

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



**“Implementación de una herramienta software
para la gestión de una central telefónica
basada en una computadora personal”**

PRESENTADO POR:

WILLIAN OVIDIO REYES MEDRANO

PARA OPTAR AL TITULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, DICIEMBRE DE 2008

UNIVERSIDAD DE EL SALVADOR

RECTOR :

MSC. RUFINO ANTONIO QUEZADA SÁNCHEZ

SECRETARIO GENERAL :

LIC. DOUGLAS VLADIMIR ALFARO CHÁVEZ

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO :

ING. MARIO ROBERTO NIETO LOVO

SECRETARIO :

ING. OSCAR EDUARDO MARROQUÍN HERNÁNDEZ

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR :

ING. JOSÉ WILBER CALDERÓN URRUTIA

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

**“Implementación de una herramienta software
para la gestión de una central telefónica
basada en una computadora personal”**

Presentado por :

WILLIAN OVIDIO REYES MEDRANO

Trabajo de Graduación Aprobado por :

Docente Director :

Dr. CARLOS EUGENIO MARTÍNEZ CRUZ

San Salvador, Diciembre de 2008

Trabajo de Graduación Aprobado por:

Docente Director :

Dr. CARLOS EUGENIO MARTÍNEZ CRUZ

Dedicatoria

A mis padres y hermanos, por creer en mi y apoyarme desde el inicio hasta la culminación de mi carrera, son ellos realmente los únicos que han hecho aportes invaluableles en mi crecimiento académico y personal.

Agradecimientos

Primeramente agradecerle a Dios por darme vida, salud y sabiduría todos estos años, los cuales fueron elementos fundamentales para finalizar mis estudios.

A mi querido y amado hermano Carlos Alfredo Reyes, a quien estaré eternamente agradecido, por haber sacrificado muchos logros personales sin importar lo valioso que éstos fueran, con la finalidad de coronar mi carrera universitaria.

Infinitos agradecimientos a mis adorados padres Emigdio Reyes y Milagro Medrano que junto a mis hermanos Wilfredo, Raúl, Zulma y Carlos, me brindan la mejor familia que cualquier persona desearía tener.

Al futuro de la familia Reyes Medrano, mis sobrinos Jennifer, Henry, Amália, Samántha, Juan José, Raúl Eduardo y mi querida cuñada Nohemi Privado.

Mi novia Claribel Renderos, gracias por ser tan comprensible y cariñosa estos tres años que hemos compartido juntos.

A todos mis amigos y compañeros con quienes compartí momentos inolvidables en la Universidad de El Salvador: Roberto Romero, Higinio Abel, Arquímedes Castellón, Oscar Vasquez, Joel Rivera, Juan Carlos López, Rene Mauricio Jerónimo, Rene Vides y Carlos Molina.

Amistades sinceras y queridas como Elmer Bonilla, Cesar Benavidez, Nimrod Joel, Alejandro Antonio, Mario Valdes y Edgar Agustín.

A mi asesor de Tesis el Dr. Carlos Eugenio Martínez, quien estuvo dispuesto a trabajar en todo momento y me guió de la mejor forma hasta la finalización del presente trabajo.

Y por último pero no menos importante a toda la comunidad desarrolladora de software libre, ya que sin sus aportes, este trabajo no se hubiera podido llevar a cabo.

Resumen

Debido al rápido crecimiento de la VoIP y a las ventajas que ésta ofrece, se implementará una central telefónica de VoIP en una computadora personal que utiliza el sistema operativo Debian GNU/Linux. La cual permitirá realizar llamadas telefónicas dentro de toda la red LAN. Para lograr esto se hará uso de Asterisk, un software libre que hace posible, que una computadora personal funcione como PBX.

Asterisk dispone de versiones que permiten su configuración por medio de entornos gráficos. No obstante se construirá una herramienta de configuración personalizada que facilita la administración de la central telefónica VoIP por medio de un navegador web. Se echará mano exclusivamente de software libre. La centralita permitirá realizar llamadas telefónicas dentro del entorno de una red de área local y se ampliará su cobertura utilizando el estándar IEEE 802.11 conocido como Wi-Fi.

ACRÓNIMOS

VoIP:	Voice over Internet Protocol.
IAX:	Inter-Asterisk eXchange.
SIP:	Session Initiation Protocol.
RTP:	Real-time Transport Protocol.
GSM:	Global System for Mobile communications.
ATA:	Analog Telephone Adapter.
TOS:	Type Of Service
FXO:	Foreing eXchange Office.
FXS:	Foreing eXchange Station.
NAT:	Network Address Traslators.
DHCP:	Dynamic Host Configuration Protocol.
PSTN:	Public Switched Telephone Network.
UIT/ITU:	Unión Internacional de Telecomunicaciones/ International Telecommunication Union.
IETF:	Internet Engineering Task Force.
IEEE:	Institute of Electrical and Electronics Engineers.
WI-FI:	Wireless Fidelity.
PBX:	Private Branch eXchange.
SS7:	Sistema de Señalización 7.
CODEC:	Codificador-DECodificador.
DTMF:	Dual Tone MultiFrecuency.
SCCP:	Skinny Client Control Protocol.
MGCP:	Media Gateway Control Protocol.
ANT:	Another Neat Tools.
SQL:	Structured Query Lenguage.
HQL:	Hibernate Query Lenguage.
API:	Application Programming Interface.
JDBC:	Java DataBases Connectivity.
JSP:	JavaServer Pages.
JDK:	Java Development Kit.
HTML:	HiperText Markup Lenguage.
XML:	eXtensible Markup Lenguage.
W3C:	World Wide Web Consortium.
CLI:	Command Line Interface.
ALSA:	Advanced Linux Sound Architecture.
PYME:	Pequeña y Mediana Empresa.
EIE.	Escuela de Ingeniería Eléctrica.

ÍNDICE GENERAL

CAPITULO I. INTRODUCCIÓN.....	1
1.1. Interés de la investigación.....	1
1.2. Motivación.....	2
1.3. Objetivos de la tesis y organización.....	3
CAPITULO II. CONCEPTOS INVOLUCRADOS EN LA HERRAMIENTA.....	5
2.1. VoIP.....	5
2.1.1. ¿Cómo Funciona? y ¿qué Tecnologías Soporta?.....	6
2.1.2. Protocolos de Señalización	8
2.1.2.1. H.323.....	8
2.1.2.2. SIP.....	8
2.1.2.3. IAX.....	9
2.1.3. Ventajas y Desventajas de VoIP.....	9
2.2. PBX.....	10
2.3. Debian, El Sistema Operativo.....	10
2.4. Asterisk.....	11
2.5. Base de Datos Mysql.....	13
2.6. Servidor Web Tomcat.....	14
2.7. Lenguajes de Programación.....	14
2.7.1. HTML.....	14
2.7.2. XML.....	15
2.7.3. Java.....	15
2.7.4. JavaServerPages (JSP).....	15
2.7.5. Script.....	16
2.8. Hibernate.....	16
2.9. ANT.....	17
2.10. Enlace Wi-Fi.....	17
2.11. Conclusiones.....	17
CAPITULO III. INSTALACIÓN DEL SOFTWARE DE DESARROLLO.....	18
3.1. Obteniendo Debian.....	18
3.2. Instalación de Asterisk.....	18
3.3. Instalación de JDK.....	21
3.4. Instalación de Tomcat	22
3.5. Obteniendo JDBC.....	23
3.6. Instalación de Mysql.....	24
3.7. Obteniendo Hibernate.....	25
3.8. Instalación de ANT.....	25
3.9. Conclusiones.....	26

CAPITULO IV. CONFIGURACIÓN DE ASTERISK.....	27
4.1. Directorios usados por Asterisk.....	27
4.2. Configuración de Asterisk.....	28
4.2.1. sip.conf.	33
4.2.2. iax.conf.....	35
4.2.3. extensions.conf.....	36
4.2.3.1. Contextos.....	37
4.2.3.2. Extensiones.....	37
4.2.3.3. Prioridades.....	38
4.2.3.4. Aplicaciones.....	38
4.2.4. voicemail.conf.....	39
4.3. Conclusiones.....	40
CAPITULO V. IMPLEMENTACIÓN DE LA HERRAMIENTA DE CONFIGURACIÓN.....	41
5.1. Descripción del funcionamiento.....	41
5.2. Configuración de la herramienta.....	44
5.3. Directorios de la herramienta.....	45
5.4. Base de datos y sus tablas.....	46
5.5. Tipos de menú.....	48
5.5.1. Menú de administrador.....	48
5.5.2. Menú de usuarios.....	48
5.6. Pruebas y resultados.....	50
5.7. Conclusiones.....	52
CONCLUSIONES Y LINEAS FUTURAS	53
6.1. Conclusiones.....	53
6.2. Lineas Futuras.....	54
APÉNDICES.....	55
A. Configuración de tarjeta de sonido.....	56
B. Configuración de softphone.....	58
C. Sentencias SQL.....	62
D. Practica desarrollada con la herramienta de configuración.....	64

LISTA DE FIGURAS

[2.1] <i>Proceso de llamadas entre clientes SIP[1]</i>	8
[2.2] <i>Hibernate un motor de persistencia objeto-relacional</i>	16
[3.1] <i>Pagina de bienvenida de tomcat</i>	23
[4.1] <i>Implementación de una centralita telefónica VoIP con Asterisk, para dos usuarios (softphone)</i>	31
[5.1] <i>Diagrama de funcionamiento de la herramienta configuración</i>	42
[5.2] <i>Esquema de la herramienta de configuración desarrollada con java, implementado el método de n-capas</i>	43
[5.3] <i>Pagina de bienvenida de la herramienta de configuración</i>	44
[5.4] <i>Menú del administrador de la herramienta de configuración</i>	45
[5.5] <i>Tablas que forman parte de la base de datos “central”</i>	46
[5.6] <i>Menú de usuarios de la herramienta de configuración</i>	49
[5.7] <i>Diagrama de la red LAN, en la cual se puso a prueba el funcionamiento de la herramienta de configuración</i>	50
[5.8] <i>Prueba realizada con la herramienta de configuración, haciendo uso del estándar IEEE 802.11 para ampliar la cobertura de la centralita telefónica</i>	51
[A.1] <i>Inicio de configuración de la tarjeta de sonido utilizando alsacnf</i>	56
[A.2] <i>Tarjetas de sonido disponibles en el sistema</i>	57
[A.3] <i>Control de volumen de los dispositivos de sonido</i>	57
[B.1] <i>Parámetros de configuración del softphone Twinkle</i>	58
[B.2] <i>Pantalla principal del softphone Ekiga</i>	59
[B.3] <i>Parámetros de configuración del softphone Ekiga</i>	60
[B.4] <i>Pantalla principal del softphone X-lite</i>	60
[B.5] <i>Parámetros de configuración del softphone X-lite</i>	61
[D.1] <i>Flujo de llamadas disponibles en la PYME</i>	65

LISTA DE TABLAS

[2.1] Comparación entre los diferentes tipos de codec [41,42].....	6
[2.2] Especificaciones de softphone: protocolos que soportan y el sistema operativo en el que pueden ser utilizados [43].....	7
[2.3] Tipos de codec que soporta Asterisk [1].....	12
[2.4] Tipos de protocolos de señalización que puede ser utilizados con Asterisk [1].....	12
[2.5] Requerimientos mínimos de una computadora personal para implementar una centralita telefónica utilizando Asterisk[1].....	13
[2.6] Algunas de las etiquetas HTML utilizadas en el desarrollo de la herramienta.....	14
[4.1] Directorios utilizados por Asterisk en el sistema operativo GNU/Linux[1].....	27
[4.2] Comandos utilizados dentro de la CLI de Asterisk.....	30
[4.3] Parámetros generales de los usuarios sip.conf.....	33
[4.4] Parámetros a configurar en el área de usuarios del archivo sip.conf.....	34
[4.5] Parámetros generales de los usuarios iax.conf.....	35
[4.6] Parámetros a configurar en el área de usuarios del archivo iax.conf.....	36
[4.7] Aplicaciones utilizadas en el dialplan de Asterisk[2].....	38
[5.1] Resultados obtenidos, después de configurado la centralita telefónica con la herramienta.....	50
[D.1] Condiciones a cumplir de una central telefónica VoIP, implementada en una PYME.....	65
[D.2] Contextos, extensiones y protocolos a utilizar en la central telefónica de la PYME.....	66
[D.3] Parámetros de los usuarios de la central telefónica.....	67
[D.4] Checklist: Comprobamos la configuración de la central telefónica.....	67

CAPITULO I

INTRODUCCIÓN

Las telecomunicaciones son una de las industrias que se mantuvieron intactas por las innovaciones de los softwares de código abierto. Esto se mantuvo así hasta que se introdujo un nuevo concepto que permitió combinar las telecomunicaciones (PSTN) con las redes de datos, la VoIP permite transportar la voz por medio del protocolo de Internet. A raíz del nacimiento de la VoIP, surgió una gran cantidad de empresas que se dedican al desarrollo de tecnologías para VoIP.

La VoIP nos permite realizar llamadas entre dos ó mas computadoras que se encuentran en diferentes lugares del planeta, sin ningún tipo de costo, el único requerimiento es tener acceso a Internet de banda ancha. También se pueden realizar llamadas desde una computadora a la PSTN con la ayuda de hardware adicional, el costo por llamada es inferior que realizarlo desde un teléfono tradicional.

Asterisk es un software de código abierto que permite implementar una centralita telefónica VoIP (PBX), fue creado por Marck Spencer y liberada bajo la licencia GPL, con la incorporación de hardware adicional permite conectarse a la PSTN.

En la presente Tesis se implementará una centralita telefónica en una computadora personal que utiliza Asterisk como PBX y hace uso únicamente de software libre. Para facilitar la administración y configuración de la centralita telefónica se construirá una herramienta de configuración que permitirá realizar cualquier modificación a la central telefónica por medio de un navegador web (Epiphany, Iceweasel, Mozilla, etc.).

1.1. Interés de la investigación.

Con los avances de la internet y el nacimiento de la VoIP (Voice over Internet Protocol), en estos días se trata de obtener la mayor cantidad de beneficio de las redes de datos. Para comprender a profundidad el concepto de VoIP, es necesario conocer a fondo el principio de su funcionamiento, que tecnologías soporta y cuales son sus ventajas y desventajas.

La VoIP se ha desarrollado rápidamente en los últimos 5 años, permitiendo realizar más que una simple llamada, actualmente permite realizar videoconferencias. Con el uso de VoIP pueden sustituirse sistemas de telefonía o mejorarlos considerablemente. Por ejemplo, dos oficinas que se encuentran en diferentes países y que se realizan llamadas mutuamente por medio de la telefonía tradicional (PSTN), pueden reducir considerablemente sus costos de llamadas al implementar un sistema de VoIP en su red de datos existente.

