email address entered below.

Salutation: Ms. ▾

First Name *: Aurora

Last Name *: Cepeda

Company: espol

Street Address *: espol

Additional Address Information:

City *: Guayaquil

State **: ▾

ZIP / Postal Code *: EC901

Country *: Ecuador ▾

Phone:

Fax:

Email *: cepeda.tatiana@gmail.com

Retype email *: cepeda.tatiana@gmail.com

Yes, please keep me informed about important product features, notices and communications.

Note:
 The license key will be sent to the email address entered above.
 * Required fields.
 ** Required fields for USA, Canada, Japan and China addresses.

SUBMIT

Figura 3- 3. Formulario para obtener licencia de Wowza.

Se debe ubicar el directorio donde se instaló el software y se ejecuta el siguiente script.

```
cd /usr/local/WowzaMediaServer/bin
./startup.sh
```

Finalmente se ingresa la licencia, como se indica en [11].

```
[root@localhost bin]# ./startup.sh
Configure logging: file:///usr/local/WowzaMediaServer/conf/log4j.properties
Please enter license key number:
XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX[-XXXXXXXXXXXXX]
```

Figura 3- 4. Ingresar Número de Licencia de WowzaMediaServer.

3.4.7. Instalación de Flashphoner

3.4.7.1. Instalación de Aplicación Servidor

Primero, se debe instalar el puente de comunicación entre Asterisk y Wowza usando la aplicación Flashphoner Server. La aplicación se la descarga del

sitio web de Flashphoner. Luego se ubica el directorio `/usr/src/` y se descarga el software con el comando `tar -xvzf Flashphoner-1.0.5.1086-trunk.tar.gz`

Se ingresa al directorio recientemente creado

```
cd Flashphoner-1.0.5.1086-trunk/
```

A continuación, se ejecuta el script de instalación

```
chmod +x install.sh  
./install.sh
```

Luego se mostrarán los pasos de la instalación, se debe presionar la tecla enter para aceptar los términos de la aplicación.

Después la aplicación pedirá ingresar la dirección ip del servidor. En este caso se ingresa la siguiente dirección ip:

```
192.168.1.103
```

A continuación se debe ingresar la dirección local. Que en este caso es la misma del servidor:

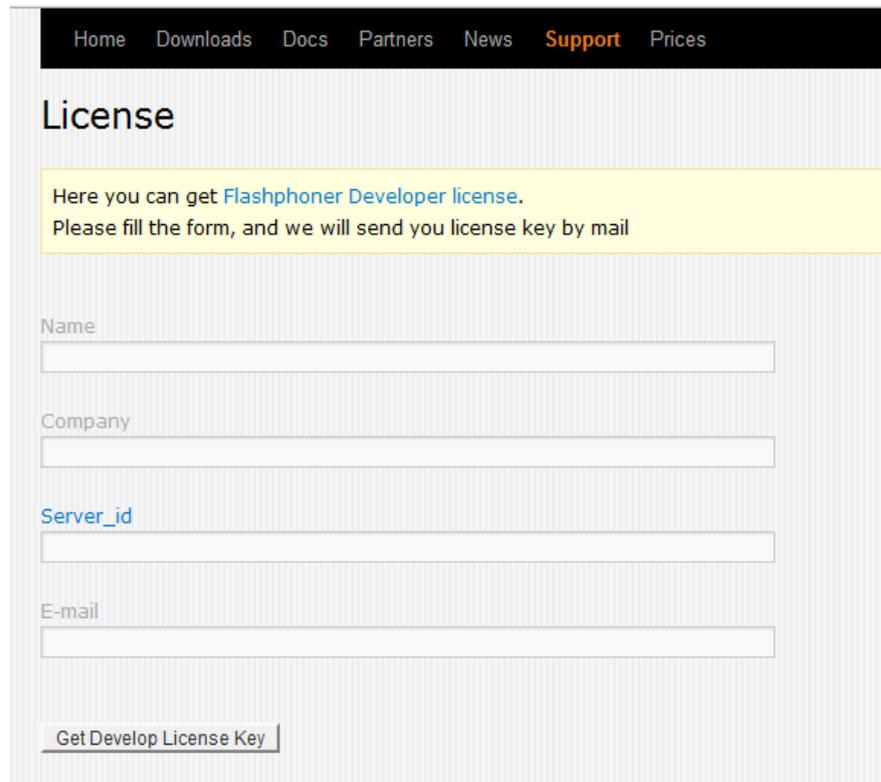
```
192.168.1.103
```

También se necesita informar a la aplicación sobre la ubicación del kit de desarrollo de java versión 1.6. En este caso el JDK se encuentra ubicado en siguiente directorio

```
cd /opt/java
```

Después de la instalación, se debe pedir una licencia del producto en la página oficial. Fijarse que se ha generado una clave llamada `server_id`, esta clave es importante y se debe copiar en un lugar seguro.

Se ingresan los datos en un nuevo formulario en la página para conseguir la licencia. Lo más importante es ingresar la clave server_id que fue generada previamente y una cuenta de correo electrónico válida para recibir la licencia.



The image shows a web page titled "License" with a navigation bar at the top containing links for Home, Downloads, Docs, Partners, News, Support, and Prices. Below the navigation bar, there is a yellow highlighted box with the text: "Here you can get [Flashphoner Developer license](#). Please fill the form, and we will send you license key by mail". The form itself consists of four input fields: "Name", "Company", "Server_id", and "E-mail". At the bottom of the form is a button labeled "Get Develop License Key".

Figura 3- 5. Formulario para Licencia Flashphoner

Después de ingresar la licencia, empezará la instalación y se debe reiniciar el servicio de Wowza.

Para este momento ya se tienen creados los archivos de configuración necesarios para hacer un puente de comunicación entre asterisk y el servidor rtmp.

3.4.7.2. Instalación de Aplicación Cliente

Ahora se debe instalar la aplicación cliente flash siguiendo los pasos que se detallan a continuación:

Primero se descarga la aplicación, después se debe copiar y pegar la aplicación en el directorio `/usr/src/`.

A continuación, se descomprime la aplicación.

```
tar -xvzf Flashphoner-client-1.0.180
```

Enseguida se ubica la carpeta que se creó al descomprimir la aplicación.

```
cd Flashphoner-client-1.0.180
```

Se otorga el permiso de ejecución al script `install.sh` y se inicia la instalación del cliente flash.

```
chmod +x install.sh
```

```
./install.sh
```

Se abrirá la consola de instalación donde debemos ingresar una ruta para ubicar la aplicación. Se recomienda utilizar el directorio que tiene configurado por default el servidor web Apache, el cuál es el siguiente: `/var/www/html/`

Finalmente se completa la instalación.

3.4.8. Instalación de Cliente Web

Ahora solo necesita tener un navegador web o cliente web en el computador que se usará como cliente. Se puede escoger el navegador que se desee, en este caso se escogió Mozilla firefox 3.6.

Seguidamente, es muy importante instalar el plugin de flashplayer versión 10 en el navegador web que se está utilizando, ya que con esta versión de plugin se han realizado las respectivas pruebas. La información explicada en esta sección fue tomada de [10].

3.5. CONFIGURACIONES

A continuación se muestran las direcciones Ip de la configuración de red usada en la implementación y pruebas de este proyecto.

Dirección IP	192.168.0.103
Mascara de red	255.255.255.0
Puerta de enlace	192.168.0.1

Tabla 3- 11. Direcciones IP de Servidor (Máquina Virtual)

Dirección IP	192.168.0.100
Mascara de red	255.255.255.0
Puerta de enlace	192.168.0.1

Tabla 3- 12. Direcciones IP de Cliente 1 (Máquina Virtual)

Dirección IP	192.168.0.101
Mascara de red	255.255.255.0
Puerta de enlace	192.168.0.1

Tabla 3- 13. Direcciones IP de Cliente 2 (Laptop)

3.5.1. Configuración de archivos de Asterisk

Después de haber instalado Asterisk solo necesitamos configurar algunos archivos según la aplicación que deseamos implementar, en nuestro caso necesitaremos de dos archivos: sip.conf y extension.conf

Estos archivos se encuentran en el directorio `/etc/asterisk/`.

3.5.1.1. Configuración del archivo SIP.CONF

El archivo sip.conf permite configurar todas las cuentas que usarán el protocolo de señalización SIP. Este archivo se ha configurado en dos

secciones. La sección general donde se establece la configuración general para todos los usuarios sip y la sección de extensiones donde se especifican los parámetros de configuración de manera individual a cada usuario o cliente SIP.

3.5.1.1.1. Sección General

A continuación se muestra la sección general del archivo sip.conf aquí se definen las configuraciones por defecto para todos los usuarios SIP. Más adelante, se explica los parámetros la configuración.

[general]
bindport=5060
binaddr=0.0.0.0
context=default
dtmfmode=info
videosupport=yes
disallow=all
allow=alaw
allow=h264
tos_sip=cs3
top_audio=ef
session-minse=90
session-expires=90
session-refresher=uac

Tabla 3- 14. Configuración de Sección General del archivo sip.conf

[general]	Etiqueta usada para la introducción de la configuración.
bindport=5060	Parámetro que coloca el puerto UDP 5060 en el estado “escuchando” para conectar las extensiones SIP con el servidor asterisk.
binaddr=0.0.0.0	Este parámetro indica que dirección de ip podrá conectarse al ordenador, en este caso cualquier dirección se conectará.
context=default	Nos indica el nombre del contexto definido en extensions.conf, el cual posee las configuraciones de las llamadas entrantes.
dtmfmode=info	Especifica el método por el cual se enviarán los tonos fuera de banda. Usamos info para usar el protocolo sip.
videosupport=yes	Este línea es muy importante porque nos permite habilitar el uso de las conexiones de video.
disallow=all allow=alaw allow=h264	Se deshabilitan todos los códecs para asegurarse de trabajar solo con los codecs alaw y h264.
tos_sip=cs3 top_audio=ef	Esta configuración es usada por la red para ofrecer un poco de calidad de servicio a pesar de que ésta este congestionada por el trafico.
session-minse=90	Este parámetro contiene el mínimo intervalo de actualización de la sesión en segundos.

session expires=90	Es el valor para que asterisk publica en la cabecera de la solicitud INVITE.
session-refresher=uac	Define que sea el uac (user agent client) quién defina el envío de la actualización de la sesión.