El nacimiento de la VoIP permitió interconectar dos mundo que permanecieron separados por mucho tiempo (PSTN - Redes de Datos). Al realizarse esta fusión , nació uno de los softwares libres, que revolucionaría completamente la comunicación de VoIP. Asterisk es la primera aplicación de software libre que permite convertir una simple computadora personal en una central telefónica (PBX).

Para comprender la VoIP es necesario conocer el principio de su funcionamiento: Los protocolos de señalización que soporta (SIP, IAX, h.323, Skinny, etc) y los codec de compresión para la voz (ilbc, g729, g711, g726, etc.). Los protocolos y codec, están divididos en dos grupos, los tipo cerrados -privativos y los abiertos-libres. Skinny es un protocolo privativo al igual que lo es el codec g729. También es necesario conocer algunas de las tecnologías que soporta la VoIP como son: Softphone, ATA, Telefonos IP, y Tarjetas de interfaz.

Los importantes avances en el desarrollo de softwares libres, permiten realizar muchas aplicaciones sin incurrir en ningún tipo de costo, además se puede experimentar libremente, sin ningún tipo de ataduras por falta de licencias. La combinación adecuada de algunos softwares libres con diferentes lenguajes de programación permiten desarrollar herramientas que facilitan las comunicaciones a través de la VoIP.

1.2. Motivación.

Gracias al surgimiento de una aplicación software de código abierto(Asterisk), que permite utilizar una computadora personal como una central telefónica, el cual surgió como resultado del rápido crecimiento de la VoIP. Se pretende obtener el máximo beneficio de la internet, al utilizar la red de datos existente para implementar una central telefónica y reducir los costos por llamada.

Al implementar nuestra propia central telefónica de VoIP se puede gozar de los siguientes beneficios:

- Buzón de voz.
- Musica en espera.
- Grabación de llamadas.
- Transferencia de llamadas.
- Contestador automático.

Para poder gozar de todos estos beneficios y muchos más, no es tan fácil, siempre hay un precio que pagar. El precio que se tiene que pagar al utilizar Asterisk, no es de tipo monetario, hay que invertirle mucho tiempo y dedicación, para lograr configurarlo exitosamente. Asterisk dispone de una gran variedad de versiones entre las cuales están algunas que incluyen un herramienta de configuración gráfica, entre estas se pueden mencionar AsteriskNow, FreePBX y Business Edition.

A pesar de la existencia de estas versiones con las cuales se puede configurar Asterisk con un entorno gráfico. Se optó realizar una herramienta de configuración web por los siguientes motivos: Tener una herramienta personalizada, versátil, que se

acople a nuestras necesidades y de fácil expansión. *AsteriskNow* se apodera por completo del ordenador y no permite que ninguna otra aplicación corra simultáneamente, si se tiene varias particiones con dos sistemas operativos, elimina las particiones y borra toda la información de ambos sistemas. Uno de los principales problemas de *FreePBX* es la incompatibilidad que tiene con algunas versiones de *Asterisk*. *Business Edition* es una versión que esta dirigida a empresas, por lo cual tiene un costo que oscila entre los \$275 y \$3,624.

1.3. Objetivos de la tesis y organización.

Los objetivos planteados en la presente tesis son:

Objetivo general.

- *Implementar una herramienta software para la gestión de una central telefónica .*

Objetivos específicos.

- *Estudiar los conceptos básicos de voz IP, los protocolos de señalización y los codec que utiliza.*
- *Configurar una computadora personal para que funciones como PBX.*
- *Conocer los diferentes dispositivos que se pueden utilizar en una central telefónica de VoIP.*
- *Desarrolla la herramienta de configuración haciendo uso únicamente de software libre.*
- *Ampliar la cobertura de llamadas mediante métodos inalámbricos a través del uso del estándar IEEE 802.11 conocido como Wi-Fi.*

Organización.

El presente trabajo de graduación esta organizado en los siguiente capítulos:

Capitulo I, Introducción.

- *En este capitulo se presentaron los intereses de la investigación, cual fue el motivo de realizar esta trabajo de graduación y los objetivos planteado para el presente trabajo.*

Capitulo II, Conceptos involucrados en la herramienta.

- *El principal aporte de este capitulo sera proporcionar todos los conceptos que están involucrados directamente en el desarrollo de este trabajo.*

Capítulo III, Instalación del software de desarrollo.

- *En esta parte del trabajo se inicia con las preinstalaciones de los softwares, para que funcione correctamente la herramienta de configuración.*

Capítulo IV, Configuración de Asterisk.

- *Este capítulo está dedicado completamente a la configuración de Asterisk. Incluye un ejemplo de configuración para facilitar el aprendizaje.*

Capítulo V. Implementación de la herramienta de configuración.

- *En este capítulo se presenta una herramienta que permite configurar Asterisk a través de un navegador Web.*

Conclusiones y Líneas Futuras:

- *Presenta todas las conclusiones obtenidas en el desarrollo del trabajo de graduación. Además se plantea una serie de propuestas para mejorar la herramienta de configuración en trabajos futuros.*

CAPITULO II

CONCEPTOS INVOLUCRADOS EN LA HERRAMIENTA

En este capítulo se mencionarán todos los conceptos necesarios para poder desarrollar la herramienta de configuración de la centralita telefónica. Se iniciará definiendo VoIP sus ventajas y desventajas, los protocolos de señalización y codec de compresión de audio que se pueden utilizar en VoIP. Se establecerán los requerimientos mínimos de hardware que una computadora personal debe satisfacer para que pueda ser utilizada como central telefónica utilizando Asterisk como pbx, y por último se en listará la base de datos, enlace wi-fi y todos los lenguajes de programación que se utilizarán para poder desarrollar una herramienta de configuración web para la central.

2.1. VoIP.

El acrónimo VoIP por sus siglas en ingles Voice Over Internet Protocol, se puede definir como la transmisión de voz previamente convertida en dato con el uso de codec a través de paquetes IP, esto permite transmitir voz desde una pc a otra por Internet o dentro de una red LAN.

Historia.

En 1995 [40] se anunció el lanzamiento del primer softphone que llamaron “Internet Phone Software”, y fue desarrollado por la compañía Vocaltec, Inc. El software funcionaba comprimiendo la señal de voz, convirtiéndola en paquetes de datos que posteriormente eran enviados por Internet (exactamente igual que hoy).

En 1998 VoIP dió otro gran salto. Un grupo de emprendedores comenzó a fabricar los primeros ATA/gateways para permitir las primeras comunicaciones PC-Teléfono convencional.

En 1999 Cisco vende sus primeras plataformas corporativas para VoIP. Se utilizaba principalmente el protocolo de señalización H.323

En el año 2000 Mark Spencer un estudiante de la Universidad de Auburn crea Asterisk, la primer central telefónica basada en Linux de código fuente abierto.

En el 2002 el protocolo SIP comienza a desplazar el protocolo H.323.

En el 2003 dos jóvenes universitarios - Jan Friis y Niklas Zennstrom - crean un softphone gratuito al que llamaron Skype. Skype tiene un sólo problema: no usa un protocolo standard sino uno propio, que no divulga. Eso no permite que un usuario de Skype pueda comunicarse con otro usuario que no sea de Skype.

En el 2005 En este año Asterisk se afianza como símbolo de VoIP ya que es de código abierto y soporta múltiples protocolos de señalización.

2.1.1. ¿Cómo Funciona? y ¿qué Tecnologías Soporta?

La VoIP al igual que la PSTN necesita protocolos para señalización, la PSTN son las redes de telefonía tradicional (todas las compañías locales). En el caso de PSTN se pueden mencionar el protocolo SS7, quien se encarga de la gestión del establecimiento de las llamadas, fue desarrollado por AT&T y la UTI. Para VoIP están los protocolos de señalización de tipo privativos como son Skype y Skinny, este ultimo propiedad de Cisco System, Inc. También está la gama de protocolos abiertos entre los cuales se pueden mencionar SIP, H.323, IAX, etc. Posteriormente explicaremos los protocolos de sistemas abiertos que son los de interés para el trabajo de graduación.

Además de la señalización para la transmisión de VoIP es necesario hacer uso de Codec (Codificador Decodificador), los cuales se encargan de digitalizar la voz en datos (bits) y los datos en voz. Un codec recibe una señal analógica y la convierte en digital y viceversa. Se considera un codec mejor que otro cuando es capaz de ofrecer mejor calidad de voz usando la misma cantidad de ancho de banda[33].

Entre los diferentes tipos de codec podemos mencionar el G.729, G711u, G711a, iLBC, GSM, etc. de los cuales el único que requiere de licencia para su uso es el G.729, tiene un costo de \$10.0 por canal (computadora) y se puede adquirir en la pagina oficial de Digium[48].

Codec	Bit Rate (Kb/s)	Bandwidth Ethernet (kbps)
G.729	8	31.2
iLBC	13	27.7
GSM	13	29.2
G711	64	84

TABLA 2.1 Comparación entre los diferentes tipos de codec [41,42].

Entre las tecnologías soportadas por VoIP se pueden mencionar.

- Softphone
- ATA
- Telefonos IP
- Tarjetas de Interfaz a la PSTN.

Los softphone son softwares que nos permiten realizar llamadas (simulador de teléfonos) por medio de VoIP. Para su correcto funcionamiento el ordenador debe tener una tarjeta de sonido bien configurada [Apéndice A] y activar codecs de alta compresión. Hay diferentes tipos de softphone, en la TABLA 2.2. se mencionan algunos de ellos.

Softphone	Protocolo	Sistema Operativo
Twinkle	SIP	Linux
X-Lite	SIP	Linux, Windows , MAC
Ekiga	H.323 y SIP	Solari, Linux, Windows
Skype	Skype	Linux, Windows, MAC
iaxcomm	IAX2	Linux, Windows, MAC
KIAX	IAX2	Linux, Windows
cubix	IAX2 y SIP	Windows

TABLA 2.2 *Especificaciones de softphone: protocolos que soportan y el sistema operativo en el que pueden ser utilizados [43].*

Los ATA permiten conectar teléfonos tradicionales analógicos o digitales a una red de VoIP, en un extremo tiene un conector RJ-11 (para el teléfono) y en el otro extremo un RJ-45 (Ethernet punto de dato). Los teléfonos IP se conectan directamente a un red de telefonía IP, pueden accederse desde la WEB, programables, y se les asigna una dirección IP dentro de la red en la que están conectados. Por medio de las tarjetas de Interfaz a la PSTN es posible conectar un sistema de VoIP a la telefonía tradicional (realizar llamadas a teléfonos fijos y móviles), estas tarjetas incluyen puertos de tipo FXS y FXO. La forma correcta de conectarse a la tarjeta es la siguiente: la línea telefónica (PSTN) se conecta al puerto FXO y el aparato telefónico se conecta al FXS, de invertir estas conexión se dañara la tarjeta.

A la hora de adquirir cualquiera de estos dispositivos hay que tomar en cuenta lo siguiente:

- Que protocolo de señalización se utilizará.
- Los dispositivos que se compran deben soportar los codec que se consideran convenientes para su aplicación.

2.1.2. Protocolos de Señalización

La señalización en las comunicaciones sirven para ofrecer información del estado de la llamada al usuario, algunas señales son los tonos de marcado, tonos de ocupado, tono de fuera de uso, etc. Para poder enviar la señalización al realizar una llamada se necesitan protocolos, que se define como un método estándar que permite la comunicación entre procesos, es decir, es un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos. Existe una gran variedad de protocolos de señalización entre los cuales describiremos H.323, SIP e IAX.

2.1.2.1. H.323

El protocolo H.323 [33, 34] es una recomendación de la ITU, se creó originalmente para proveer de un mecanismo para el transporte de aplicaciones multimedia en LANs (Redes de área local) pero ha evolucionado rápidamente para dirigir las crecientes necesidades de las redes de VoIP.

El protocolo H.323 se encuentra implementado por varias aplicaciones de internet que funcionan en tiempo real como Microsoft Netmeeting y Ekiga que anteriormente se llamaba GnomeMeeting.

H.323 fue el primer protocolo de VoIP en adaptar el estándar de IETF (RTP) para transportar audio y vídeo sobre redes IP.

2.1.2.2. SIP

El protocolo SIP [1,2] fue desarrollado por la IETF, y al igual que el protocolo H.323 no transportan audio, únicamente transmiten la señalización para que se pueda realizar una conversación entre dos o más clientes. Es el RTP el que se encarga de transferir el audio y vídeo. La señalización SIP y la conversación viajan por caminos diferentes.

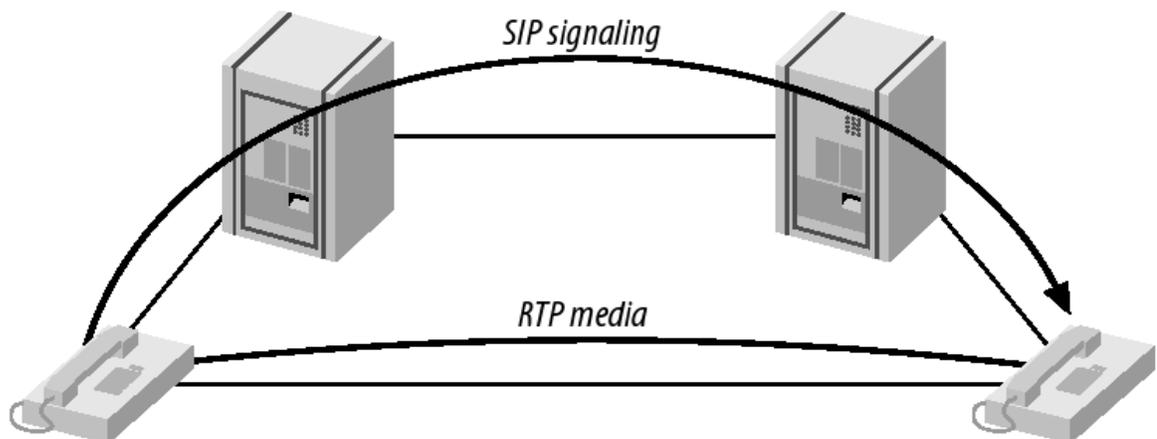


Figura 2.1 Proceso de llamadas entre clientes SIP[1].

Uno de los inconvenientes que se presentan al utilizar el protocolo de señalización SIP, son las NAT, El concepto de NAT, nace como resultado del supuesto agotamiento de direcciones IP, es un mecanismo usado por los router para asignar direcciones IP privadas en aplicaciones domésticas u oficinas y se dispone de una IP publica. Los NAT son los grandes enemigos del RTP [33], el efecto de una NAT en voz sobre IP es la dificultad de recibir conexiones iniciadas desde el exterior de la propia red [33]. Existen varios problemas relacionados con VoIP y NAT, uno de los más comunes es conocido como audio en una sola dirección [33].

2.1.2.3. IAX

La principal diferencia de este protocolo con respecto al SIP y H.323[2,34] es que no presenta ningún inconveniente con las NAT, esto se debe a que no hace uso de RTP para transportar el audio.

IAX[1,2,33,34] fue desarrollado por los creadores de Asterisk, es un protocolo en el cual la señalización y la voz viajan en un mismo canal, IAX es capaz de empaquetar llamadas simultaneas en un sólo flujo de paquetes IP. Este mecanismo es conocido como *trunking* y el resultado es una ahorro en el consumo del ancho de banda.

IAX no es una recomendación de la IETF o UTI. Las especificaciones para IAX están abiertas. Este protocolo es puesto en práctica en los productos de Digium. Una de las desventajas es que al ser un protocolo reciente la disponibilidad en los dispositivos de VoIP es menor a SIP y H.323.

2.1.3. Ventajas y Desventajas de VoIP

Ventajas:

- Una de las primeras ventajas es su bajo costo por llamada entre un sistema VoIP y la PSTN.
- Las llamadas entre sistemas de VoIP son gratuitas (PC-PC, PC-Telefono IP).
- Se utiliza la red de datos existente para implementar un sistema de VoIP.
- Se puede realizar una llamada desde cualquier lugar donde haya conectividad de Internet.
- Además se goza de las siguientes características: Llamada en espera, transferencia de llamadas, música de espera, mensajería de voz etc.
- Se puede ampliar la cobertura por medio de tecnologías como WI-FI.

Desventajas:

- Para mantener una buena transmisión es necesario disponer de una conexión de banda ancha.
- VoIP requiere de una conexión eléctrica para los dispositivos que utiliza.
- No están disponibles Llamadas de emergencia 911.
- Al utilizar la red de datos es susceptible a todo tipo de virus.

2.2. PBX.

El acrónimo PBX de las siglas en inglés *Private Branch Exchange*, para el cual su traducción al español sería *Línea Privada de Intercambio* o *Central Telefónica*.

Una PBX Es un dispositivo que se encarga de direccionar las llamadas internas (locales) y externas (Red Telefónica Local) hacia las diferentes extensiones. El uso de un PBX evita conectar todos los teléfonos de una oficina de manera separada a la red de telefonía local (PSTN), evitando de esta forma los cargos mensuales por cada una de las llamadas que se realizan dentro de la oficina, pagando únicamente las llamadas salientes.

Características de una PBX:

- Contestador automático
- Distribuidor automático de tráfico de llamadas
- Desvío de llamadas
- Transferencia de llamadas
- Mensaje de Bienvenida
- Música en espera

2.3. Debian, El Sistema Operativo.

Hablar de un sistema operativo tan completo como Debian [37,44] se llevaría la mayor parte del trabajo de graduación. Hay una gran cantidad de libros, tutoriales y foros por medio de los cuales se pueden aprender a utilizar este sistema operativo, a continuación se tratara de introducir brevemente algunas generalidades de este potente sistema.

Debian fue desarrollado por Ian Murdok, es un sistema operativo libre que utiliza Linux como núcleo, pero la mayor cantidad de herramientas viene del proyecto GNU.

GNU (GNU No es Unix), es un proyecto que nace con el objetivo de crear un sistema operativo libres y esto se cumple cuando se une con el kernel Linux en 1992.

Debian GNU/Linux ofrece más que un S.O. puro; viene con 18733 paquetes, precompilados distribuidos en un formato que hace más fácil la instalación en su computadora.

Debian maneja simultáneamente tres distribuciones. Estas son:

- Estable (Etch)
- De Prueba (Lenny)
- Inestable (Sid)

Si eres nuevo en el uso de este sistema operativo se recomienda instalar siempre la distribución estable. Como un dato curioso, el origen de los nombres de las distribuciones (Etch, Lenny, Sid) provienen de la película "Toy Story".

Se eligió este sistema operativo como la plataforma para implementar la herramienta de configuración de una central telefónica por ser libre, muy potente y además es el que tiene mayor soporte, aparte de eso posee una gran cantidad de paquetes que serán de utilidad para el desarrollo de esta herramienta, entre los cuales se pueden mencionar mysql, ant, tomcat, asterisk, etc...

2.4. Asterisk.

Asterisk es una aplicación software de una central telefónica VoIP[1,2,3] que soporta múltiples protocolos de señalización, fue liberada bajo la licencia GPL (General Public License) y creada originalmente para funcionar en el sistema operativo GNU/Linux, actualmente corre también en sistemas como OpenBSD, FreeBSD, Mac OS X, Solaris Sun y Windows, pero GNU/Linux sigue siendo la que mayor soporte ofrece a esta aplicación. Fue desarrollada por Mark Spencer, propietario de la empresa Linux Support Service, Inc que posteriormente se convirtió en Digium, Inc.[48].

Mark Spencer se basó en la idea de Jim Dixon[1], quien creía que serían mucho más económicos los sistemas de telefonía si existiera una tarjeta creada únicamente de componentes electrónicos básicos y controlada por software desde una CPU. Dixon fue el creador de "Zapata Telephony Project"[45].

Para poder implementar un aplicación de una central telefónica con Asterisk es necesario realizar después de su instalación, la configuración de los archivos que se encuentran dentro del directorio `/etc/asterisk/`, algunos de ellos son `sip.conf`, `iax.conf`, `extension.conf`, `voicemail.conf`, etc., los archivos a configurar dependerán de la forma que se quieran tratar cada una de las llamadas y el protocolo de señalización que se desee utilizar.

Si desea implementar una aplicación sencilla sin la necesidad de conectarnos a la PSTN simplemente bastará instalar el paquete de Asterisk, pero si la aplicación a desarrollar incluye conexiones a la PSTN se deberá instalar algunos paquetes extras, los cuales son:

- zaptel
- libpri

Zaptel sirve para configurar los driver de la tarjeta PCI que se utiliza para conectarse a la PSTN. La herramienta que se desarrollará no incluye conexiones a la PSTN, por lo tanto no se explicará el proceso de instalación de estos paquetes, pero se recomienda leer la documentación en la cual se encuentra toda los pasos de su instalación[1].

Si desea instalar zaptel debe estar seguro que las cabeceras del núcleo de Linux (linux-headers) estén instaladas en el sistema, la secuencia de instalación es la siguiente: zaptel, libpri y por ultimo asterisk.

Asterisk dispone de una gran variedad de versiones tales como:

- Asterisk 1.2
- Asterisk 1.4
- Asterisk 1.6

- AsteriskNow
- AsteriskFreePBX
- Asterisk_TriBox
- Asterisk Business Edition

Algunas de las versiones antes mencionadas como AsteriskNOW, AsterisFreePBX , Asterisk TrixBox y Asterisk Business Edition, disponen de una herramienta de configuración, no así el resto de versiones las cuales se tiene que configurar manualmente. Asterisk Business Edition[48] es la única versión, que se tiene que pagar para adquirir, los precios oscilan entre \$275 y \$3624 e incluyen soporte técnico.

Como ya se vio anteriormente para poder transmitir VoIP es necesario un protocolo de señalización y codec para comprimir el audio, asterisk soporta los siguientes protocolos[1] y codec[1,46].

Codec	Bit Rate (Kbps)	Requiere de Licencia
G.711	64	No
G.726	32	No
G.723.1	5.3,6.3	Si
G.729a	8	Si
GSM	13	No
iLBC	13.3,15.2	No
Speex	entre 2.5 y 22.4	No

TABLA 2.3 *Tipos de codec que soporta Asterisk [1].*

Protocolo	De Uso
IAX	Abierto-Libre
SIP	Abierto-Libre
H.323	Abierto-Libre
MGCP	Abierto-Libre
Skinny/SCCP	Cerrado-Propietario
UNISTIM	Cerrado-Propietario

TABLA 2.4 *Tipos de protocolos de señalización que pueden ser utilizados en Asterisk [1].*

*MGCP: Media Gateway Control Protocol, es una recomendación de la IETF.
UNISTIM: Propiedad de Nortel Networks Corporation.*

Para poder utilizar Asterisk como central telefónica en una computadora personal, es necesario que pueda satisfacer algunos requerimientos mínimos de hardware[1]. Los requerimientos dependerán de la aplicación.

Propósito	Cantidad de Canales	Requerimiento mínimo
Uso Domestico	No mas de 5	400-MHz x86, 256 MB RAM
Pequeña Oficina	de 5 a 10	1-GHz x86, 512 MB RAM
Pequeño Sistema de Negocios	Hasta 15	3-GHz x86, 1 GB RAM
De Mediano a Grandes Sistemas	Mucho mas de 15	Dual CPUs, posiblemente también múltiples servidores

TABLA 2.5 *Requerimientos mínimos de una computadora personal para implementar una centralita telefónica utilizando Asterisk[1].*

2.5. Base de Datos Mysql.

Mysql es un sistema de base de datos relacional [26,27,28]. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad (Int, String, Date, Long, etc.). Utiliza el lenguaje SQL para realizar consultas.

Una base de datos en su definición más sencilla es un conjunto de tablas las cuales incluyen una colección de datos (registros) relacionados y organizados en campos (columnas) y filas.

Mysql fue creado por Mysql AB y es una base de datos de carácter libre (liberada bajo la licencia GPL) y multiplataforma (GNU/Linux, FreeBSD, Mac OS X, Solaris, Windwos, etc.).

Existe una gran variedad de API que permiten a aplicaciones escritas en diversos lenguajes de programación acceder a la base de datos, algunos de los lenguajes de programación que se pueden utilizar para gestionar la base de datos son: Java, PHP, C, C++, Perl, Python, etc. la herramienta a desarrollar en este trabajo utilizará el lenguaje Java y el API que permitirá acceder a Mysql es JDBC.

Para implementar una herramienta en la cual se pueda consultar la base de datos vía web es necesario instalar Mysql-Common, Mysql-Server y Mysql-Client. Existen otras herramientas como Mysql-Admin la cual nos permite administrar y crear bases de datos por medio de un entorno gráfico.

2.6. Servidor Web Tomcat.

Tomcat es un contenedor servlet de java [22,23,32] y un servidor web creado por Apache Software Foundation, fue escrito bajo el lenguaje de programación java, es de código abierto, multiplataforma y por defecto utiliza el puerto TCP 8080.

La mayor parte de la herramienta a desarrollar está basada en lenguaje Java y JSP, es por ese motivo que se utilizará este servidor. Tomcat es el encargado de compilar las páginas JSP y convertirlas en servlet, las últimas versiones disponibles de tomcat son tomcat-4.0, tomcat-5.0, tomcat-5.5, tomcat-6.0, etc. la versión que se utilizará para este trabajo de graduación es la 5.5 y se puede obtener de la página oficial de tomcat [47] y también viene incluido en los paquetes de Debian.

2.7. Lenguajes de Programación

Para desarrollar la herramienta de configuración web de la central telefónica será necesario usar los siguientes lenguajes de programación: HTML, XML, Java, JavaServer Pages (JSP) y Script.

A continuación se realizará una breve descripción de los lenguajes de programación antes mencionados y su función dentro de la herramienta.

2.7.1. HTML.

Es un conjunto o serie de etiquetas incluidas en archivos de texto que definen la estructura de una página web y sus vínculos con otros documentos[30,31].

Los navegadores leen estos archivos de texto e interpretan esas etiquetas para determinar cómo desplegar la página Web.

Serán estas páginas junto a las JSP que permitirán interactuar con el usuario por medio de un navegador.

En la herramienta a desarrollar se utilizan las siguientes etiquetas:

Etiqueta	Descripción
<Form>	Son formularios y permiten solicitar información al visitante.
<Table>	Genera tablas en la cual se puedan presentar los datos.
	Da formato al texto, colores, tamaño, etc.
	Permite insertar imágenes en la página
	Crea listas con viñetas ordenadas.
<A>	Hipervínculo, permite hacer saltos de pag.
<Input>	Permite ingresar datos en un formulario.

TABLA 2.6 Algunas de las etiquetas HTML utilizadas en el desarrollo de la herramienta de configuración.

2.7.2. XML.

XML es un lenguaje utilizado para hacer referencia a otros lenguajes (metalenguaje) extensible de etiquetas [31], desarrollado por el World Wide Web Consortium (W3C), permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

La aplicación de XML en la herramienta de configuración sera para definir archivos en los cuales se especifica al servidor tomcat (*context.xml* y *web.xml*) el nombre de la base de datos que se utilizará y otros parámetros adicionales. También se utiliza para configurar los archivos de mapeo de hibernate(*hibernate.cfg.xml*, *Sip.hbm.xml*, etc.) y para crear el *build.xml* el cual permitirá compilaran todas las classes de java.

2.7.3. Java.

Java es un lenguaje de programación orientada a objetos [12,13,15,19,20], de código libre y multiplataforma. sus creadores(*Sun Microsystems*) defines a java como “*un lenguaje simple , orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra , de altas prestaciones, multitarea y dinámico*”.

Cuando se programa en java no se empieza de cero, cualquier aplicación que se desarrolle depende de una gran cantidad de classes pre-existentes que forman parte de la API de java.

Para realizar una aplicación con java se requiere de un programa que nos permita compilar y ejecutar dichas aplicaciones, Sun Microsystems distribuye gratuitamente el JDK (java Development Kit), el cual consiste en un conjunto de programas y librerías que permiten compilar y ejecutar programas en java.

Toda la lógica de la herramienta de configuración de la central telefónica se basara en este potente lenguaje de programación.

2.7.4. JavaServerPages (JSP).

JavaServer Pages es una tecnología basada en Java que simplifica el proceso de desarrollo de sitios Web dinámicos [18,24,25]. Con JSP, tanto los desarrolladores como los diseñadores Web pueden incorporar de forma rápida elementos dinámicos en páginas Web, utilizando código Java y una serie de etiquetas especiales determinadas.

Como una primera aproximación se puede decir que JSP es un tipo de lenguaje de script de servidor, pero internamente funciona de manera completamente distinta.

Las páginas JSP se encuentran como ficheros de texto que poseen las extensión .JSP. Los ficheros JSP contienen código HTML tradicional combinado con código en Java (llamado scriptlet) y una serie de etiquetas especiales (<%, <%=, <%@, etc..), que se ejecutarán en el servidor y darán lugar al resultado de la ejecución de la página en forma de código HTML.

Por medio de las paginas JSP se verificara que todos los datos de entrada en la herramienta de configuración web sean correctos, y no permite que se envíe ningún campos vacío de los fomularios de las paginas HTML.

2.7.5. Script.

Un script se puede define como un conjunto de instrucciones que facilitan la ejecución de una tarea, haciéndolo automático. los scrip se ejecutan desde la consola del sistema operativo GNU/Linux.

La utilización de script en esta herramienta es únicamente para iniciar y detener los tres servidores (Asterisk, Tomcat y Mysql) simultáneamente (*startpbx.sh* y *stoppbx.sh*). De no haber estos script se deben iniciar por separado cada uno de los servidores.

2.8. Hibernate.

Es un motor de persistencia objeto-relacional [6,7,8,10] para el lenguaje Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Cabe destacar que es multiplataforma y de código abierto.