Tabla 3- 15. Listado de parámetros configurados en la sección general del archivo sip.conf.

3.5.1.1.2. Sección de extensiones

En esta sección se muestra la configuración que tiene cada usuario que usa el protocolo SIP.

Para este proyecto se han configurado varias cuentas desde la 2001 hasta la 2100, a continuación se presenta una configuración.

[2000]
type=friend
secret=2000
host=dynamic
canreinvite=no
dtmfmode=info
nat=yes

Tabla 3- 16. Configuración de extensiones del archivo sip.conf

[2000]	Numero de la extensión que en este caso representa al usuario o cliente sip.
--------	--

type=friend	Este parámetro indica el tipo de extensión a utilizar. Puede ser peer, user y friend, peer solo realiza llamadas, user solo recibe y friend es user + peers, es decir realiza y recibe llamadas.
secret=2000	Define el password para este usuario.
host=dynamic	Esta línea indica que este usuario puede tener configurada cualquier dirección ip para poder conectarse con el servidor asterisk.
nat=yes	Definimos nat porque el usuario puede estar conectado al servidor detrás de un firewall, si no será el caso podemos poner no.
canreinvite=no	Colocamos no para que asterisk haga el puente entre las dos extensiones que se quieren comunicar. Yes, si se quiere conectar directamente con la extensión de llamada.

Tabla 3- 17. Listado de parámetros configurados en una extensión del archivo sip.conf.

3.5.1.2. Configuración del archivo EXTENSIONS.CONF

Este archivo es el encargado de gestionar las llamadas entrantes y salientes para cada usuario, contiene el dialplan de asterisk que es el plan maestro de control o el flujo de ejecución de todas las operaciones según la información de [3], por esta razón es considerado como uno de los archivos más importantes del servidor. Se divide en secciones y contextos. Las secciones son [general] y [globals], y en ellas están las configuraciones estáticas y las definiciones. En cambio, los contextos son enteramente definidos por el

administrador y en ellos se encuentran las operaciones ejecutables del dialplan.

3.5.1.2.1. Sección General

[general]
autofallthrough=yes
clearglobalvars=no

Tabla 3- 18. Configuración de Sección General del archivo extensions.conf.

[general]	Establece las configuraciones generales que se aplica a todos los contextos.
autofallthrough=yes	Esta línea indica al servidor asterisk que debe colgar la llamada después de haber realizado todas las tareas, si se configura no se quedará esperando que otra extensión sea marcada.
clearglobalvars=no	De este modo cualquier variable global que definamos no se eliminarán después de hacer un reload al servidor.

Tabla 3- 19. Listado de parámetros configurados en Sección General del archivo extensions.conf.

3.5.1.2.2. Contexto

[default]
exten=> 2001,1,Dial(SIP/2001)
exten=> 2002,1,Dial(SIP/2002)
exten=> 2003,1,Dial(SIP/2003)
exten=> 2004,1,Dial(SIP/2004)
exten=> 2005,1,Dial(SIP/2005)

Tabla 3- 20: Configuración de contexto internos del archivo extensions.conf.

[default]	Contexto definido para el proceso de operaciones.
exten=> 2000,1,Dial(SIP/2000) exten=> 2001,1,Dial(SIP/2001) exten=> 2002,1,Dial(SIP/2002) exten=> 2003,1,Dial(SIP/2003)	Con estas configuraciones nos aseguramos que al marcar la extensión deseada asterisk nos comunique con el usuario correspondiente usando el protocolo SIP.Ej: Si marco la extensión 2001, llamaremos al usuario 2001

Tabla 3- 21. Configuración de contexto internos del archivo extensions.conf.

3.5.2. Configuración de archivos Wowza Media Server

Asterisk es una herramienta poderosa que se puede usar en conjunto con otros software Open Source, para este proyecto se escogió Wowza media Server debido a la fácil configuración que se usa para realizar el puente de comunicación entre el protocolo rtmp y el protocolo sip que maneja nuestro servidor asterisk.

Los archivos de configuración de este servidor se encuentran en el directorio /usr/local/WowzaMediaServer/conf/, a continuación se detallan los pasos para configurar el puente de comunicación obtenidos en [11].

Primero se debe ubicar el directorio:

```
cd /usr/local/WowzaMediaServer/conf/
```

Luego se deben editar los siguientes parámetros en el archivo flashphoner.properties

```
ip                =192.168.1.103
ip_local          =192.168.1.103
video_enabled     =true
```

Donde La dirección ip es la dirección externa del servidor rtmp, es decir si este servidor está ubicado solo en una red LAN, entonces la dirección ip es igual a la dirección ip_local, pero si éste servidor está detrás de un servidor nat o en una red wan, se debe poner la dirección externa de la red o del servidor nat. En este caso la dirección ip es igual a la dirección ip_local que es la ip del computador donde instalamos el servidor asterisk y Wowza. Otro parámetro importante es video_enabled el cuál debe estar activado con true para aceptar las conexiones de video que provienes del cliente flash.

Los demás parámetros se dejan con los valores que se crearon por default en el momento de la instalación. La configuración final del archivo es:

ip	=192.168.1.103
ip_local	=192.168.1.103
port_from	=30000
port_to	=31000
media_port_from	=31001
media_port_to	=32000
waiting_answer	=60
serial_number	=E42A0995C.....
user_agent	=Flashphoner/1.0
balance_header	=balance
cost_header	=cost
video_enabled	=true
domain	=
outbound_proxy	=
work_around	=false
log_level	=5
enable_context_logs	=false
rtp_activity_detecting	=true,5
force_h264_to_sorenson	=false

Tabla 3- 22. Configuración de archivo flashphoner.properties

El paso final para completar la configuración de nuestro puente de comunicación es editar el archivo falsphoner.xml ubicado en el directorio

`/var/www/html/flashphoner_client/`

Los parámetros importantes son los siguientes:

```
<rtmp_server>rtmp://192.168.1.103:1935</rtmp_server>
```

```
<register_required>true</register_required>
```

```
<application>phone_app</application>
```

rtmp_server

Contiene la dirección ip del servidor al que nos queremos conectar, y el puerto que utilizará. Por default el puerto es el 1935.

register_required

Si es true, obliga a registrar al cliente con el servidor VOIP que tengamos instalado, si es falso si debe utilizar el cliente de Flashphoner como un botón Click2Call.

application

Aquí debe ir el nombre de la aplicación del lado del servidor, pone_app es la que viene por defecto. Los demás parámetros los dejamos igual. La configuración final es:

```
<flashphoner>
    <rtmp_server>rtmp://192.168.1.103:1935</rtmp_server>
    <register_required>true</register_required>
    <application>phone_app</application>
    <video_width>176</video_width>
    <video_height>144</video_height>
    <check_validation_callee>true</check_validation_callee>
    <use_enhanced_mic>true</use_enhanced_mic>
```

```

<ring_sound></ring_sound>

<busy_sound></busy_sound>

<register_sound></register_sound>

<finish_sound></finish_sound>

</flashphoner>

```

Tabla 3- 23. Configuración de archivo flashphoner.xml.

3.6. LÓGICA DEL SISTEMA DE VIDEO LLAMADA

En esta sección se explica la lógica del sistema mediante el uso de los gráficos esquemáticos y diagramas de flujo.

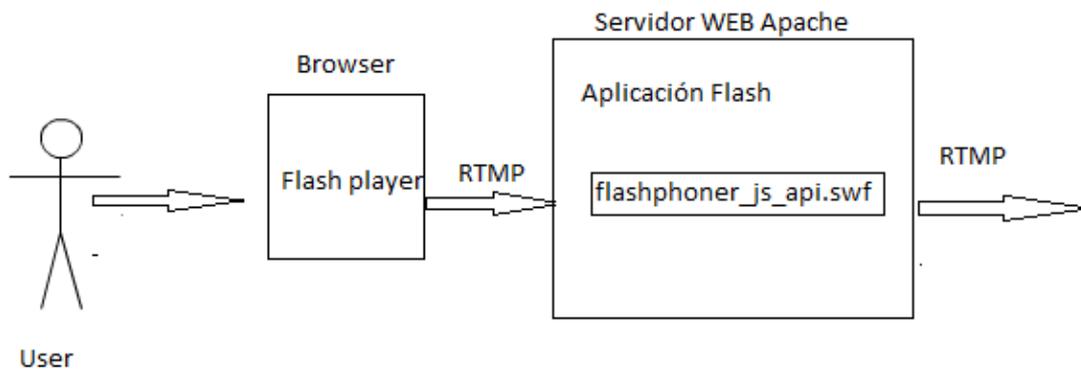


Figura 3- 6. Diagrama Esquemático de cliente Flash.

Primero la aplicación web será llamada por el cliente firefox, el cual cargará las páginas html, javascript y css, dentro de la misma existe un archivo muy importante llamado flashphoner_js_api.swf, que permitirá capturar los dispositivos de audio y video de un computador y enviar el flujo multimedia al servidor rtmp.

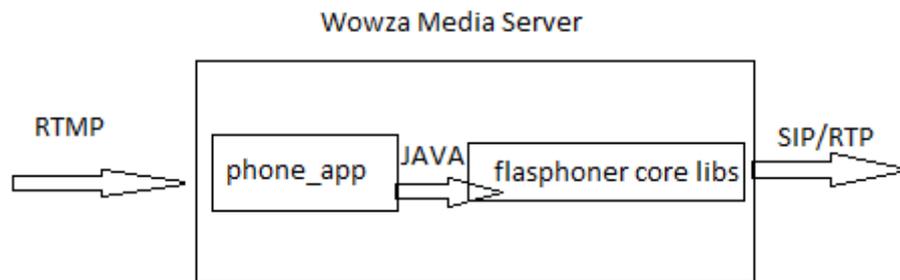


Figura 3- 7. Diagrama Esquemático del funcionamiento de Wowza.