Se entenderá por objetos persistentes: a la capacidad de un lenguaje de programación que nos permita hacer que nuestros datos (objetos) puedan ser almacenados y recuperados para poder reutilizarlos en otros procesos.



Figura 2.2 *Hibernate un motor de persistencia objeto - relacional.*

Hibernate nos facilita hacer persistente los objetos de nuestro programa en una base de datos relacional. Funciona para las Bases de Datos: Oracle, DB2, Mysql etc...

La manera tradicional para conectarse a la base de datos era utilizar JDBC a través de sentencias SQL dispersas en muchas clases de la aplicación, este tipo de conexión es adecuada para proyectos pequeños, ya que un cambio mínimo requerirá la revisión de casi todo el código del programa.

Hibernate ayuda a separar las sentencias SQL del código de las clase de java. Por lo tanto hibernate es el puente entre la aplicación desarrollada en Java y la base de datos, sus funciones van desde la ejecución de sentencias SQL (utiliza JDBC) y HQL, hasta la creación, modificación y eliminación de objetos persistentes.

Hibernate funciona asociando a cada tabla de la base de datos una clase de java en la cual se define métodos set y get.

La asociación entre las tablas de la base de datos y las clases se definen en los archivos de mapeo de Hibernate (*.cfg.xml* y *.hbm.xml*).

2.9. ANT.

Ant es la primera herramienta de gestión para entornos Java que facilita la compilación de las clases de java [16,17], permite la documentación de nuestra aplicación. Ant fue desarrollada por Apache Software Foundation y es de código abierto. Está escrito enteramente en Java, y es independiente de la plataforma. Usa etiquetas XML para construir el archivo *build.xml* en el cual se define lo que desea realizar con ANT (compilar, agregar classpath, documentar, empaquetar, etc.).

Una vez definido el archivo *build.xml* simplemente falta ejecutar el comando *ant* desde la consola y todo lo que se especificó en el archivo *build.xml* se realiza. Cuando ejecute el comando *ant* debe encontrarse en el directorio donde se encuentra el archivo *build.xml*.

2.10. Enlace Wi-Fi.

El acrónimo WIFI de las siglas en inglés Wireless Fidelity, es un conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE-802.11. Creado para ser utilizado en redes locales inalámbricas, es frecuente que en la actualidad también se utilice para acceder a internet, por ello facilita las comunicaciones en una sociedad dinámica como la actual.

Una vez concluida la herramienta de configuración de la central telefónica, se hará uso de este estándar para poder ampliar la cobertura de nuestra red.

2.11. Conclusiones.

En este capítulo se determinaron y explicaron todos los conceptos que están involucrados directamente en el desarrollo del presente trabajo de graduación. También se explicó la función que tendrá cada uno de los programas o softwares en la implementación de la herramienta de configuración de la central telefónica. Cuando se desee implementar la herramienta de configuración, será mucho más fácil elegir el ordenador que se utilizará como central telefónica en base a la tabla 2.5.

CAPITULO III

INSTALACIÓN DEL SOFTWARE DE DESARROLLO

Este capítulo tratará exclusivamente de la obtención e instalación de todos los paquetes y/o software de los cuales se harán uso para poder desarrollar la herramienta de configuración. Entre los software que se utilizarán podemos mencionar el sistema operativo (Debian Etch) , el servidor de VoIP (Asterisk), el servidor web (Tomcat) , nuestra base de datos (Mysql) y otros.

Como puede verse este capítulo proporciona la base para poder desarrollar la herramienta. Un factor común que presentan todos los software que se utilizarán es que son de **código abierto**, esto hace más atractiva la herramienta ya que todo su desarrollo estaría bajo software libre.

A continuación se mencionarán en el orden correspondiente de su instalación todos los software. Los pasos de instalación y/o configuración se deberán realizar como root, ya que se copiarán archivos en directorios en los cuales solo tiene permiso de escritura el superusuario. Todas las utilidades deben instalarse en la computadora que será nuestra centralita telefónica.

3.1. Obteniendo Debian.

Para obtener Debian GNU/Linux, se ingresa a la página web oficial de Debian [37], se navega a la opción *Getting Debian* y se descargan los torrent para CD o DVD "stable" [36]. para este proyecto en particular se descargaron los torrent i386 ya que se utiliza un microcontrolador Intel Pentium 4.

El proceso de instalación del sistema operativo no se explicará en este trabajo de graduación ya que no es el objetivo de este, por lo que se recomienda leer la documentación [35,44], creada por los desarrolladores de Debian GNU/Linux.

3.2. Instalación de Asterisk.

En el capítulo anterior se dio una explicación de Asterisk, a continuación se explicarán las formas de obtener e instalar Asterisk y todo el capítulo siguientes se le dedicará a su configuración.

Antes de poder instalar Asterisk es necesario tener instalado ciertos paquetes en el sistema operativo para poder compilar y ejecutar Asterisk correctamente, los paquetes necesarios son los siguientes:

Paquete	Dependencia
1) gcc-4.1	libssp0
2) bison	
3) openssl	
5) libssl-dev	lib6c libc6-dev libc6-i686 linux-kernel-headers zlibg-dev
6) g++	g++-4.1 libstdc++6-4.1-dev
7) libncurses5-dev	
8) mpg123-0.61	

Todos los paquetes se pueden instalar desde la consola (#apt-get install bison g++ etc.) [39] o con la herramienta Gestora de Paquetes Synaptic. Debes tener configurado tu source.list [ver pag. 20]

Una vez realizadas las preinstalaciones tenemos que seleccionar una versión de Asterisk. Actualmente la última versión estable de Asterisk es la 1.6, pero se trabajará con la versión 1.2.13 ya que es la más soportada y está incluida en los paquetes del sistema operativo.

Formas de Instalar Asterisk

Asterisk se puede instalar de dos formas [1, 2 y 4], una de ellas es compilando el código fuente y la otra es instalándolo con el gestor de paquetes Synaptic.

a) Primera forma: **Obtención y compilación del código fuente**

El código fuente se puede obtener directamente de la pagina de Asterisk:

www.asterisk.org

Uno de los inconvenientes de obtenerlo de esta forma es que solo aparecen las ultimas dos versiones estables liberadas recientemente, por lo que se recomienda bajarlo de la pagina siguiente.

<http://downloads.digium.com/pub/asterisk/releases/>

Ahí podremos encontrar todas las versiones estables disponibles de Asterisk, navegar y descargamos el paquete [asterisk-1.2.13.tar.gz](http://downloads.digium.com/pub/asterisk/releases/asterisk-1.2.13.tar.gz), copiarlo en el directorio /usr/src y posteriormente se descomprime.

```
CORINTO:~# cp -p asterisk-1.2.13.tar.gz /usr/local/  
CORINTO:/usr/local# tar -xzvf asterisk-1.2.13.tar.gz /usr/local/
```

Se entra a la carpeta que se crea al descomprimir el archivo .tar.gz y se procede a su instalación.

```
CORINTO:/usr/local# cd asterisk-1.2.13/
```

Asterisk se instala por medio del compilador make.

```
CORINTO:/usr/local/asterisk-1.2.13#make clean
CORINTO:/usr/local/asterisk-1.2.13#./configure
CORINTO:/usr/local/asterisk-1.2.13#make
CORINTO:/usr/local/asterisk-1.2.13#make install
CORINTO:/usr/local/asterisk-1.2.13#make samples
```

Verificar que la instalación se haya realizado correctamente iniciando Asterisk.

```
CORINTO:#asterisk          // inicia el servidor
CORINTO:#asterisk -vvvr    // entrar a la CLI de asterisk
```

```
Connected to Asterisk 1.2.13 currently running on CORINTO (pid = 22931)
Verbosity was 0 and is now 3
CORINTO*CLI>
```

Detener el servidor y salir del CLI de Asterisk con el comando *stop now*

```
CORINTO*CLI> stop now
CORINTO*CLI>
Disconnected from Asterisk server
Executing last minute cleanups
CORINTO:/usr/local#
```

b) Segunda forma: Instalación con Synaptic

Para poder instalar Asterisk con Synaptic es necesario configurar previamente el archivo `source.list` [39] en el cual se especifica si se instalará de un CD, DVD ó algún repositorio(internet).

La instalación para este trabajo se realizara desde un repositorio, para esto es necesario editar el archivo `source.list` de la siguiente forma.

```
CORINTO:#gedit /etc/apt/source.list

deb http://ftp.us.debian.org/debian/ etch main
deb-src http://ftp.us.debian.org/debian/ etch main

deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib
```

Una vez realizados los cambios en el archivo, se guarda y se actualizan los paquetes del sistema operativo con el siguiente comando.

```
CORINTO:#apt-get update
```

En el gestor de paquetes synaptic se seleccionan los paquetes siguientes:

- Asterisk
- Asterisk-classic
- Asterisk-config
- Asterisk-sounds-main
- mpg123-0.61

Se Aplican los cambios y se instalan todos los paquetes antes mencionados. Una vez instalado se realiza la misma prueba que se explicó anteriormente. El paquete mpg123-0.61 permite reproducir archivos con formato mp3 en música en espera.

3.3. Instalación de JDK.

La página principal para poder descargar JDK es la siguiente

<http://java.sun.com/javase/downloads/index.jsp>

Descargue JDK 6 Update 5 ("Self-Extracting-Binary") jdk-6u5-linux-i586.bin, copiar el archivo "jdk-6u5-linux-i586.bin" en el directorio /usr/local

```
CORINTO:home/ovidio# cp -p jdk-6u5-linux-i586.bin /usr/local
```

Se entra a la carpeta /usr/local para cambiar los permisos y hacer ejecutable el archivo.

```
CORINTO:/usr/local# chmod +x jdk-6u5-linux-i586.bin
```

Una vez hecho el cambio se instala.

```
CORINTO:/usr/local# ./jdk-6u5-linux-i586.bin
```

Cambiamos el nombre de la carpeta que se creo a *jdk*

```
CORINTO:/usr/local# mv jdk-6u5-linux-i586 jdk
```

Es necesario configurar las variables de entorno editando en el archivo /home/ovidio/.bashrc, donde "ovidio" es el nombre del usuario actual(el punto del archivo .bashrc indica que es de tipo oculto).

```
CORINTO:/usr/local# gedit /home/ovidio/.bashrc
```

Hay que añadir las siguientes líneas después de los primeros comentarios

```
JAVA_HOME="/usr/local/jdk"  
PATH="$PATH:/usr/local/jdk/bin"  
export JAVA_HOME PATH
```

Se Actualizan las variables de entorno:

```
CORINTO:/usr/local# source /home/ovidio/.bashrc
```

Para verificar que fue instalado correctamente realizar la siguiente prueba.

```
ovidio@CORINTO:~$ java -version
```

```
java version "1.4.2"
```

```
gij (GNU libgcj) version 4.1.2 20061115 (prerelease) (Debian 4.1.1-20)
```

```
Copyright (C) 2006 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

ii Si no despliega esta información lo mas probable es que cometió algún error !!

3.4. Instalación de Tomcat

Descargar [32] la distribución binaria 5.5 de <http://tomcat.apache.org/download-55.cgi>, navegar y descargar el archivo CORE .tar.gz, se copiamos en /usr/local

```
CORINTO:/home# cp -p /apache-tomcat-5.5.26.tar.gz /usr/local/
```

Descomprimir y cambiarle el nombre de la carpeta a tomcat

```
CORINTO:/usr/local# tar -zxvf apache-tomcat-5.5.26.tar.gz
```

```
CORINTO:/usr/local# mv apache-tomcat-5.5.26 tomcat
```

Para configuración tomcat también es necesario definir sus variables de entorno en el archivo /home/ovidio/.bashrc.

```
CATALINA_HOME=/usr/local/tomcat
```

```
export CATALINA_HOME
```

El archivo .bashrc quedara de la siguiente forma con los PATH de JDK y Tomcat.

```
# ~/.bashrc: executed by bash(1) for non-login shells.
```

```
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
```

```
# for examples
```

```
# //////////////////////////////////////////////////////////////////// JDK y Tomcat
```

```
JAVA_HOME="/usr/local/jdk"
```

```
CATALINA_HOME="/usr/local/tomcat"
```

```
PATH="$PATH:/usr/local/jdk/bin"
```

```
export JAVA_HOME CATALINA_HOME PATH
```

Actualizar las variables de entorno:

```
CORINTO:/usr/local# source /home/ovidio/.bashrc
```

Por último se ejecuta el script para verificar que tomcat se configuro correctamente.

```
CORINTO:/home# /usr/local/tomcat/bin/startup.sh
Using CATALINA_BASE: /usr/local/tomcat
Using CATALINA_HOME: /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME: /usr/local/jdk
```

Si todo a ido bien debería ver la página de bienvenida de Tomcat, introduciendo la URL <http://localhost:8080> en un navegador (Mozilla, Epiphany, Iceweasel, etc.)



Figura 3.1 Pagina de bienvenida de tomcat.

De no ver la Figura 3.1, posiblemente se a cometido algún error en la configuración y se recomienda revisar nuevamente todos los pasos anteriores. Para detener el servidor tomcat se utiliza el script shutdown.sh

```
CORINTO:/home# /usr/local/tomcat/bin/shutdown.sh
```

3.5. Obteniendo JDBC.

Para obtener el API (JDBC) que permite la ejecución de operaciones sobre bases de datos desde una aplicación realizada con lenguaje Java, se hace desde la página

<http://dev.mysql.com/downloads/connector/jj/5.0.html>

Seleccionar el Source and Binaries (.tar.gz) 5.0.8 8.3M, una vez descargado se descomprime en cualquier lugar del sistema operativo y se copia el paquete : mysql-connector-java-5.0.8-bin.jar en la carpeta lib de tomcat, como se muestra a

continuación.

```
CORINTO:/home/ovidio# tar -xvzf mysql-connector-java-5.0.8.tar.gz
CORINTO:/home/ovidio# cp mysql-connector-java-5.0.8bin.jar /usr/local/tomcat/..
common/lib
```

3.6. Instalación de Mysql.

La base de datos Mysql se puede instalar de varias formas [26, 28 y 29], una de ellas es extrayendo el código binario desde la página oficial de mysql e instalarlo manualmente, otra forma y sera la que se utilizará, es instalarlo con el gestor de paquetes synaptic.

Paquetes a instalar:

- mysql-server-5.0
- mysql-cliente-5.0
- mysql-common

Una vez instalados iniciamos el servidor mysql, recordar que para iniciar los servidores debe tener permisos de superusuario.

```
Corinto:~# /etc/init.d/mysql start
Starting MySQL database server: mysqld already running.
CORINTO:~#
```

Entrar a la base de datos mysql en primer plano

```
Corinto:~# mysql
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
Mysql>_
```

Se le asigna el password al usuario root (!!!no modificar el password=root!!!). Si desea modificar el nombre de usuario o password se recomienda leer la nota del archivo PRIMERO_LEEME que se encuentra en centralita.tar.gz. dentro del CD.

```
mysql>set password=password('root');
mysql>quit
Bye
```

Y por último finalizar el servidor mysql

```
Corinto:~# /etc/init.d/mysql stop
Stopping MySQL database server: mysqld.
```

3.7. Obteniendo Hibernate.

El motor de persistencia objeto-relacional [5, 6 y 7] se obtiene directamente de la pagina oficial de hibernate.

<http://www.hibernate.org/30.html>

Descargar *Hibernate Core 3.3* y descomprimirlo

Los archivos .jar que necesitaremos son los siguientes y se encuentran en la carpeta lib de hiberntate3.3.

- antlr-2.7.5H3.jar
- cglib-2.0-rc2.jar
- commons-collections-2.1.jar
- commons-logging-1.0.3.jar
- dom4j-1.4.jar
- ehcache-0.6.jar
- hibernate3.jar
- jdbc2_0-stdext.jar
- jta.jar

En el capítulo V se especifica en que carpeta de la herramienta están incluidos todos los archivos .jar de hibernate.

3.8. Instalación de ANT.

Esta herramienta de compilación [16, 17] es necesaria instalarla solo si se tienen intenciones de modificar el código fuente de las classes de java, ya que facilita la compilación de todas las classes.

La herramienta ANT se instala desde la consola con *apt-get*, también se puede hacer desde el gestor de paquetes synaptic.

```
CORINTO:~#apt-get install ant
CORINTO:~# ant
Buildfile: build.xml does not exist!
Build failed
CORINTO:~#
```

Con todos los paquetes utilitarios instalados ya se esta listo para poner en marcha la herramienta de configuración.

3.9. Conclusiones.

En este capítulo se explicó todo el proceso de obtención e instalación de los paquetes utilitarios para poder desarrollar la herramienta de configuración. Se realizaron pruebas con cada uno de ellos para verificar que su instalación y configuración fuera correcta. Las pruebas de cada una de las utilidades son de mucha importancia, si una utilidad no está configurada correctamente, cuando se realicen las pruebas con la herramienta, de igual forma no funcionará ya que hace uso de todas las utilidades antes mencionadas.

La utilidad *ant* sirve para compilar todas las clases de Java, si no se pretende modificar las clases de la herramienta, se puede obviar su instalación. Se recomienda que se instale para usos futuros.

CAPITULO IV

CONFIGURACIÓN DE ASTERISK

En los capítulos anteriores se determinaron las formas de instalar Asterisk, los protocolos de señalización y los codec de compresión que pueden ser utilizados. Además se determinaron los requerimientos mínimos de hardware que una PC debe cumplir para que pueda ser utilizada como central telefónica de VoIP.

En el presente capítulo se explicará la forma de iniciar y detener Asterisk, se mencionaran algunos comandos que serán de mucha utilidad en el CLI (Command Line Interface) para verificar el estado de las llamadas entre usuarios. Además se explicará como están distribuidos los directorios de asterisk en el sistema operativo.

Asterisk contiene un aproximado de 50 archivos de configuración y la cantidad de archivos a configurar dependerá de la aplicación que se implemente. Se explicaran los parámetros necesarios para definir usuarios con protocolo IAX y SIP. Se editará un dialplan básico, ya que este define la forma en la que se tratarán las llamadas entrantes y salientes.

Los archivos de Asterisk que se configuraran en el presente capítulo son *sip.conf*, *extension.conf* y *voicemail.conf*.

4.1. Directorios usados por Asterisk.

Asterisk utiliza varios directorios[1] en el sistema GNU/Linux para gestionar los diferentes aspectos y funciones, tales como grabaciones de voz, mensajes de voz y archivos de configuración. En esta sección se examinan los directorios principales que se crean en la instalación y quedan configurados en el archivo *asterisk.conf*.

A continuación se detallará cada uno de los directorios utilizados por Asterisk, la distribución de los principales directorios de Asterisk se pueden ver en el archivo *asterisk.conf*.

DIRECTORIO	DESCRIPCIÓN
/etc/asterisk	Contiene todos los archivos de configuración de Asterisk (<i>sip.conf</i> , <i>iax.conf</i> , <i>manager.conf</i> , etc.) solo el <i>zaptel.conf</i> se encuentra en <i>/etc/</i> .

TABLA 4.1 Directorios utilizados por Asterisk en el sistema operativo GNU/Linux [1].

DIRECTORIO	DESCRIPCIÓN
/var/lib/asterisk	En este directorios se encuentra el archivo <i>astdb</i> el cual contiene la base de datos de la información de Asterisk, son como registros.
/usr/lib/asterisk/modules	Contiene todos los módulos (codec, formatos y canales utilizados) que se cargan al iniciar Asterisk.
/usr/share/asterisk	Contiene los directorios: -firmware -mohmp3 - sound Cuando se configura asterisk con musica en espera es en el directorio <i>mohmp3</i> que se copian los archivos .mp3 a reproducir. En el directorio <i>sound</i> se encuentran todos los sonidos disponibles a utilizar.
/var/spool/asterisk	Contiene los directorios: - outgoing - tmp - voicemail En el directorio <i>tmp</i> simplemente se guarda información temporal. Todos los correos de voz que se envian a los usuarios que tiene un buzón de voz se guardan en <i>voicemail</i> .
/var/run/asterisk	Contiene la información del identificador de procesos(PID), de todos los que se encuentran activos en el sistema.
/var/log/asterisk	En este directorio es donde Asterisk registra información mediante los archivos de registro(.log), Puede controlarse el tipo de información que se desea registrar editando el fichero de configuración <i>logger.conf</i> .

TABLA 4.1 (continuación).

4.2. Configuración de Asterisk.

Antes de iniciar con la configuración de Asterisk es necesario conocer algunos comandos[1,2,3,4] que serán de mucha ayuda.

Iniciando Asterisk

Asterisk puede ser iniciado de varias formas, la más fácil es iniciar Asterisk ejecutando el comando *#asterisk* directamente desde la consola de linux. Para iniciar Asterisk debe tener los permisos de superusuario, de lo contrario dará el siguiente mensaje: *bash: asterisk: command not found*.

Si se edita desde la consola el comando `#asterisk` (`CORINTO:/home/ovidio# asterisk`) este será cargado como un demonio. También hay algunas banderas que permiten conectarse a la CLI de Asterisk. Para consultar toda la variedad de banderas disponibles ejecute: `CORINTO:/home/ovidio# asterisk -h`

A continuación se explicaran algunas de las bandera más utilizadas.

- C
Consola. Esto permite conectarse a la CLI de Asterisk.
- v
Verbosidad. Envía las salidas del depurador del CLI.
- r
Remoto. Reconectar remotamente Asterisk si fue iniciado como demonio.
- rx "stop now"
Ejecución. Se utiliza para ejecutar comandos de la CLI sin estar conectados a la CLI de Asterisk. "stop now" es el comando que se ejecutará, puede ser cualquier comando valido de Asterisk.

Algunos ejemplos.

- Iniciar asterisk y conectarse a la CLI en modo verbosidad.
`CORINTO:/home/ovidio# asterisk -cvvv`
Salir de CLI: combinación de teclas Ctrl+C.
- Conectarse a la CLI cuando Asterisk fue iniciado como demonio.
`CORINTO:/home/ovidio# asterisk -vvvr`
Salir de CLI: combinación de teclas Ctrl+C.
- Detener Asterisk sin necesidad de entrar a la CLI.
`CORINTO:/home/ovidio# asterisk -rx "stop now"`

Comandos utilizado en la CLI de Asterisk

Para verificar la cantidad de canales que están activos(llamadas en curso), ver los nombres de los usuarios con protocolos IAX ó SIP y los parámetros de cada unos de ellos. es necesario conocer algunos de los comando mas utilizados dentro de la CLI de Asterisk. A continuación se presentan algunos de ellos.

Comando	Descripción
help	Despliega todos los comandos que se pueden utilizar en la CLI.
reload	Siempre que se modifica un archivo de configuración .conf es necesario ejecutar este comando, de lo contrario no aplicara los cambios. (No corta las llamadas en curso)
restart now	Igual que <i>reload</i> a excepción que este comando corta todas las llamadas en curso.
stop now	Detiene el servidor Asterisk y cierra la CLI.
show dialplan	Muestra en la CLI como esta estructurado el archivo extesions.conf
show channels	Muestra los canales que están siendo utilizados (usuarios activos), de todos los protocolos.
show codecs	Se pueden ver todos los codecs que soporta Asterisk
show voicemail users	Ver todos los usuarios que tiene servicios de buzón de voz.
sip show channels	Muestra los canales que están siendo utilizados (usuarios activos), solo del protocolo SIP. Al sustituir la palabra sip por iax2 vemos los canales utilizados por los usuarios iax2, esto se aplica para todos los comandos que inician con sip.
sip show channel <i>name</i>	Muestra el estado de un usuario sip, donde <i>name</i> es el nombre del usuario.
sip show user <i>name</i>	Ver los parámetros de configuración de un usuarios de nombre <i>name</i> .
sip show users	Podemos ver todos los usuarios sip que forman parte de la central telefónica.
quit, exit , Ctrl + C	salir de la CLI.

TABLA 4.2 Comandos utilizados dentro de la CLI de Asterisk.

Con los conocimientos obtenidos hasta este momento ya se puede iniciar la configuración de Asterisk. No hay mejor forma de aprender a configurar Asterisk, que realizar un pequeño ejemplo documentado de una aplicación real. Se crearán dos usuarios dentro de una red LAN con el protocolo de señalización SIP y se cumplirá lo siguiente.

- Llamadas ilimitadas entre ellos.
- Call ID.
- Música en Espera .
- Buzón de voz.

Para realizar esta aplicación es necesario configurar los archivos `sip.conf`, `extensions.conf` y `voicemail.conf`.

En el archivo `sip.conf` se especifican los parámetros de los usuarios que utilizan el protocolo de señalización SIP, si hubiera algún usuario utilizando el protocolo IAX se tendrían que editar sus parámetros en el archivo `iax.conf`. El plan de llamada o dial plan se detalla en el archivo `extensions.conf`. La clave y la extensión del buzón de voz se especifica en el `voicemail.conf`.

Requerimientos:

- 3 Computadoras:
Una PC será el servidor (Asterisk) y las otras dos clientes (softphone), el servidor debe trabajar con el SO Debian.
- Copiar un archivo `.mp3` en el directorio `/usr/share/asterisk/mohmp3` para que se reproduzca cuando se utilice la música en espera.

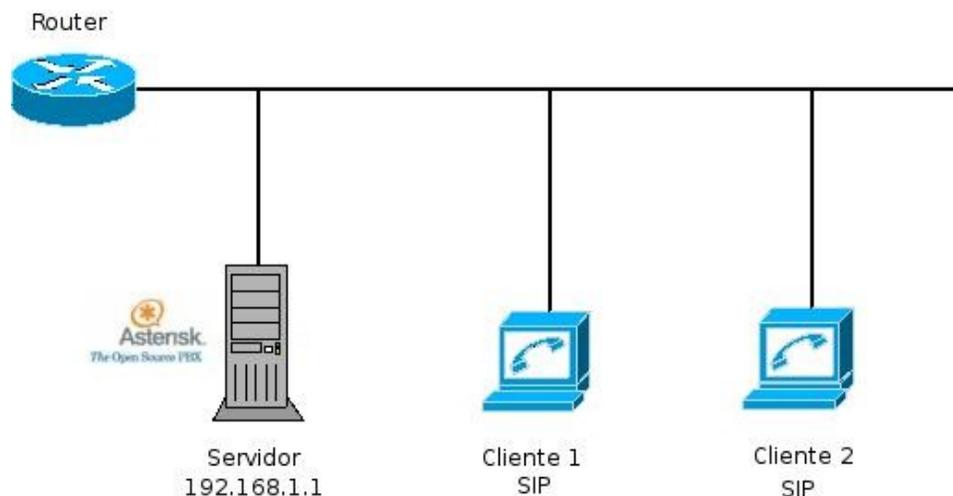


Figura 4.1 Implementación de una centralita telefónica VoIP con Asterisk, para dos usuarios (softphone).

Los dos clientes debe tener instalados softphones que soporte el protocolo SIP (twinkle, ekiga, X-lite, etc), no importa que sistema operativo utilice cada usuario.

Pasos para configurar la centralita telefónica:

1. Editar los parámetros de los usuarios SIP, en el archivo sip.conf

```
[general] ; son parámetros comunes para todos los usuarios.
srvlookup=yes ; permite establecer llamadas en nombre de dominio.
canreinvite=yes ; permite intercambiar mensajes RTP directamente.
bindport=5060 ; puerto que utiliza SIP para comunicarse con asterisk.
```

```
[cliente1]
type=friend ; puede hacer y recibir llamadas.
username= cliente1 ; nombre del usuario.
secret= cliente1pass ; clave del usuario.
callerId= cliente1 ; nombre que aparecerá cuando se esta llamando.
host=dynamic ; la IP de la maquina es dinamica.
context=interno ; al contexto que pertenece en extensions.conf.
mailbox=111@mail_box ; la extensión de buzón de voz.
nat=no ; se encuentra detrás de nat.
disallow=all ; deshabilita todos los codec de audio.
allow=all ; habilita todos los codec de audio.
```

```
[cliente2]
type=friend
username= cliente2
secret= cliente2pass
callerId= cliente2
host=dynamic
context=interno ; ambos usuarios pertenecen al mismo contexto.
mailbox=222@mail_box
nat=no
disallow=all
allow=all
```

2. Crear el plan de llamada ó dial plan en el archivo extensions.conf.