Después el servidor Wowza recibirá el flujo multimedia y lo pasará a la aplicación `phone_app`, la cual contiene un conjunto de clases desarrolladas en java que permiten establecer el puente de comunicación entre los protocolos rtmp y SIP.

Luego, los mensajes serán enviados al servidor Asterisk el cual es el responsable de encaminar la llamada, interpretar los mensajes, registrar al usuario, en otras palabras el servidor asterisk maneja la parte más importante de la llamada interpretando las acciones del usuario mediante mensajes SIP que recibe. Los archivos que implementa el proyecto son: `sip.conf` y `extensions.conf` los mismos que fueron mencionados en la sección 3.5.1.

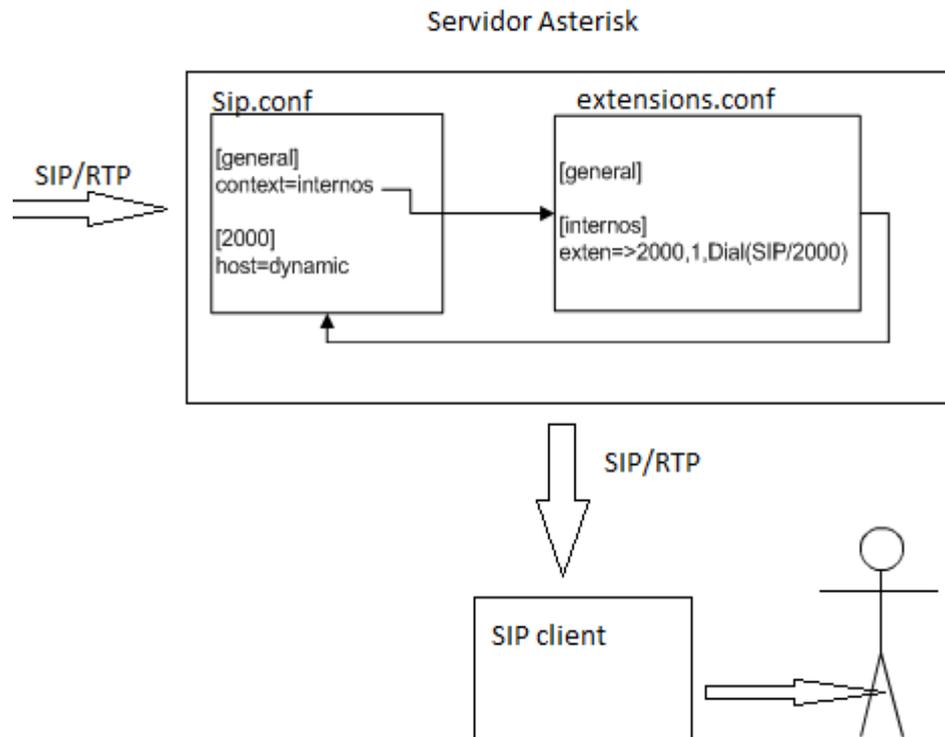


Figura 3- 8. Diagrama esquemático de Servidor Asterisk.

En la implementación del proyecto las acciones a realizarse son registrar en el servidor e invitar a un usuario para realizar una videoconferencia (llamar).

3.6.1. Acción registrar

Se realiza cuando el servidor recibe un mensaje SIP tipo "register". Primero el usuario debe registrarse en la central PBX de Asterisk. La lógica que utiliza el sistema para registrar un usuario es la siguiente:

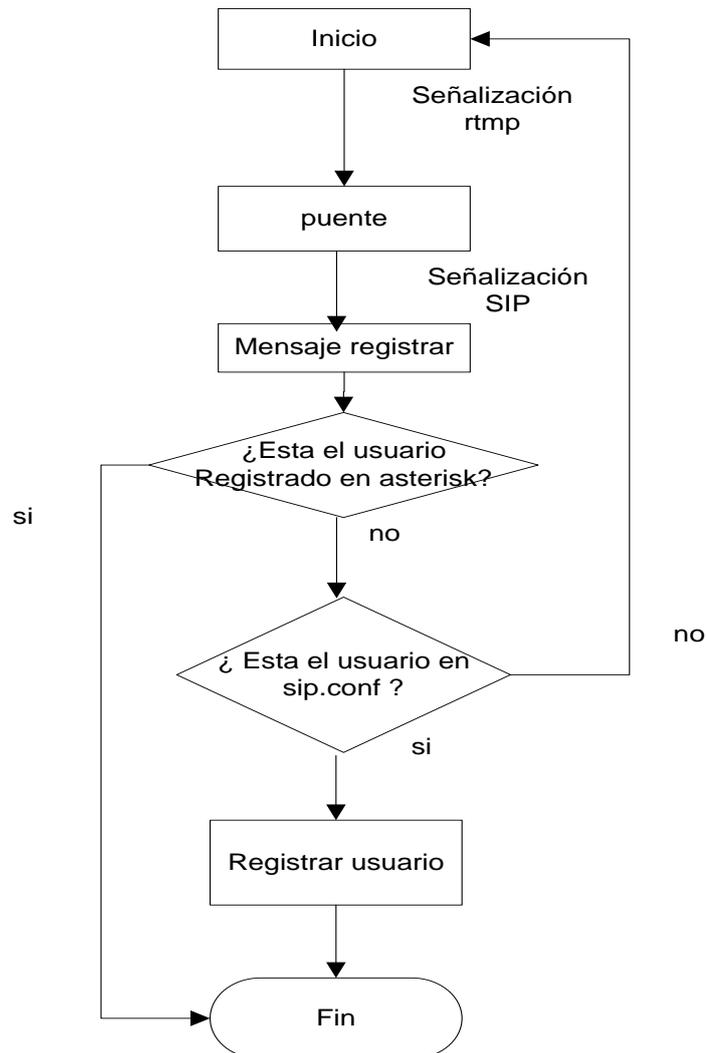


Figura 3- 9. Diagrama de Flujo del registro de un usuario.

Se observa claramente que después del puente de comunicación, asterisk recibe un mensaje sip para realizar el registro, este mensaje se llama “REGISTER”.

Cuando asterisk recibe el mensaje, lo interpreta y entiende que el cliente le solicita un registro. Si no está registrado, revisará el nombre de usuario en el archivo de configuración sip.conf. Si está correctamente configurado, lo registra, caso contrario volverá al inicio.

Si el usuario ya está registrado y vuelve a solicitar el registro, el servidor no hará nada y se mantendrá el estado registrado.

3.6.2. Acción llamar a usuario

La llamada se realiza solo si ambos usuarios (el que realiza la llamada y el que la recibe) se encuentran registrados en el servidor asterisk. El siguiente diagrama de flujo muestra la lógica usada por el sistema para realizar la llamada.

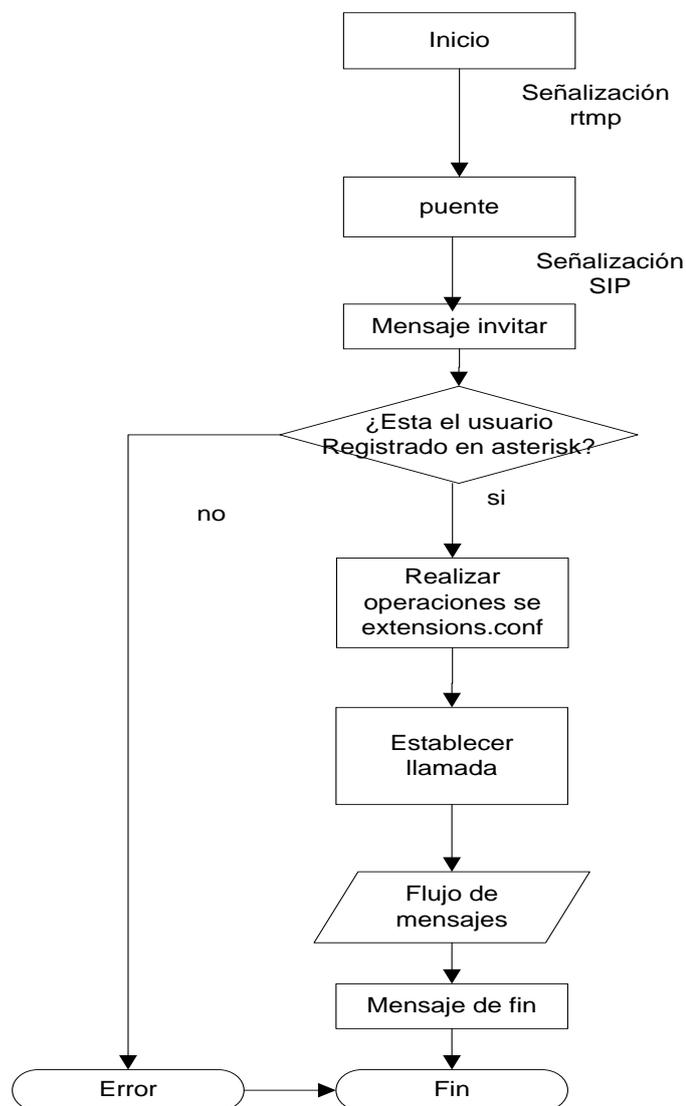


Figura 3- 10. Diagrama de Flujo de la llamada entre dos usuarios.

Después de enviar el mensaje de invitación, el servidor asterisk buscará al usuario invitado el mismo que deberá estar previamente registrado, si no lo está el servidor enviará un mensaje de error y no realizará la llamada, si lo está revisará el archivo extensions.conf y sip.conf, para ejecutar la sentencia:

```
exten=>2002,1,Dial(SIP/2002);
```

Donde **Dial** es una función propia de asterisk que permite realizar llamadas mediante el establecimiento de un canal de comunicación entre el solicitante y el usuario solicitado. Después de que el canal sea contestado el cliente que solicitamos enviará un mensaje de aceptación denominado "ACK" para establecer el canal de comunicación, en este punto ya estamos listos para intercambiar flujo multimedia.

Si se desea detener el canal o en otras palabras, se necesita colgar la llamada, se debe ejecutar la acción detener que será recibida en el servidor como un mensaje Bye y de esta forma se liberará el canal.

El diagrama de flujo final es el siguiente:

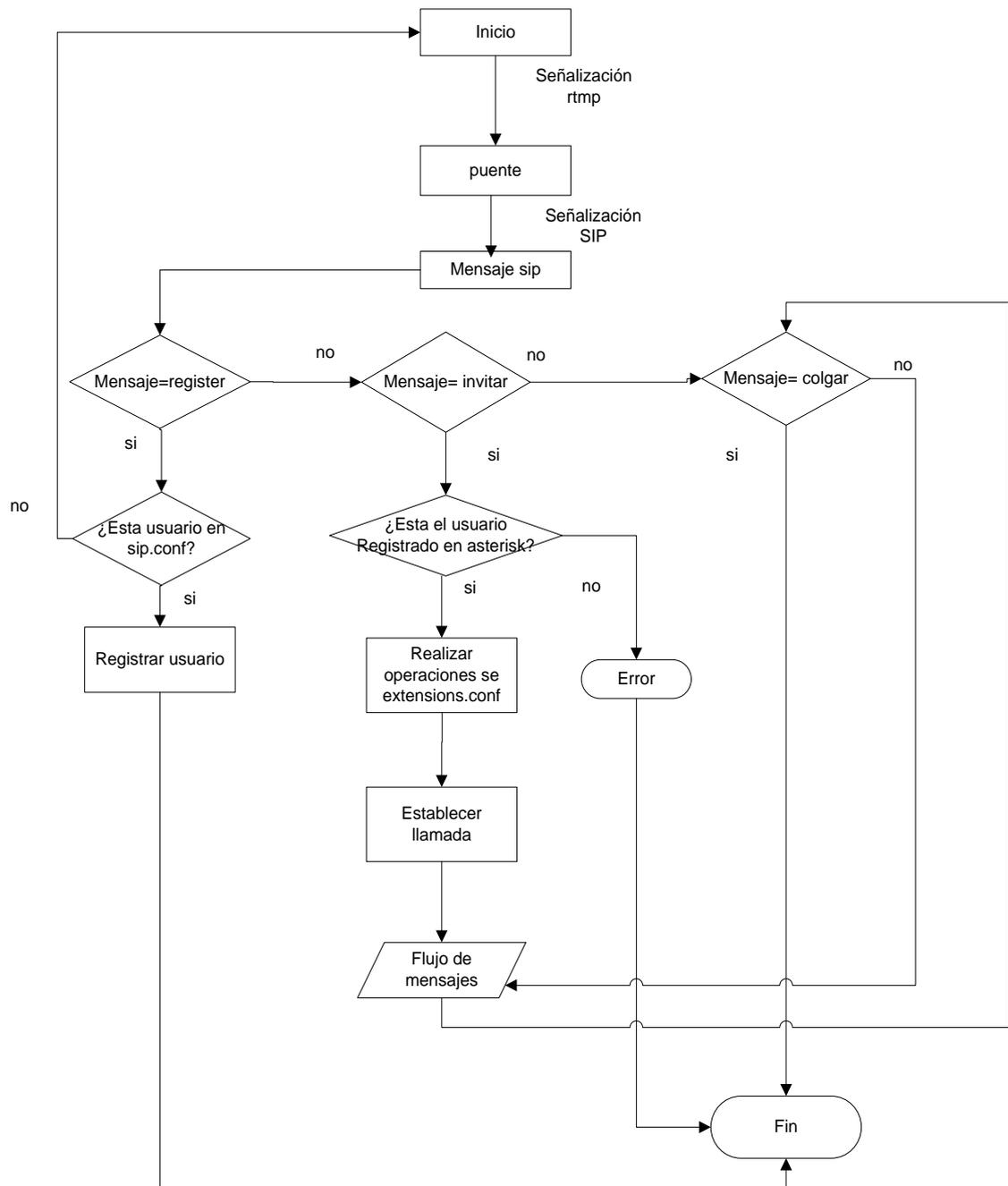


Figura 3- 11. Diagrama de flujo del funcionamiento del Sistema de Videollamada

CAPÍTULO 4

4. FUNCIONAMIENTO Y PRUEBAS

Después de instalar las aplicaciones necesarias y configurar todos los archivos mencionados en el capítulo 3, se muestra el funcionamiento del sistema y las respectivas pruebas realizadas siguiendo los pasos que se explican en esta sección.

4.1. FUNCIONAMIENTO DEL SISTEMA

4.1.1. Iniciando los Servicios

Primero se inician los servicios del servidor RTMP, VoIP y web respectivamente.

```
[root@localhost ~]#  
[root@localhost ~]# /etc/init.d/WowzaMediaServer start  
WowzaMediaServer: starting [ OK ]  
[root@localhost ~]# /etc/init.d/asterisk start  
Starting asterisk: [ OK ]  
[root@localhost ~]#  
[root@localhost ~]# /etc/init.d/httpd start  
Starting httpd: [ OK ]  
[root@localhost ~]#
```

Figura 4- 1. Iniciando los Servicios.

4.1.2. Ingresando a la Aplicación Web

Para ingresar a la aplicación web se debe abrir el navegador y escribir la dirección ip del servidor web, luego añadimos la ruta de la página principal de la aplicación:

http://192.168.0.103/flashphoner/flashphoner_client/PhoneJS.html

Se presiona entrar y aparecerá la siguiente pantalla:

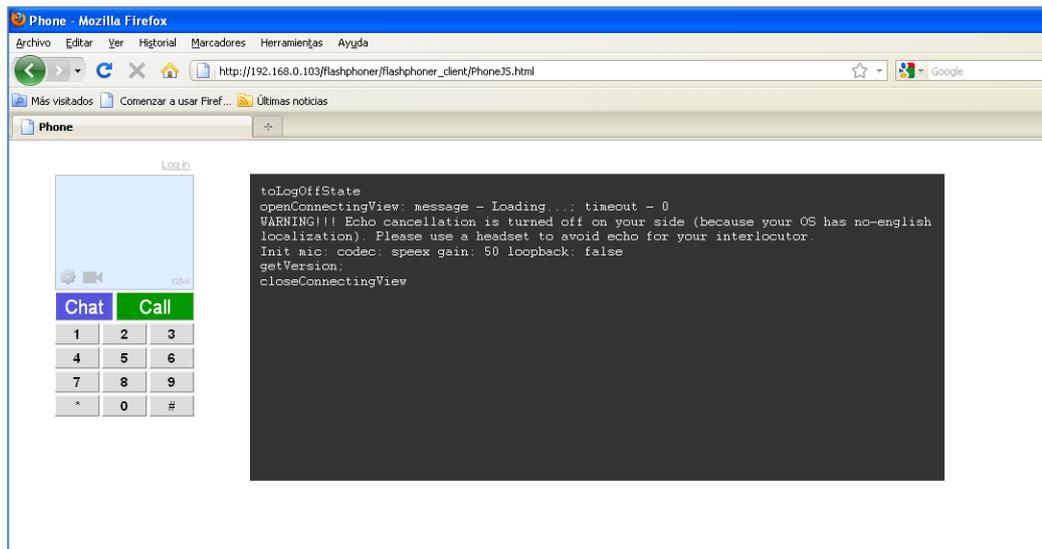


Figura 4- 2. Página principal de aplicación web.

El segundo paso es registrar el usuario en el servidor, damos click en Log in y llenamos los campos requeridos.

Figura 4- 3. Configuración de usuario en servidor asterisk.

Login._ Es el número de extensión del usuario.

Auth. Name._ Es el nombre del usuario.

Password._ Es la contraseña del cliente sip.

Server._ Es la dirección ip de nuestro servidor asterisk.

Port._ Es el número de puerto que utiliza el servidor para recibir peticiones de los clientes SIP.

Después de llenar los campos y presionar el botón Login, el usuario 2003 quedará registrado en el servidor asterisk como se observa en la figura 4.4. Para comprobar el registro es necesario revisar la consola del servidor VoIP la cuál mostrará los clientes sip que se encuentran conectados.

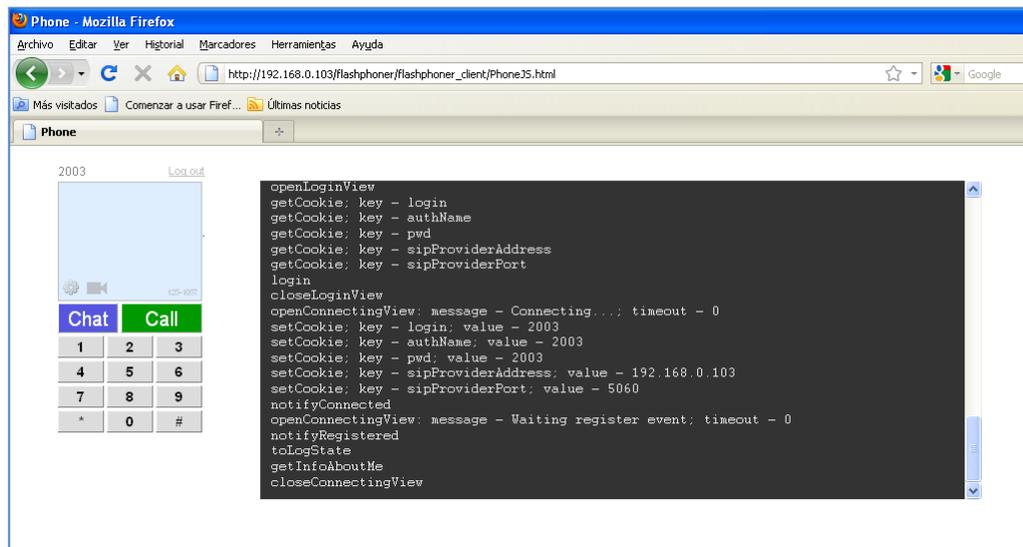


Figura 4- 4. Registro de usuario en servidor asterisk.