```
[interno] ; Inicia el contexto interno
; definir los parámetros de los usuarios que
; pertenecen a este contexto.
exten => 123,1,Dial(SIP/cliente1,10) ; extensión de cliente1 123. prioridad 1.
exten => 123,2,Playback(vm-sorry) ; sonido al no contestar en 10s. prioridad 2.
exten => 123,3,Voicemail(111@mail_box) ; extensión de buzón de voz. prioridad 3
exten => 123,4,HangUp() ; cuelga la llamada. al pasar por las 3
; prioridades anteriores.

exten => 124,1,Dial(SIP/cliente2,10)
exten => 124,2,Playback(vm-sorry)
exten => 124,3,Voicemail(222@mail_box)
exten => 124,4,HangUp()
```

exten => 8080,1,VoicemailMain(@mail_box) ; extensión para escuchar los mensajes.
 exten => 5050,1,MusicOnHold(default) ; extensión de musica en espera.

3. Definir claves y extensiones para el buzón de voz.

[mail_box] ; contexto de correos.
 attach=yes ; adjuntar archivos.
 format=wav ; formato del mensaje de voz (mp3, gsm)
 111 => 111,cliente1,cliente1@gmail.com ; extensión => clave, cliente y email.
 222 => 222,cliente2,cliente2@yahoo.com

4. Configurar los softphone[Apéndice B].

Una vez configurados los archivos, se ingresa a la CLI de Asterisk y se utiliza el comando *reload* para aplicar todos los cambios.

En este momento ya se podrá realizar llamadas entre los usuarios !!

Para que se puedan enviar los mensajes de voz al correo electrónico de cada usuario es necesario tener:

- Una IP publica en el servidor.
- Asignar un DNS al servidor (ejemplo http://central.sip.dyndns.org).
- Y por ultimo es necesario tener instalado un servidor de mensajería (exim)

En los apartados siguientes se explicará detalladamente cada uno de los parámetros de los archivos configurados en el ejemplo.

4.2.1. sip.conf.

En este archivo se especifican todos los parámetros de configuración de los usuarios que utilizan el protocolo de señalización SIP [1,2]. Los parámetros se pueden diferencian en dos grupos: los generales que son comunes para todos los usuarios SIP y los específicos de cada usuario. A continuación se explican algunos de ellos.

Parámetro	Descripción
srvlookup	Los registros DNS SVR son una manera de establecer una dirección lógica donde pueda ser alcanzado. Esto permite a las llamadas puedan ser reenviadas a diferentes localizaciones sin la necesidad de cambiar la dirección lógica. El uso de DNS SRV es recomendado.

TABLA 4.3 *Parámetros generales de los usuarios sip.conf.*

Parámetro	Descripción
maxexpiry	Establece el tiempo máximo en segundos para registra un usuarios. por defecto se utiliza 3600, cuando el tiempo se agota el usuario queda fuera de uso.
canreinvite	Se recomienda establecer este parámetro en yes, de lo contrario obliga a Asterisk que los puntos finales no intercambien mensajes RTP directamente.
bindaddr	Permite controlar la interfaz IP, con 0.0.0.0 le decimos a Asterisk que escuche en todas las direcciones IP de la red.
bindport	Establecemos el puerto por medio del cual deseamos aceptar la conexión SIP. El puerto usado por SIP es 5060.
rtptimeout	Numero de segundos para terminar una llamada si no hay tráfico RTP. por defecto se establece 60.

TABLA 4.3 (continuación).

Parámetro	Descripción
type	Es un de los campos más importantes y puede tener el valor: <i>friend</i> , <i>user</i> y <i>peer</i> . <i>friend</i> : Puede realizar y recibir llamadas. <i>user</i> : Solo puede recibir llamadas. <i>peer</i> : Solo puede hacer llamadas.
username	Se especifica el nombre del usuario. Este campo se necesitara para configurara el softphone.
secret	Asigna una clave al usuario. Este parámetro también se necesita para poder confirmar el softphone.
callerId	Es un identificador de llamada.
host	Permite establecer la IP o host del usuario, si se trabaja en un con DHCP se recomienda establecer este parámetro con el valor: <i>dynamic</i> para que aprenda la IP de la maquina del usuario automáticamente cada vez que se cambia.
context	especifica a que contexto del dialplan (<i>extensions.conf</i>) pertenece el usuario.

TABLA 4.4 *Parámetros a configurar en el área de usuarios del archivo sip.conf.*

Parámetro	Descripción
mailbox	Se especifica que el usuario dispone de un correo de voz, especificamos la extensión y el contexto del correo.
nat	Indica si se está detrás de NAT. (yes o no).
disallow	Deshabilita los codec. Pueden especificarse los siguientes campos: <i>all, ulaw, alaw, g726, g729, gsm, ilbc, etc.</i>
allow	Habilita los codec. Pueden especificarse los siguientes campos: <i>all, ulaw, alaw, g726, g729, gsm, ilbc, etc.</i>
dtmfmode	Modo en el cual se transmiten los tonos DTMF. los valores pueden ser: <i>inband</i> (se transmiten como parte del audio stream), <i>info</i> o <i>rfc2833</i> (fuera de banda). La opción recomendada es <i>rfc2833</i> .

TABLA 4.4 (continuación).

A parte de los parámetros descritos anteriormente hay una gran variedad [1,2] que no se especificaran en este trabajo, los parámetros que no se especifican en los archivos de configuración toman valores por defecto.

4.2.2. iax.conf.

Cuando un usuario utiliza el protocolo IAX2, es en el archivo *iax.conf* donde se especificaran todos los parámetros del usuario. Los parámetros de configuración son similares a los utilizados en *sip.conf*, pero hay algunos campos distintos. La parámetros están agrupados de la misma forma: generales y específico.

Comando	Descripción
bindaddr	Permite controlar la interfaz IP, con 0.0.0.0 le decimos a Asterisk que escuche en todas las direcciones IP de la red.
bindport	Establecemos el puerto por medio del cual deseamos aceptar la conexión IAX. El puerto usado por SIP es 5036.
bandwidth	Podemos especificar un nivel de ancho de banda. Puede tomar los siguientes valores: <i>low, medium, high</i> . Se ajusta automáticamente con los codec utilizados.
jitterbuffer	Permite configurar el jitterbuffer. Dejar este parámetro en <i>jitterbuffer=no</i> .

TABLA 4.5 Parámetros generales de los usuarios *iax.conf*.

Parámetro	Descripción
trunkfreq	Establecemos el tiempo en milisegundo entre mensajes cuando se utiliza trunking. EL valor predeterminado es de 20 ms y debería ser bueno para la mayoría de las instalaciones .

TABLA 4.5 (continuación).

Comando	Descripción
type	Igual que sip.conf
username	Igual que sip.conf
secret	Igual que sip.conf
callid	Igual que sip.conf
host	Igual que sip.conf
trunk	Permite utilizar el mecanismo de trunking, se recomienda establecer en <i>trunk=yes</i> .
tos	Puede fijar el tipo de servicio (TOS). bits en el encabezado IP para ayudar a mejorar rendimiento de los router. Los valores disponibles son: <i>lowdelay, throughput, reliability, mincost, none</i> .

TABLA 4.6 Parámetros a configurar en el área de usuarios del archivo *iax.conf*.

Todos los parámetros de configuración de los archivos *sip.conf* e *iax.conf* se pueden ver en [1,2] los libros que están dedicados a la configuración de Asterisk.

4.2.3. extensions.conf.

Este es uno de los archivos de configuración más importantes de Asterisk[1,2], es en este archivo donde se configura el plan de llamada (*dialplan*). El dialplan es realmente el corazón de Asterisk y define la forma como se manejaran las llamadas entrantes y salientes de cada usuario, en otras palabras consiste en una lista de instrucciones o paso que seguirá Asterisk. Para poder crear un sistema con Asterisk se tiene que entender el dialplan.

Un dialplan está formado por cuatro parte principales: *contextos, extensiones, prioridad y aplicaciones*.

Cada una de las partes del dialplan se discutirán a continuación, se recomienda ir consultado el ejemplo anterior para tener una idea más clara.

4.2.3.1. Contextos.

Un contexto se define como un grupo de usuarios, donde cada usuario tiene su propia extensión. De la misma forma se puede decir que un contexto es un grupo de extensiones[1,2].

El inicio de un contexto se indica poniendo el nombre del contexto entre corchetes “[interno]”. Todas las instrucciones colocadas después de una definición de contexto son parte de ese contexto, hasta el siguiente contexto que se define.

Uno de los usos más importantes de contextos es hacer cumplir la seguridad. Mediante el uso de contextos, se pueden dar algunas características que no están a disposición para todos los usuarios. Esto se logra ya que solo pueden realizar llamadas entre extensiones que se encuentran en el mismo contexto. Para poder realizar llamadas a extensiones de otros contextos es necesario incluir los contextos.

```
[interno]
include => externo
exten => 1,1,Answer();
```

```
[externo]
exten => 2,1,Playback(tt-weasels)
```

Con el comando *include* se incluye el contexto externo en el contexto interno, y de esta forma todas las extensiones que están definidas en el contexto externo pueden ser alcanzadas por las extensiones del contexto interno.

4.2.3.2. Extensiones.

Las extensiones serán instrucciones que Asterisk ejecutara[1,2], desencadenada por una llamada entrante o por algún dígito marcado en un canal. Las extensiones especifican que hacer con las llamadas conforme hace su recorrido en el dialplan.

Las extensiones se pueden utilizar para especificar las extensiones en los teléfonos tradicionales (Llamar al cliente1 al 123 y al cliente2 al 124).

En Asterisk se pueden definir extensiones para: realizar llamadas, buzón de voz, música en espera. También existen extensiones reservadas que utiliza Asterisk. Algunas de estas son:

- s : Extensión de inicio. Si ningún otro número de extensión se introduce, entonces esta extensión es la que se ejecuta.
- t : Tiempo de prorrogas. Si un usuario está obligado a dar entrada, pero no lo hace con la suficiente rapidez, se ejecuta esta extensión .
- i : Extensión inválida. Si un usuario introduce una extensión inválida, se ejecuta esta extensión.

Las extensiones se especifican con la sintaxis.

`exten => 123`

Una extensión no puede declararse sola, es necesario que vaya acompañada de la *prioridad y aplicación*.

La sintaxis completa para declarar las extensiones es:

`exten => extensión, prioridad, aplicación.`

la extensión puede tener un valor alfanumérico (123, 12ovidio, carlos1, raul, etc).

4.2.3.3. Prioridades.

Las extensiones pueden tener múltiples pasos, llamados prioridad, las prioridades están enumeradas secuencialmente, iniciando con 1. Cada prioridad ejecuta una aplicación específica.

```
exten => 123,1,Voicemail(111@mail_box) // Inicia con el buzón de voz.
exten => 123,2,HangUp() // Corta la llamada.
```

También está la prioridad “n” la cual va sumando uno automáticamente.