4.1.3. Configuración del Cliente SIP

El cliente SIP que se implementó se llama Eyebeam. A continuación se presenta la configuración del cliente para su correcto funcionamiento.

Luego de ejecutar el Eyebeam se busca la opción SIP Account Settings y se edita lo siguiente:

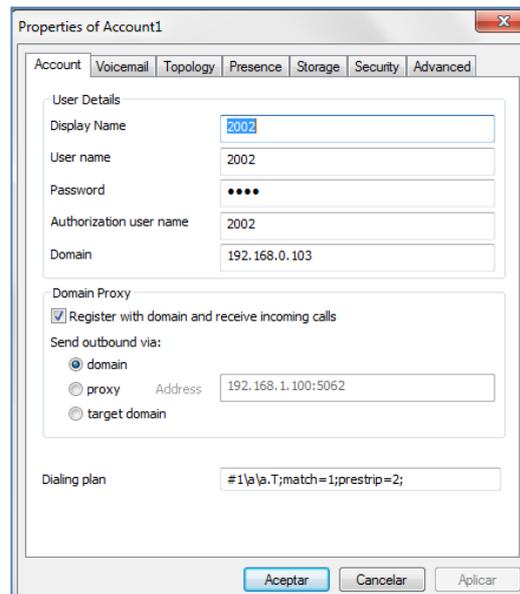


Figura 4- 5. Configuración de Eyebeam.

Luego se aplican los cambios y se presiona aceptar, en la figura mostrada el parámetro más importante es domain ya que contiene la dirección ip del servidor asterisk.

Después de realizar las respectivas configuraciones, el cliente SIP 2002 registrará en el servidor como se observa en la figura 4.6.



Figura 4- 6. Usuario registrado en Eyebeam.

4.1.4. Videollamada desde Cliente Web Flash

En esta sección se muestran los pasos para iniciar una videoconferencia desde un cliente web a un cliente SIP y a un cliente web.

Primero se debe iniciar el navegador y marcar la extensión del cliente que se desea llamar en este caso la 2002 que pertenece al cliente sip.

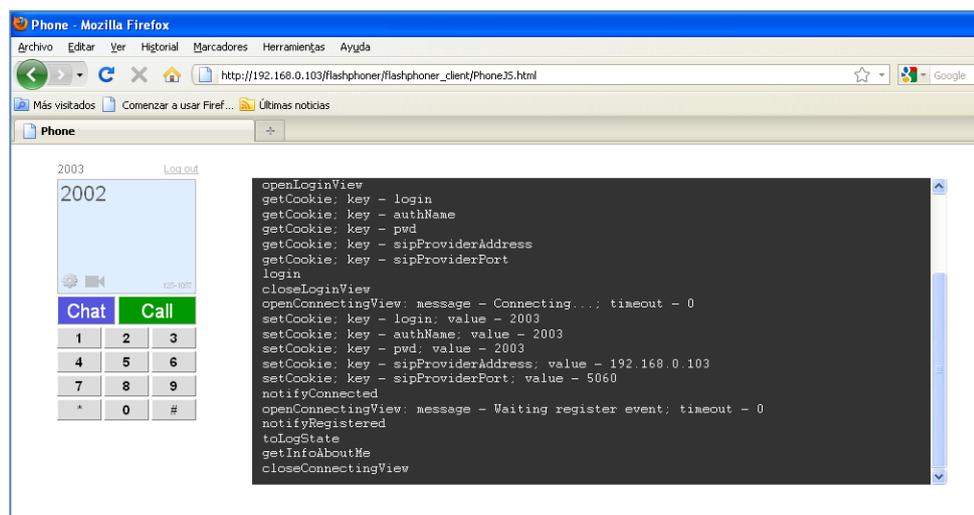


Figura 4- 7. Marcación desde cliente Web.

Después se escuchará un timbre que comprueba que el cliente Web está iniciando una petición de llamada y se observa lo siguiente en el browser:

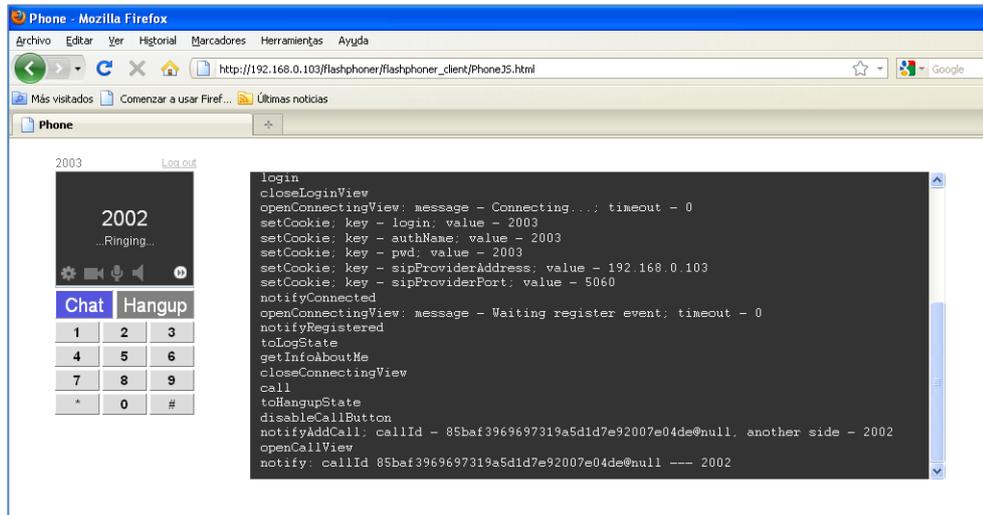


Figura 4- 8. Iniciando petición de llamada desde cliente Web.

Luego, el cliente SIP recibe un aviso de llamada entrante con el número de extensión de usuario que originó la llamada que es el 2003, como se puede observar en la figura 4.9.



Figura 4- 9. Llamada entrante desde Cliente Web Flash a Cliente SIP Eyebeam.

En el caso de que el cliente solicitado sea otro cliente web, este cliente recibirá un aviso de la llamada entrante como se muestra en la figura, en este ejemplo el usuario 2003 que está registrado desde una página web recibe una llamada del usuario 2007 que también está registrado desde una página web.

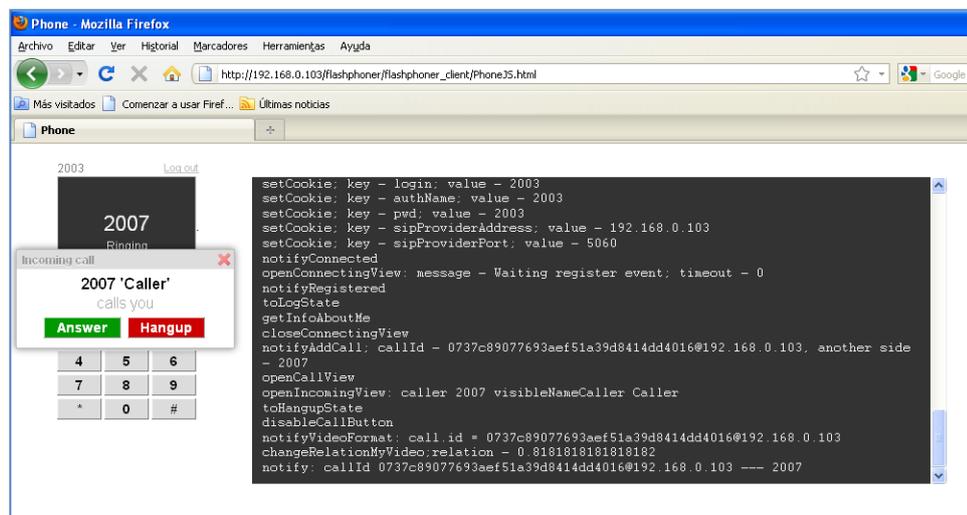


Figura 4- 10. Llamada entrante de Cliente Web Flash a Cliente Web Flash.

Finalmente una vez que el cliente solicitado acepte la conexión, se iniciará la videoconferencia y la transmisión de audio y video.

4.1.5. Videollamada desde Cliente SIP/ Eyebeam

Para iniciar una videoconferencia desde un cliente SIP a un cliente web y a otro cliente sip, se debe marcar la extensión del cliente al que queremos llamar en este caso la 2003 que pertenece al usuario 2003 como observamos en la figura 4.11.



Figura 4- 11. Marcación desde Cliente SIP Eyebeam.

Luego, se escuchará un timbre que comprueba que el cliente SIP está iniciando la petición de llamada. Observar la figura 4.12.



Figura 4- 12. Iniciando petición de llamada entrante desde Cliente SIP Eyebeam.

Después el cliente solicitado (2003) recibe un aviso de llamada entrante con el número de extensión del usuario que originó la llamada (2002).

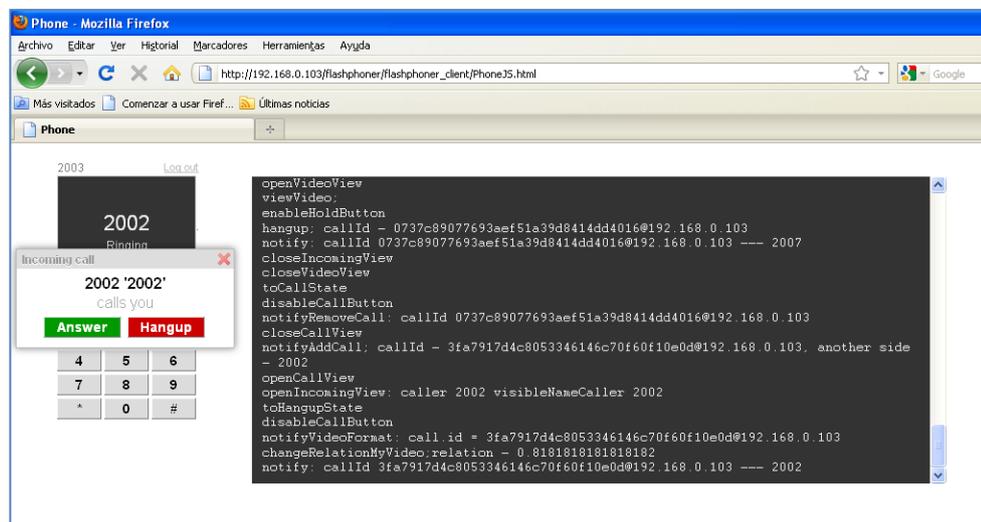


Figura 4- 13. Llamada entrante de Cliente SIP Eyebeam a Cliente Web Flash.

Finalmente se debe aceptar la llamada, para realiza la videoconferencia.

4.2. PRUEBAS REALIZADAS

Para comprobar el funcionamiento correcto de cada software instalado con el fin de transmitir voz y video se realizaron diversas pruebas que permiten asegurar que las aplicaciones y las configuraciones son correctas e idóneas para la implementación del proyecto.

4.2.1. Pruebas con el Servidor de Asterisk detenido

Primero, se comprobó que cuando el servidor Asterisk se encuentra detenido no se puede realizar la comunicación. Para ello debe a detener el servicio de asterisk con el comando que se muestra en la figura 4.14 y luego reiniciar el servidor RTMP como se muestra en la figura 4.15.

```
[root@localhost ~]# service asterisk stop
Stopping safe_asterisk: [ OK ]
Shutting down asterisk: [ OK ]
[root@localhost ~]#
```

Figura 4- 14. Comando para detener el servicio de Asterisk.

```
[root@localhost ~]# /etc/init.d/WowzaMediaServer restart
WowzaMediaServer: stopping
waiting for processes to exit
waiting for processes to exit [ OK ]
WowzaMediaServer: starting [ OK ]
[root@localhost ~]#
```

Figura 4- 15. Comando para reiniciar el servidor RTMP.

Para comprobar que el servicio de Asterisk se ha detenido, se ingresa a la consola del servidor con el comando:

```
asterisk -r
```

Y se observa que no es posible tener acceso a la consola porque conexión remota a asterisk esta deshabilitada.

```
[root@localhost ~]# asterisk -r
Asterisk 1.6.2.20, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for detail
s.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Unable to connect to remote asterisk (does /var/run/asterisk/asterisk.ctl exist?
)
[root@localhost ~]#
```

Figura 4- 16. Consola de Asterisk deshabilitada.

Luego de detener el servicio de Asterisk se intentó registrar a un usuario sin poder lograrlo, para comprobar lo anterior se ingresa al browser y se intenta registrar al cliente con el botón Login, la aplicación demora un poco en enviar una respuesta, ver figura 4.17 hasta que finalmente arroja un mensaje como se observa en la figura 4.18, el cual nos recomienda revisar las configuraciones de las cuentas SIP.

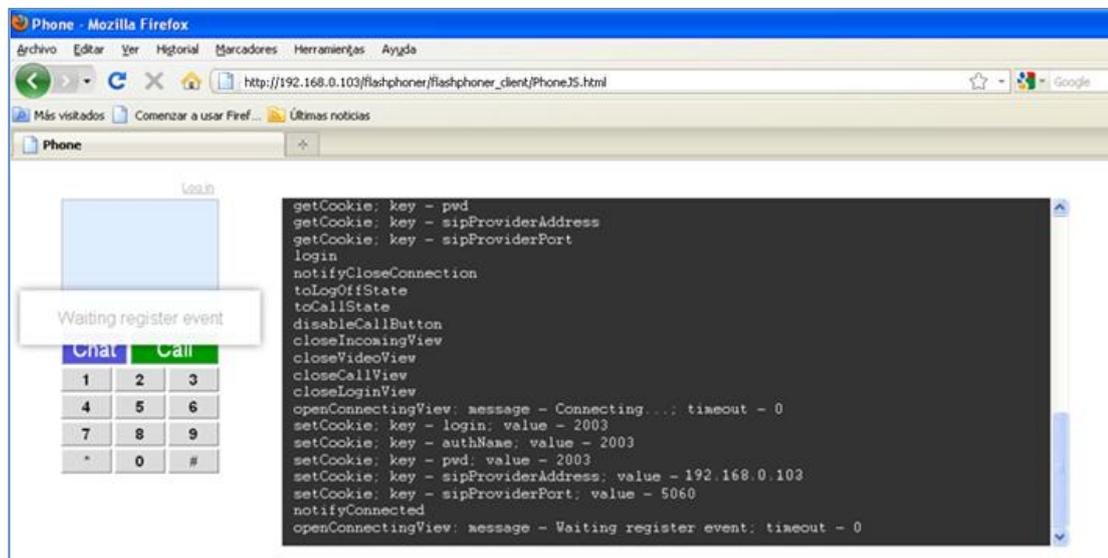


Figura 4- 17. Esperando la respuesta de registro del Cliente web en el servidor Asterisk.

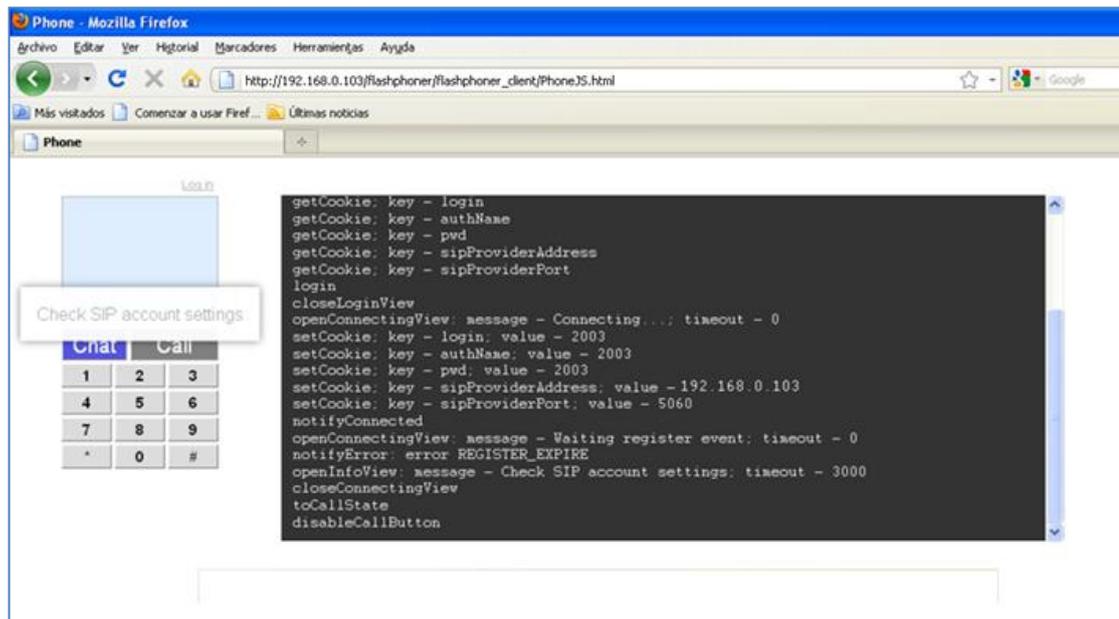


Figura 4- 18. Error de registro en cliente web.

Además se puede comprobar que existe un error de registro, esto se muestra en la consola que está en la página web, la cual indica el estado de la conexión con un error de expiración de registro. Esto se debe a que el servidor RTMP le envía la petición de registro al servidor Asterisk, pero el mensaje no llega a su destino debido a que los servicios de asterisk se han detenido.

Por otro lado, si se desea registrar un cliente SIP desde el Eyebeam, se obtendrá un error de registro como se observa en la figura 4.19. Esto sucede debido a que el cliente SIP nunca alcanza al servidor asterisk, ya que sus servicios están detenidos.



Figura 4- 19. Error de registro de cliente SIP debido a asterisk detenido.

4.2.2. Pruebas con el Servidor de Medios Wowza detenido

Ahora se comprobará que el sistema no funciona si se encuentra detenido el servidor RTMP, aunque el servicio de asterisk este corriendo. Para ello, se debe iniciar nuevamente asterisk y detener el servidor WowzaMedia, ver la figura 4.20.

```
[root@localhost ~]# /etc/init.d/asterisk start
Starting asterisk: [ OK ]
[root@localhost ~]# /etc/init.d/WowzaMediaServer stop
WowzaMediaServer: stopping
waiting for processes to exit
waiting for processes to exit [ OK ]
[root@localhost ~]#
```

Figura 4- 20. Comandos para iniciar asterisk y detener el servidor RTMP.

Se comprueba que los servicios de asterisk se han iniciado, para lograrlo se debe ingresar a la consola del servidor escribiendo:

```
asterisk -r
```

Como se puede observar en la figura 4.21 los servicios de asterisk están activos nuevamente.

```
[root@localhost ~]# asterisk -r
Asterisk 1.6.2.20, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.6.2.20 currently running on localhost (pid = 18585)
Verbosity is at least 3
localhost*CLI> █
```

Figura 4- 21. Consola de Asterisk habilitada.

Luego se comprueba que la aplicación que utiliza el protocolo rtmp no funciona correctamente, para ello el usuario debe ingresar a la página web y seguir los respectivos pasos de registro, obteniendo como resultado una falla en la conexión como lo indica la figura 4.22.