```
exten => 123,1,Answer() // Primero contesta (1).
exten => 123,n,Voicemail(111@mail_box) // Inicia con el buzón de voz (2).
exten => 123,n,HangUp() // Corta la llamada (3).
```

4.2.3.4. Aplicaciones.

Finalmente se llega a la aplicación y es aquí donde se le dice a Asterisk que es lo que realmente queremos realizar. Algunas de las aplicaciones más útiles son.

Comando	Descripción
Answer()	Contesta la llamada automáticamente.
Playback(filename)	Reproduce un archivo .wav o .gsm. Los archivos disponibles se encuentran en /usr/share/asterisk/sound
Background(filename)	Similar a Playback(), a excepción que está esperando una entrada por parte del usuario.
Goto(contexto, extension, prioridad)	Permite realizar saltos.

TABLA 4.7 Aplicaciones utilizadas en el dialplan de Asterisk[2].

Comando	Descripción
Voicemail(extension@contexto)	Transfiere una llamada a un correo de voz
VoicemailMain(extension@contexto)	Permite el acceso al buzón de voz. Puede escuchar todos los mensajes que le han enviado.
MusicOnHold(default)	Permite dejar con música en espera, a un usuario para atender otra llamada.
Dial(protocolo/usuario, tiempo de espera, opción)	Esta aplicación es la que permite realizar llamadas. <i>tiempo de espera:</i> Es el tiempo en segundos que estará dando tonos de llamada antes que pase a la siguiente prioridad. <i>opción:</i> La mas utilizada es "r", el que llama escuchará un tono de llamada, mientras que el canal de destino es notificado de una llamada entrante. El uso no es indispensable ya que Asterisk genera los tonos automáticamente.

TABLA 4.7 (continuación).

4.2.4. voicemail.conf.

Los correos de voz son una de las características más populares en los sistemas de telefonía moderna. Naturalmente Asterisk tiene un flexible sistema de correos de voz. Al igual que *extensions.conf* en el *voicemail.conf* se definen contextos y tiene la misma función.

Se mencionaran alguno parámetros de configuración que son muy útiles.

- attach : Permite adjuntar los archivos cuando se envía al email. (yes, no).
- format: Establece el formato del archivo de audio (wav, wav49, gsm).
- maxmessage: Longitud máxima del mensaje en segundos.

al crear un correo de voz se tiene que definir dentro de un contexto, y se especifica de la siguiente forma:

voicemailbox => password, username, emailaddress, pageraddress, options

- voicemailbox: Es el numero del buzón de voz ó extensión del buzón de voz. Tiene que ser un valor numérico.

password:	Clave para ingresar al buzón de voz. Tiene que ser un valor numérico.
username:	Es el nombre del propietario del buzón de voz, este valor tiene que ser igual al <i>username</i> del archivo <i>sip.conf</i> .
emailaddress:	Dirección email a la cual se enviaran los correos de voz, siempre que se tenga un servidor de mensajería SMTP.

A continuación se especifica un buzón de voz, con sus parámetros de configuración.

222 => 222,cliente2,cliente2@yahoo.com.

4.3. Conclusiones.

Se revisó como están distribuidos en el sistema operativo los principales directorios utilizados por Asterisk, el directorio mas importante de Asterisk es el contiene todos los archivos de configuración(*/etc/asterisk*). Del total de archivos de configuración, que son aproximadamente cincuenta, únicamente se trabajo en el *sip.conf*, *extensions.conf* y *voicemail.conf*. También se describieron algunos de los parámetros que se utilizan para crear usuarios SIP e IAX.

Al trabajar con Asterisk es necesario conocer alguno de los comandos mas importantes utilizado en la CLI, por medio de los cuales se puede verificar el estado de las llamadas, cantidad de usuarios sip, parámetros de usuarios. etc

Para comprender como funciona Asterisk se realizó un pequeña ejemplo, en el cual se desarrolló un dialplan básico. Para realizar el dialplan fue necesario configurar directamente en los archivos *sip.conf*, *extensions.conf* y *voicemail.conf*.

CAPITULO V

IMPLEMENTACIÓN DE LA HERRAMIENTA DE CONFIGURACIÓN

El capítulo anterior se dedicó por completo a la configuración de Asterisk, se explicaron algunos de los parámetros utilizado para configurar los archivos sip.conf, iax.conf, extension.conf y voicemail.conf. También se explicó como esta estructurado un dialplan ó plan de llamada y las partes que lo forma. Se realizó un pequeño ejemplo en el cual se configuraron directamente los archivos de Asterisk para crear dos usuarios SIP.

En este capítulo se presenta una herramienta que nos permitirá configurar Asterisk a través de un navegado web. Con el uso de la herramienta se podrá: Crear y eliminar usuarios que utilizan el protocolo SIP, ver los parámetros de configuración de todos los usuarios, crear contextos, se tendrás acceso a un directorio telefónico donde se puede ver la cantidad de usuarios y extensiones que tiene asignado cada uno de ellos, musica en espera, buzón de voz y otros.

La herramienta de configuración tiene dos tipos de menú, uno para el administrador y otro para los usuarios. Todos los usuarios pueden ver y/o modificar algunos de sus parámetros de configuración. Solamente el administrador tiene los privilegio de crear y borrar usuarios, ver y/o modificar todos los parámetros.

5.1. Descripción del funcionamiento.

Esta herramienta fue creada para facilitar la administración de una central telefónica VoIP que utiliza Asterisk como PBX. De no utilizar esta herramienta, se tendrían que configurar manualmente los archivos de Asterisk en la computadora que funciona como centralita telefónica. Una de las ventajas de configurar Asterisk con la herramienta, es la flexibilidad que proporciona al permitir hacerlo desde cualquier computadora que forme parte de la red LAN utilizando un navegador web.

Con la herramienta únicamente se pueden configurar usuarios con protocolo SIP. Al entrar a la página de la herramienta, solicitará el *usuario* y la *clave*. Al ser superusuario tendremos acceso a todas las opciones, mientras un usuario norma, simplemente tendremos acceso al directorio telefónico y algunos de sus parámetros de configuración(CallID, tiempo de espera y otros).

La configuración de Asterisk manualmente es recomendable cuando la cantidad de usuarios es pequeña, si hay muchos usuarios se vuelve complicado para el administrador de la centralita, esto sucede ya que el administrador debe revisar el archivo sip.conf y extensions.conf para cerciorase que no hayan usuarios con el mismo nombre o extensiones repetidas. La herramienta de configuración verifica que los usuarios y extensiones creadas no sean repetidas, facilitando la administración de la centralita telefónica.

La mejor forma de comprender el funcionamiento general de la herramienta es a través de un diagrama, como se muestra en la figura 5.1

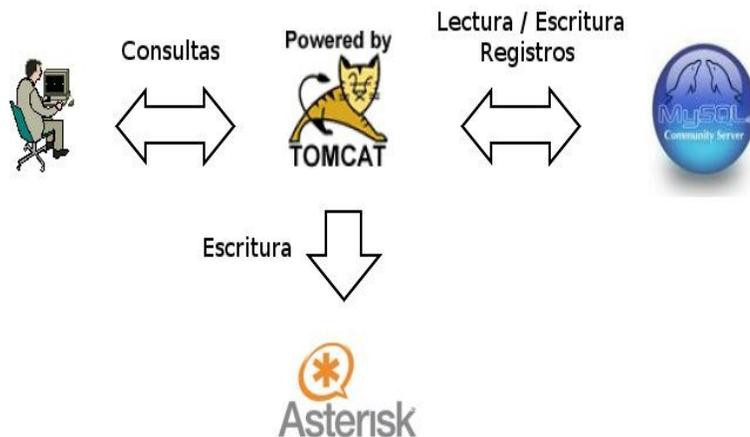


Figura 5.1 *Diagrama de funcionamiento de la herramienta de configuración.*

En el la figura 5.1 se puede ver como toda la información de los usuarios es guardada en la base de datos Mysql. Cuando se crea un nuevo usuarios o se modifican los parámetros de los existentes, se lee la base de datos para no permitir nombres de usuarios y extensiones repetidas, posteriormente se procede a escribir los parámetros en los archivos de Asterisk. En uno de los siguientes apartados se explicaran las tablas que contiene la base de datos que se utiliza en esta herramienta.

La herramienta de configuración se diseño utilizando Java como lenguaje de programación e implementando una arquitectura de n-capas. Una arquitectura n-capas, secciona todo el sistema en distintas unidades funcionales: cliente, presentación, lógica-de-negocio, lógica-de-datos y lógica-de-dominio. Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica-de-negocios. En la figura 5.2 se puede ver el esquema de la herramienta desarrollada con Java que implementa el método de n-capas.

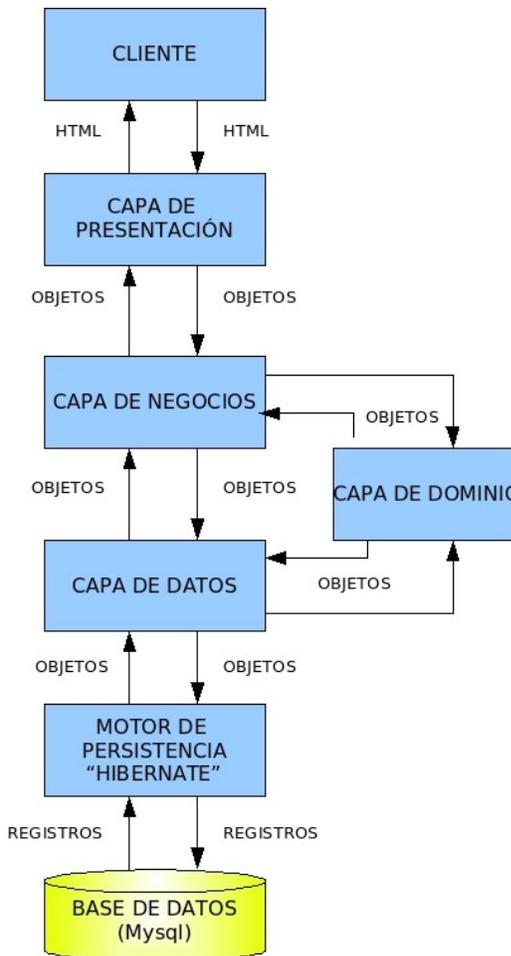


Figura 5.2 Esquema de la herramienta de configuración, desarrollada con Java, implementado el método den-capas.

DESCRIPCIÓN DE CAPAS

CAPA DE PRESENTACIÓN: En esta capa edificamos las paginas HTML y JSP, las cuales nos va a permitir interactuar con el cliente por medio de un navegador (mozilla , epiphany, iceweasel, firefox etc)

CAPA DE NEGOCIO: En esta capa se colocaran las clases java, las cuales definirán la lógica de negocio. Son todas las opciones a las que el cliente tiene accesos (crear un usuario, borrar un usuario etc..).

CAPA DE DATOS: En esta capa se colocaran las clase de java que interactuaran con la base de datos a través de hibernate (envía todos los objetos tipo usuario, recupera todos los usuarios etc..).

En estas capa se utiliza lenguaje HQL, el cual es interpretado por hibernate.

CAPA DE DOMINIO: Esta capa define las capas propias del negocio, estas clases son las que nos permiten crear objetos por medio de los cuales la capa de negocios implementa sus funciones, Se definen métodos SET (con argumento) y GET (sin argumento) que nos permitirán recuperar y guardar los objetos a través de hibernate.

5.2. Configuración de la herramienta.

En el Capítulo III se instalaron todos las utilidades para que pueda funcionar correctamente nuestra herramienta. Dentro del CD que se entregó del presente trabajo de graduación se encuentra el paquete *centralita.tar.gz.*, Descomprimirlo y entrar a la carpeta *software*, abrir el documento *LEERME_POR_FAVOR.pdf* que contiene todos los pasos de configuración, incluyendo los realizados en el Capítulo III. Para que la herramienta pueda funcionar correctamente solo hace falta realizar los pasos 1.6, 1.7 y 1.8 del documento *LEERME_POR_FAVOR.pdf.* *ii hacer un respaldo de todos los archivos de configuración de asterisk que se encuentran en /etc/asterisk, antes de utilizar la herramienta por primera vez, ya que sobrecribiré los archivos!!*

Primero iniciamos los servidores Asterisk, Mysql y Tomcat usando el script *startpbx.sh*. Antes de usar el script, asegurarse que los servidores no estén iniciados.

CORINTO:/home/ovidio# /usr/local/startpbx.sh // tienes que ser root.

```
*****  
* CONGRATULATION PBX IS READY TO BE USED *  
*****
```

Abrimos un navegador y editamos la URL http://localhost:8080/central_usuario:admin
usuario: admin
clave: admin



Figura 5.3 Pagina de bienvenida de la herramienta de configuración.

Si desea puede crear algunos usuarios, seleccionar la opción **Reload** que se encuentra en la pagina principal, para configurar Asterisk. Y aparecerá un mensaje **Asterisk Fue Configurado!!**. Si hubo algún problema en la configuración dará el mensaje **Error -- En la Configuración !!**



Figura 5.4 Menú del administrador de la herramienta de configuración.

Para detener la herramienta ejecutamos el script `stoppbx.sh`

```
CORINTO:/home/ovidio# /usr/local/stoppbx.sh
```

```
*****
* THE PBX IS OUT OF SERVICE *
*****
```

Mas adelante se explicaran todas las opciones del superusuario.

5.3. Directorios de la herramienta.

El paquete `centralita.tar.gz` incluye el archivo `Herramienta_central.pdf`, donde se especifica como están distribuidos todos los directorios de la herramienta. A continuación se nombran y explican cada uno de los directorios de `central.war`:

imágenes: Contiene todas las imágenes utilizadas en la herramienta.

- META-INF:** Únicamente tiene el archivo `context.xml`, donde se especifica el nombre y el password del usuario de la base de datos.
- `.jsp`,`.html`: Estos son todos los archivos por medio de los cuales se les da formato a los datos de entrada y salida desde y hacia el cliente web.
- WEB-INF:** *build.xml*, Este archivo permite compilar todas las clases de java. (utilizar solo si se modifica alguna de las clases).
- web.xml*, En este archivo se le especifica al servidor tomcat con que que pagina iniciara la aplicación.
- lib*, Este directorio contiene todas las librerías que se necesitan para que funcione correctamente la herramienta, en este directorio se encuentran todos los `.jar` de HibeMate.
- src*, Esta carpeta contiene todo el código fuente de las clases de java.
- classes*, Todos los `.class` de java (ejecutables) se encuentran dentro de la carpeta *voip*, también se encuentran los `.hbm.xml` y `.cfg.xml` donde esta especificada la configuración de HibeMate.

5.4. Base de datos y sustablas.

Como se vio en la figura 5.1, la herramienta utiliza Mysql para almacenar y consultar los parámetros de los usuarios.

Todos los datos son guardados en las cinco tablas que forma la base de datos *central*, a continuación se explica que parámetros contiene cada tabla.

BASE DE DATOS : CENTRAL

tabla: usuarios.

```
id int;
nombre varchar ;
apellido varchar;
sexo char;
nacimiento varchar;
protocolo varchar;
usuario varchar;
clave varchar;
```

tabla: pgsip.

```
id int;
srvlookup varchar ;
canreinivite varchar;
bindaddr varchar;
bindport varchar;
maxexpiry int;
rtptimeout int;
```

Figura 5.5 Tablas que forman parte de la base de datos “central”.

tabla: contexts.

```
id int;  
context varchar ;
```

tabla: icontexts.

```
id int;  
superior varchar ;  
inferior varchar;
```

tabla: sip.