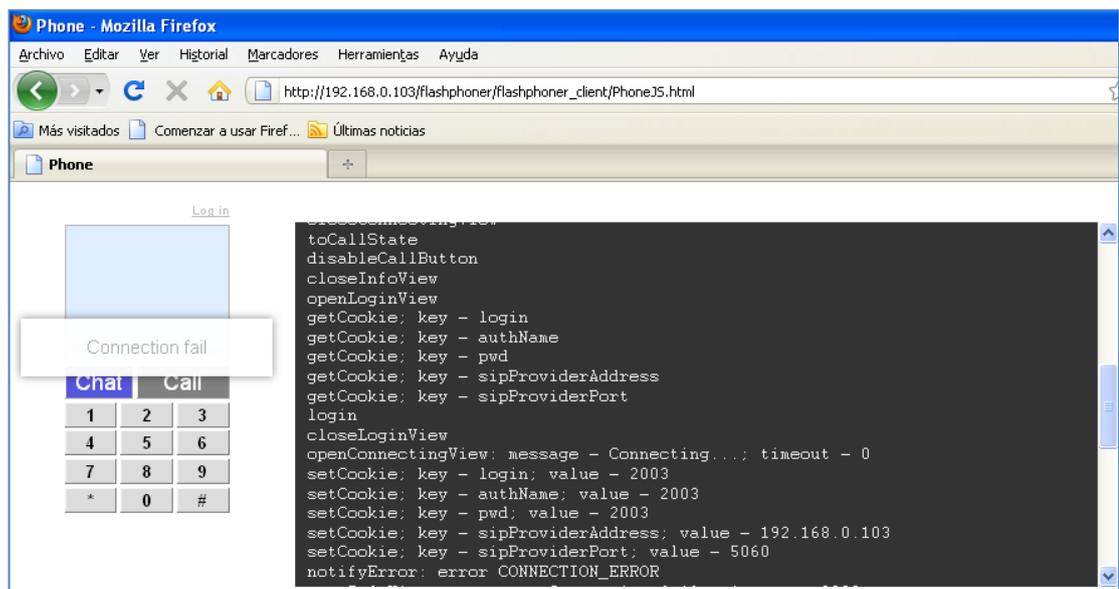


Figura 4- 22. Error de Conexión Fallida.

Este caso se diferencia de la prueba anterior, ya que la aplicación no espera por una respuesta de registro y de forma inmediata muestra el mensaje de error, debido a que la aplicación no puede conectarse con el servidor RTMP ya que éste se encuentra detenido.

Por otro lado, registrar un cliente SIP mediante la aplicación eyebeam si tiene éxito, debido a que el registro se realiza directamente con el servidor asterisk y no tiene que pasar por el servidor rtmp.



Figura 4- 23. Registro Exitoso de Cliente SIP Eyebeam.

4.2.3. Pruebas con el Servidor de Medios Wowza y Servidor Asterisk habilitados

Por último, se realizó una prueba con los dos servidores VoIP y RTMP activos para comprobar su funcionamiento.

Primero se debe escribir los comandos de la figura 4.24 para iniciar los servicios de los servidores y después se debe realizar los registros de los clientes.

```
[root@localhost ~]# /etc/init.d/WowzaMediaServer restart
WowzaMediaServer: not running
WowzaMediaServer: starting [ OK ]
[root@localhost ~]# /etc/init.d/asterisk restart
Stopping safe_asterisk: [ OK ]
Shutting down asterisk: [ OK ]
Starting asterisk: [ OK ]
[root@localhost ~]#
```

Figura 4- 24. Comandos para reiniciar Servidores asterisk y RTMP.

En la figura 4.25 se observa que dos usuarios 2002 y 2003 se han registrado correctamente en el servidor asterisk, el usuario 2002 se registro desde la aplicación web la cual se encuentra alojada en la dirección 192.168.0.101 y el usuario 2003 se registro mediante la aplicación Eyebeam que se encuentra instalado en dirección 192.168.0.103.

```
[root@localhost ~]# asterisk -r
Asterisk 1.6.2.20, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.6.2.20 currently running on localhost (pid = 5745)
Verbosity is at least 3
  -- Registered SIP '2003' at 192.168.0.103 port 30002
  -- Registered SIP '2002' at 192.168.0.101 port 50940
[Dec 29 17:31:49] NOTICE[5773]:                : Received SIP subscribe for peer without mailbox: 2002
localhost*CLI> █
```

Figura 4- 25. Registro exitoso de peers en Asterisk.

Después del registro exitoso se podrá realizar la videoconferencia, los pasos a seguir se explicaron en la sección 4.1.4 y 4.1.5. Para verificar todos los mensajes que recibe y envía el servidor para poder realizar la videoconferencia se debe observar la consola de asterisk, la figura 4.26 muestra como el usuario 2003 envía la invitación de llamada al usuario 2002 en la fase de timbrado.

```

Connected to Asterisk 1.6.2.20 currently running on localhost (pid = 14311)
Verbosity is at least 3
  -- Remote UNIX connection
  == Using SIP RTP TOS bits 184
  == Using SIP RTP CoS mark 5
  == Using SIP VRTP CoS mark 6
  -- Executing [2002@default:1] Dial("SIP/2003-00000002", "SIP/2002") in new s
tack
  == Using SIP RTP TOS bits 184
  == Using SIP RTP CoS mark 5
  == Using SIP VRTP CoS mark 6
  -- Called 2002
  -- SIP/2002-00000003 is ringing
localhost*CLI>

```

Figura 4- 26. Registro exitoso de peers en asterisk.

Luego se observa en la figura 4.27, que la llamada es aceptada por el usuario solicitado, y que existe flujo rtp, lo cual indica que la videoconferencia se está realizando.

```

  -- SIP/2002-00000003 answered SIP/2003-00000002
[Dec 29 17:37:14] NOTICE[6724]: : : Unknown RTP codec 126 r
eceived from '192.168.0.101'
[Dec 29 17:37:14] NOTICE[6724]: : : Unknown RTP codec 126 r
eceived from '192.168.0.101'
[Dec 29 17:37:14] NOTICE[6724]: : : Unknown RTP codec 126 r
eceived from '192.168.0.101'
[Dec 29 17:37:24] NOTICE[6724]: : : Unknown RTP codec 126 r
eceived from '192.168.0.101'
[Dec 29 17:37:34] NOTICE[6724]: : : Unknown RTP codec 126 r
eceived from '192.168.0.101'
localhost*CLI>

```

Figura 4- 27. Aceptación y Establecimiento de la Llamada.

En la figura 4.28 y 4.29 se observa la videoconferencia del lado del cliente web y del cliente Eyebeam respectivamente.

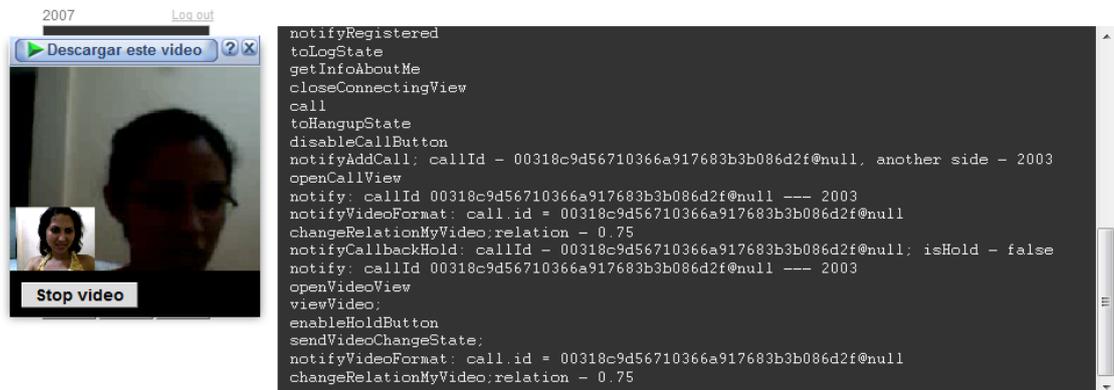


Figura 4- 28. Videollamada del lado del cliente Web Flash.



Figura 4- 29. Videollamada del lado del cliente SIP Eyebeam.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- 1) La implementación del sistema de videoconferencia, ayudó a comprobar que Asterisk puede utilizar perfectamente el protocolo rtmp usando un puente de comunicación entre el protocolo SIP, lo que lo convierte en una herramienta muy codiciada por muchos desarrolladores de aplicaciones web que usan flash para integrar sus aplicaciones al protocolo SIP y usar un servidor de código abierto, eficaz y gratuito como Asterisk.
- 2) Además se confirmó que el protocolo SIP está bien manipulado por el servidor usando todos los mensajes del rfc3261 en el módulo chan_sip y que el protocolo RTMP funciona perfectamente en el software Wowza Media el cual también es de código abierto.
- 3) Por otra parte, se pudo ratificar que asterisk no es solo una herramienta eficiente por todas las funciones que posee, sino que además es poderosa por su integración con otros programas tales como el servidor rtmp Wowza Media y el cliente flashphoner los cuales en conjunto pueden formar sistemas grandes y estables como el cliente de videoconferencia web implementado, el cual funciona perfectamente en una red lan cableada y una red lan inalámbrica.

- 4) Adicionalmente se comprobó que para realizar un sistema robusto y a la vez económico se debe utilizar como base el servidor asterisk debido a su completa gratuidad y constante mejoras.
- 5) Se concluye también que se han cumplido los objetivos propuestos, utilizando los conocimientos adquiridos a lo largo de la carrera en las materias de Fundamentos de Linux y Fundamentos de redes de datos.

Recomendaciones

- 1) Para implementar esta solución tecnológica en un ambiente de pruebas se recomienda utilizar una licencia gratuita de 30 días del servidor Wowza Media, la cual contiene todas las funcionalidades y complementos.
- 2) Si se desea ofrecer un servicio con esta solución, se recomienda utilizar la licencia mensual del servidor y de todos los complementos, ya que los costos son más económicos que la licencia indefinida, pero tenemos más opciones en el uso de canales y podemos cambiar de servidor RTMP en un futuro.
- 3) Se recomienda Utilizar la versión 1.6.2 de Asterisk debido a la compatibilidad que tiene el software con el conjunto de aplicaciones utilizadas, ya que en la versión 1.8 la forma de enviar los mensajes SIP varía, aunque ya se están desarrollando nuevos parches para esta versión todavía no han sido probados por muchos desarrolladores.

- 4) Del mismo modo, se debe escoger las mismas versiones del programa Flashphoner y Wowza Media que hemos utilizado, para un buen funcionamiento del proyecto.

- 5) Por último, en el caso de que la aplicación no se encuentre funcionando adecuadamente se debe reiniciar todos los servicios y volver a intentar, seguro así funcionará.

GLOSARIO

Digium.- Compañía especializada en el desarrollo del hardware de PBX y del software de telefonía de Asterisk.

H.323.- protocolo VoIP usado para la comunicación multimedia, es más complejo que SIP, por este y otros motivos H.323 actualmente está siendo desplazado por el protocolo de señalización SIP.

chan_sip.- Módulo perteneciente al software de Asterisk para el manejo y gestión del protocolo SIP.

JainSIP.- Librería utilizada para crear la pila SIP mediante código de programación en lenguaje JAVA.

Canal.- Ruta o camino por el cual se transmite la información desde un computador a un periférico determinado.

Instancia.- Es una copia de la versión ejecutable del sistema que se ha guardado en la memoria del computador.

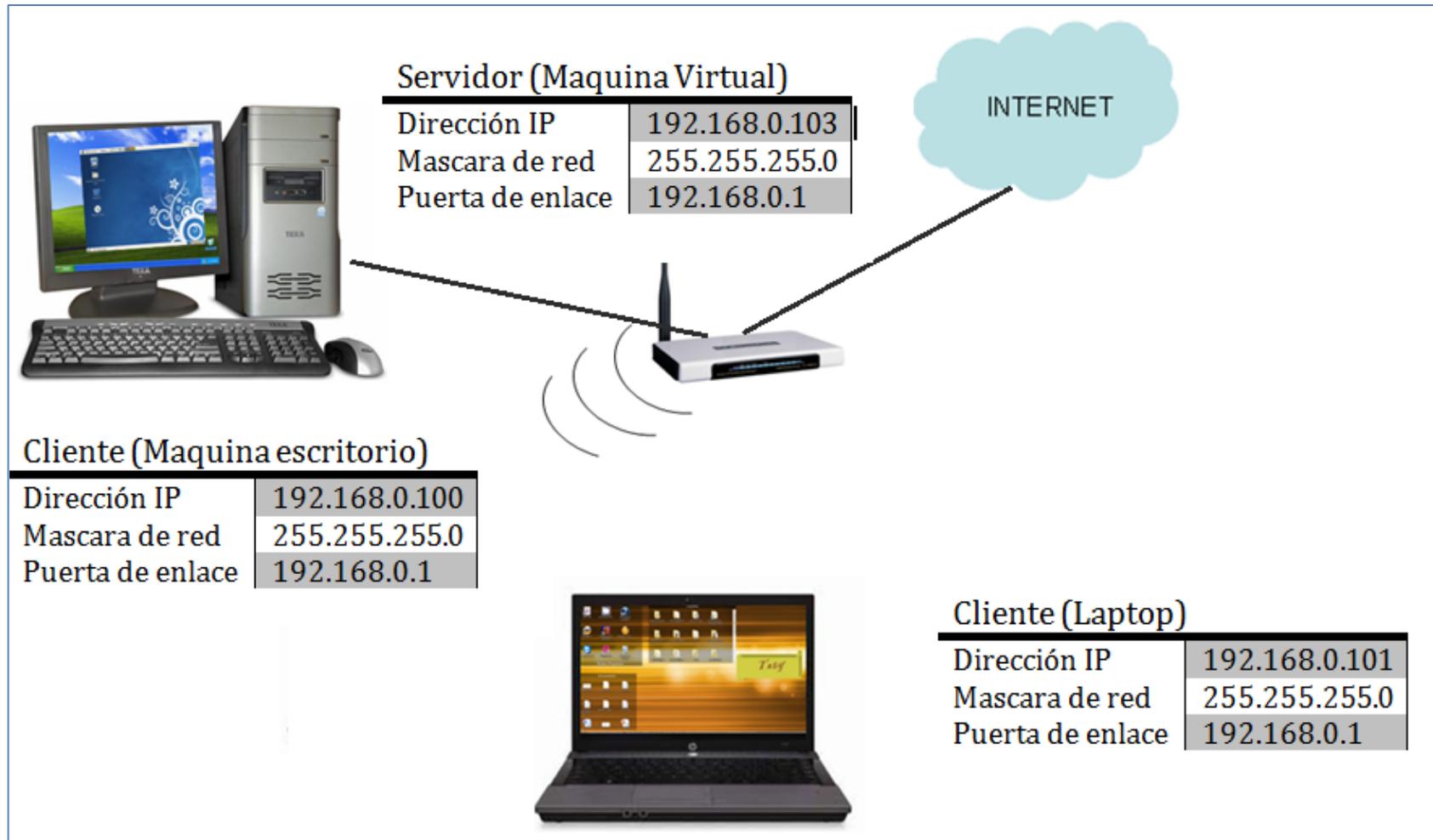
Wowza Transcoder AddOn.- Es un complemento que transforma los códecs de la señal de video de entrada al códec de video H.264 soportado por los protocolos de transmisión multimedia Flash RTMP, HTTP.

Wowza nDVR AddOn.- Es un complemento que almacena el flujo multimedia en un formato normalizado para acceder a éste con las características de pausa, retroceder y continuar. Además minimiza los requisitos de almacenamiento en red y simplifica el flujo de trabajo de entrega de todas las pantallas.

Wowza DRM AddOn.- Permite el intercambio simultáneo de claves de seguridad con múltiples plataformas DRM como Verimatrix VCAS y Microsoft PlayReady.

ANEXO

Diagrama de Red Utilizado en la Implementación del Proyecto



BIBLIOGRAFÍA

- [1] *Jim Van Meggelen, Jared Smith, y Leif Madsen, "Asterisk. The Future of Telephony" 2nd Edition, O' Reilly, 2007, cap. 3, pp 37-62.*
- [2] *Gonzalo Camarillo, "SIP Demistified", McGraw-Hill, 2002, cap. 5, pp 130-142.*
- [3] *David Gomillion y Barrie Dempster, "Construyendo Sistemas telefónicos con Asterisk", PACKT Publishing edition, Abril 2007, cap 5.*
- [4] *Asterisk-ES: Comunidad de usuarios Asterisk Español, "Introducción a Asterisk",
http://comunidad.asterisk-es.org/index.php?title=Introduccion_a_Asterisk,
Octubre 20 2011.*
- [5] *Blog de miguel, "FXS-FXO", <http://blog.pucp.edu.pe/item/7490>, Noviembre 11 2011.*
- [6] *VozTelecom, "Modelo de configuración básica de Asterisk 1.2 con la plataforma de Voztelecom",
http://www.voztele.com/esp/productos_servicios_voip/linea_ip_oigaa_direct/doc/Asterisk-1.2-VozTelecom.pdf, Agosto 15 2011.*
- [7] *Digium Inc., "Asterisk Feature List", <http://www.asterisk.org/features>,
Septiembre 20 2011.*
- [8] *Voip-Info.org, "A reference guide to all things VOIP", <http://www.voip-info.org>,
Septiembre 20 2011.*
- [9] *Asterisk CR, "Por qué encontramos bugs en versiones estables",
<http://www.asteriskcr.com/?p=697>, Abril 24 2011.*
- [10] *Flashphoner, "Flashphoner Documentation", <http://docs.flashphoner.com>,
Septiembre 11 2011.*

- [11] Wowza Media Systems, "Wowza Media Server 3: Benefits, AddOns and Features", <http://www.wowza.com/media-server>, 2011-2012.
- [12] Asterisk Guide, "El Protocolo SIP",
http://www.asteriskguide.com/mediawiki/index.php/El_Protocolo_SIP,
Febrero 17 2010.
- [13] Voip en Español, "SIP", <http://voip.megawan.com.ar/doku.php/sip>,
Noviembre 25 2011.
- [14] AsterisKGuru, "Tutorials", <http://www.asteriskguru.com>, Abril 25 2012.
- [15] Audio-Video Transport Working Group, H. Schulzrinne, GMD Fokus, S. Casner, Precept Software, Inc., R. Frederick, Xerox Palo Alto Research Center, V. Jacobson y Lawrence Berkeley National Laboratory, Network Working Group, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, IETF, Enero 1996.
- [16] J. Rosenberg, dynamicsoft, H. Schulzrinne, Columbia U., G. Camarillo, Ericsson, A. Johnston, WorldCom, J. Peterson, Neustar, R. Sparks, dynamicsoft, M. Handley, ICIR, E. Schooler y AT&T, Network Working Group, "SIP: Session Initiation Protocol", RFC 3261, IETF, Junio 2002.
- [17] Steven Sokol, "Say Hello To Asterisk",
http://www.asterisk.org/files/say_hello_to_asterisk.pdf, pag.5, Noviembre 26 2011.
- [18] 3CX, "¿Qué significan los términos FXS y FXO?", <http://www.3cx.es/voip-sip/fxs-fxo.php>, Noviembre 26 2011.

[19] *Packetizer, Jan Mastalir, DiS, "Understanding SIP-Based VoIP",*

http://www.packetizer.com/ipmc/sip/papers/understanding_sip_voip/,

Noviembre 26 2011.