```
id int;  
tipousuario varchar ;  
extensión int;  
tiempoespera int;  
callid varchar;  
host varchar;  
context varchar;  
mailbox varchar;  
extensionmailbox varchar;  
nat varchar;  
outgoinglimit int;  
incominglimit int;  
dtmfmode varchar;
```

Figura 5.5 (continuación).

tabla usuarios:	Almacena los datos personales de los usuarios.
tabla pgsip:	Almacena los parámetros generales sip.
tabla contexts:	Almacena los contextos existentes.
tabla icontexts:	Almacena todos los contextos que están incluidos en otros contextos.
tabla sip:	Almacena todos los parámetros de configuración de los usuarios.

5.5. Tipos de menú.

La herramienta de configuración tiene dos tipos de menú, uno para los usuarios y el otro para el administrador de la centralita telefónica. Se diseñó así para facilitar el trabajo del administrador, de esta forma los usuarios pueden consultar todos las extensiones disponibles en la central, cambiar su llamada, extensión y clave del buzón de voz, sin tener conocimiento de la configuración de asterisk.

A continuación se explican los dos tipos de menú y las opciones que están disponibles en cada uno de ellos.

5.5.1. Menú de administrador.

El menú del administrador (ver figura 5.4) tiene todos los privilegios, las opciones disponibles son:

Home:	Página principal , disponible un directorio telefónico para consultar la cantidad de usuarios y la extensión que tiene cada uno.
Contextos:	Esta opción permite consultar todos los contextos, crear, eliminar, incluir y excluir contextos.
Configurar Parámetros Generales:	En esta opción podemos consultar y/o modificar todo los parámetros generales de los usuarios.
Ver Parámetros de Usuarios:	En esta opción podemos consultar y/o modificar todo los parámetros de los usuarios de la central telefónica.
Crear Nuevo Usuario:	Como su nombre lo dice, es para crear usuarios.
Borrar Un Usuario:	Elimina un usuarios de la central telefónica.
Reload:	Con esta opción se configuran todos los archivos de Asterisk. Se recomienda utilizar cuando se hayan realizado todos los cambio.
Cerrar Sesión:	Salir de la herramienta de configuración.

5.5.2. Menú de usuarios.

Como se dijo anteriormente el usuario únicamente tendrá acceso a un directorios telefónico, ver y/o modificar algunos de sus parámetros de configuración, no se le permite modificar todos sus parámetros por seguridad. Es el administrador quien decide a que contexto pertenecen y el usuario no debe poder cambiar este parámetro.

En la figura 5.6 se puede ver la pagina del menú de usuarios, no están disponibles las opciones: *Contextos*, *Configurar Parámetros Generales*, *Crear Nuevo Usuario*, *Borrar Un Usuario*.



Figura 5.6 Menú de usuarios de la herramienta de configuración.

Home:

Página principal, disponible un directorio telefónico para consultar la cantidad de usuarios y la extensión que tiene cada uno.

Ver Mis Parámetros :

En esta opción podemos ver y/o modificar algunos de los parámetros de configuración (call id, extensión, clave de voicemail, extensión de voicemail, etc.).

Reload:

Con esta opción se configuran todos los archivos de Asterisk. Se recomienda utilizarla cuando se hayan realicen todos los cambio.

Cerrar Sesión:

Salir de la herramienta de configuración.

5.6. Pruebas y resultados.

Para verificar el correcto funcionamiento de la herramienta de configuración se realizaron dos pruebas. En la primera se implemento una central telefónica para una PYME [Apéndice C]. En la segunda prueba únicamente se instalaron dos antenas para ampliar la cobertura de la central usando el estándar IEEE 802.11.

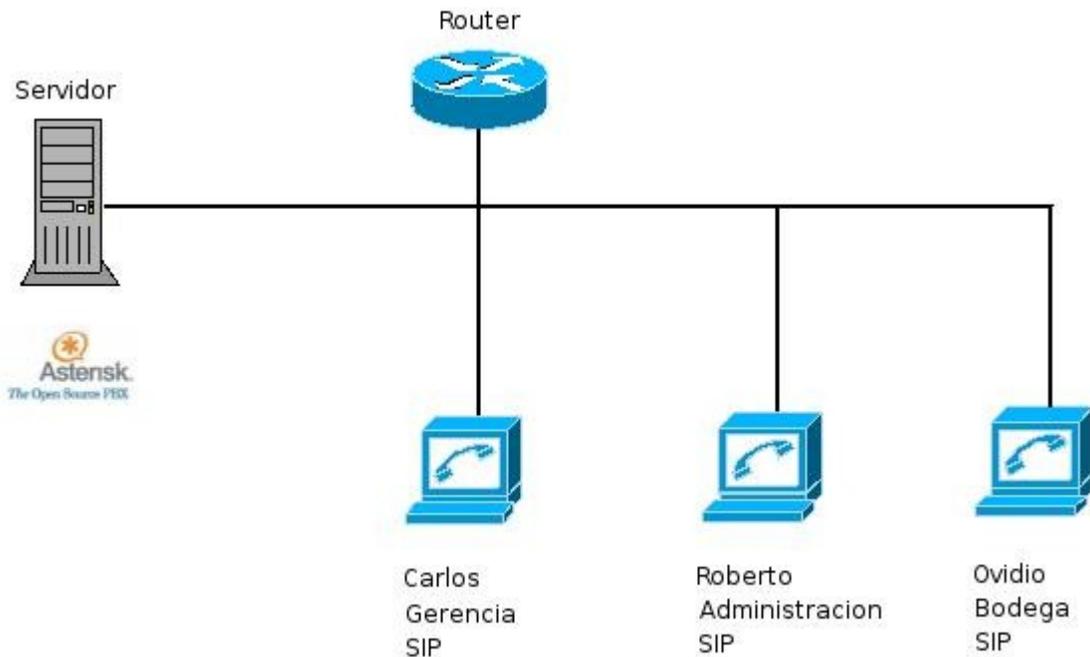


Figura 5.7 Diagrama de la red LAN, en la cual se puso a prueba el funcionamiento de la herramienta de configuración.

RESULTADO DE LAS PRUEBAS						
Empleado	Área de trabajo	Hacer llamadas		Recibir llamadas	Musica en espera	Correo de voz
Carlos	Gerencia	Administración: SI_ NO_	Bodega: SI_ NO_	Administración: SI_ NO_	SI_ NO_	SI_ NO_
		Bodega: SI_ NO_	Gerencia: SI_ NO_	Bodega: SI_ NO_		
Roberto	Administración	Gerencia: SI_ NO_	Bodega: SI_ NO_	Gerencia: SI_ NO_	SI_ NO_	SI_ NO_
		Bodega: SI_ NO_	Gerencia: SI_ NO_	Bodega: SI_ NO_		
Ovidio	Bodega	Gerencia: SI_ NO_	Administración: SI_ NO_	Gerencia: SI_ NO_	SI_ NO_	SI_ NO_
		Administración: SI_ NO_	Gerencia: SI_ NO_	Administración: SI_ NO_		

Tabla 5.1 Resultado obtenidos, después de configurado la centralita telefónica con la herramienta.

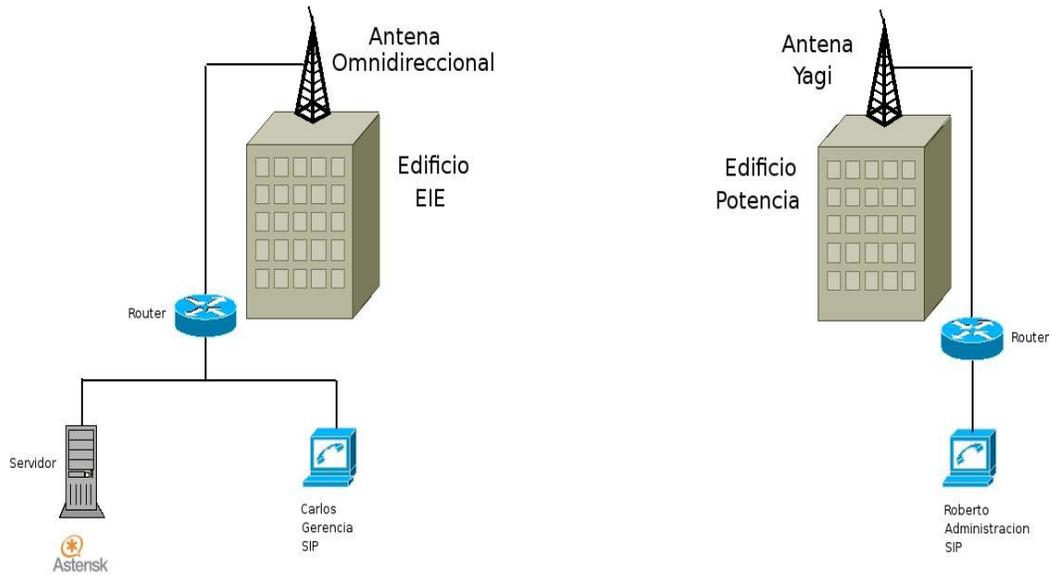


Figura 5.8 Prueba realizada con la herramienta de configuración, haciendo uso del estándar IEEE 802.11 para ampliar la cobertura de la centralita telefónica.

Los resultados de las 2 practicas realizadas con la herramienta de configuración fueron exitosas, las pruebas realizadas en la práctica de la PYME se construyó una tabla en la cual se coloreó con verde el resultado obtenido. Para la práctica donde se amplio la cobertura por medio del estándar IEEE 802.11, se mantuvo la misma configuración en la central, únicamente se verifico que pudiera registrarse el usuario y modificar sus parámetros desde el otro edificio (Potencia) y efectivamente las pruebas fueron también exitosas.

5.7. Conclusiones.

Al combinar los conocimientos de programación con la centralita telefónica Asterisk, se logró desarrollar una herramienta de configuración, que facilita la administración de la central telefónica. Una de las muchas ventajas que presenta esta herramienta es que permite configurar la central telefónica remotamente a través de un navegador web, los usuarios podrán realizar consulta y modificaciones a sus parámetros sin tener conocimiento de la configuración de los archivos de Asterisk. Cuando un usuario realiza modificaciones lo hace de una forma segura, sin poner en riesgo la configuración de los parámetros del resto de usuarios, esto se logra porque tiene acceso únicamente a sus parámetros.

Las funciones disponibles en la herramienta son: creación de usuarios SIP, creación de contexto, permite incluir y excluir contextos, borrar usuarios, modificar parámetros de los usuarios, musica en espera, buzón de voz y un directorio telefónico.

La herramienta de configuración desarrollada en este trabajo de graduación es versátil. AL utilizar el modelo de n-capas facilita su modificación para implementar nuevas funciones.

En el marco legal no se tendrá ningún tipo de inconvenientes al utilizar esta herramienta, ya que fue desarrollada completamente con software libre. No requiere de ningún tipo de licencia para su uso.

CONCLUSIONES Y LINEAS FUTURAS

En este trabajo de graduación se implementó una herramienta de configuración para facilitar la administración de una central telefónica VoIP basada en una computadora personal utilizando Asterisk como PBX.

Después de implementar la herramienta de configuración en la centralita telefónica se llegó a las siguientes conclusiones.

6.1. Conclusiones.

- En la presente tesis, se explico como funciona la VoIP, los protocolos de señalización SIP, H.323, IAX que utiliza. También se explico la importancia de los codec en este tipo de comunicación, uno de los codec mas utilizados es el g729 por su bajo uso de ancho de banda, no se utilizo en la presente tesis por ser de tipo privativo.
- Cuando se iniciaron las primera pruebas de configuración de Asterisk se determino que no es posible utilizar la computadora destinada a realizar las funciones de la central telefónica, como un cliente SIP, simultáneamente, ya que ambos Asterisk y el protocolo SIP utilizan el mismo puerto UDP 5060, no así si se utiliza un cliente IAX, ya que este utiliza el puerto UDP 4569 que difiere del utilizado por Asterisk.
- Verificar que la tarjeta de sonido de todas las computadoras utilizadas, dentro de la central telefónica funciones correctamente, realizar pruebas con el micrófono para estar seguros que no se encuentran apagados. El realizar esta sencillas pruebas, quitaran muchos dolores de cabeza.
- Se implemento una central telefónica con tres usuarios SIP dentro de una red LAN, los cuales fueron creados y configurados con la herramienta desarrollada en la presente tesis. los usuarios tenían acceso a un buzón de voz, musica en espera, se podía consultar un directorio telefónico y sus parámetros por medio de un navegador web.
- Por ultimo con la herramienta concluida se utilizo el estándar IEEE 802.11 para ampliar la cobertura de la centralita. Permitiendo de estar forma registrar y hacer consultas de usuarios, que se encontraban en un edificio distinto.
- Para consultar el buzón de voz es necesario introducir el numero de la extensión y la clave del buzón de voz con intervalos de 1 seg. como mínimo, de lo contrario el softphone no reconocerá los DTMF y negara el ingreso al buzón.

6.2. Lineas Futuras.

- Como primera propuesta, se sugiere la incorporación en la herramienta de más parámetros de configuración generales y específicos del archivo sip.conf. De la misma forma incluir más campos para la configuración del voicemail.conf, entre los cuales se pueden mencionar: *formt*, *attach*, *maxmessage*, algunos de estos campos se encuentran incluidos en la herramienta de configuración pero tiene valores predefinidos.
- También se pueden incorporar muchas más funciones y aplicaciones al dialplan, como la transferencia de llamadas, la sentencia GOTO que permite realizar saltos entre diferentes contextos.
- Implementar una CDR (Call Detail Records) que permita llevar el control del flujo de todas las llamadas realizadas en la centralita telefónica, esto facilitaría enormemente la obtención de datos para realizar la facturación. Para realizar esto se necesita utilizar el archivo */var/log/asterisk/cdr-csv/Master.csv* en el cual se guarda el detalle de todas las llamadas realizadas. Es necesario importar este archivo a una base de datos (Mysql) para que posteriormente se pueda leer toda la información desde la herramienta y presentar en forma gráfica. También necesitara instalar *asterisk-addons-1.2.tar.gz* que incluye el *cdr_mysql.conf* que permitirá realizar la conexión con la base de datos.
- Es necesario expandir la utilidad de la herramienta, permitiendo configurar usuarios que utilicen el protocolo IAX2, para realizar esto simplemente es necesario configurara el archivo *iax.conf* y *extensions.conf* .
- Adquirir un hardware que permita a la centralita telefónica conectarse a la PSTN, similar a la tarjeta PCI de seria TDM400. Para lograr configurar esta tarjeta sera necesario desinstalar Asterisk del servidor y posteriormente instalar en el siguiente orden *zaptel-1.2.tar.gz* , *libpri-1.2.tar.gz* , *asterisk-1.2.tar.gz*, *asterisk-addons-1.2.tar.gz.*, asegurarse que las cabeceras del sistema operativo(linux-headers) estén instaladas previo a la instalación de *zaptel-1.2.tar.gz*.
- Por ultimo, es necesario realizar un estudio para determinar la calidad del servicio (QoS) de la VoIP, realizar pruebas con todos los codec que soporta Asterisk para determina cual es el mas recomendable.

APÉNDICES

APÉNDICE A

Configuración de tarjeta de sonido.

Si tienes problemas con la tarjeta de sonido, a continuación se dará una breve explicación de como resolver este problema, hay muchas formas de como poder configurar correctamente la tarjeta de sonido de tu computadora. aquí se explicara una en particular.

Si todo va bien con tu tarjeta de sonido no tienes porque realizar esta configuración, pero se recomienda que se hagan pruebas con el micrófono, porque muchas veces se encuentra apagado o sin volumen y esta sencilla verificación quitara muchísimos dolores de cabeza cuando se quieran realizar pruebas con Asterisk.

Primero hay que identificar cual es la tarjeta de sonido que tiene instalada el ordenador, esto se hace de la siguiente forma:

```
CORINTO:/home/ovidio# lspci | grep audio
00:11.5 Multimedia audio controller: VIA Technologies, Inc. VT8233/A/8235/8237 AC97
Audio Controller (rev 50)
CORINTO:/home/ovidio#
```

Instalar los paquetes de alsa por medio de los cuales configura la tarjeta de sonido.

```
CORINTO:/home/ovidio# apt-get install alsa-source, alsa-base, alsa-utils
```

Una vez instalada estas utilidades se configura la tarjeta de sonido.

```
CORINTO:/home/ovidio# alsacnf
```



Figura A.1 Inicio de configuración de la tarjeta de sonido utilizando alsacnf.

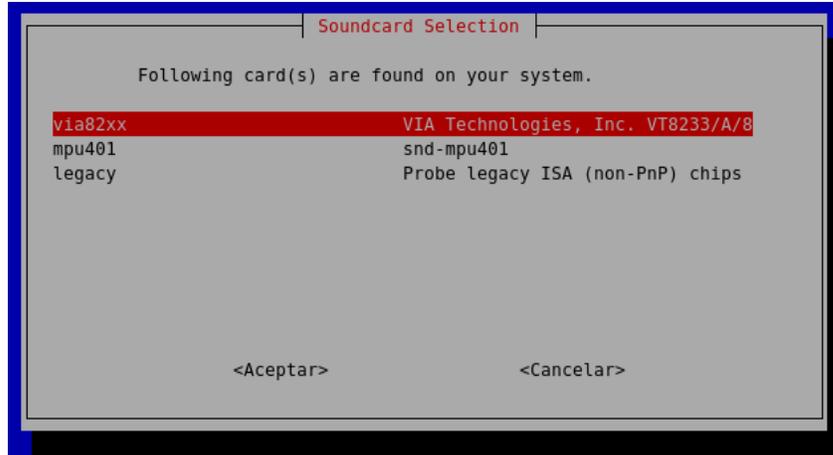


Figura A.2 Tarjetas de sonido disponibles en el sistema.

Seleccionamos la tarjeta de sonido, para nuestro caso en particular “*VIA Technologies*”. Con esto pasos se configura correctamente la tarjeta de sonido, posteriormente a su configuración verificamos que el volumen principal del sistema y el micrófono estén activos.

Para verificar esto se hace con una de las utilidades de alsa (alsamixer).

CORINTO:/home/ovidio# alsamixer

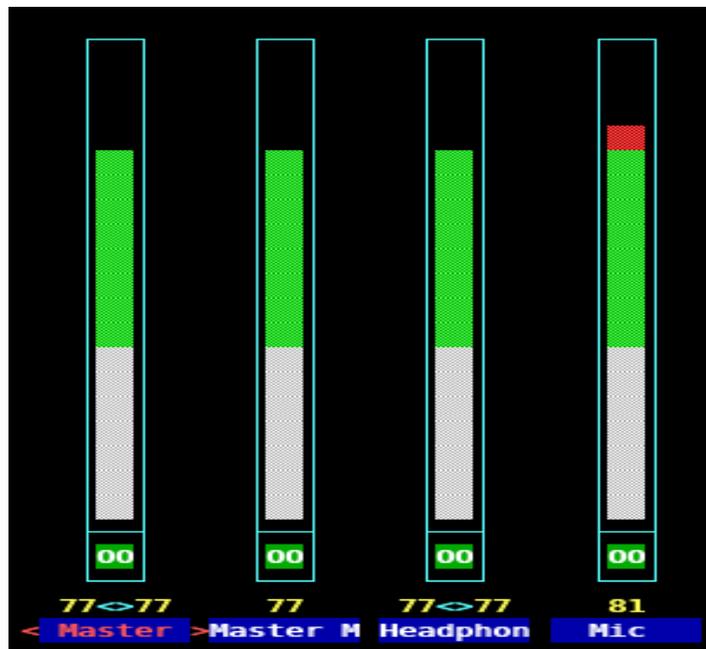


Figura A.3 Control de volumen de los dispositivos de sonido.

Verificar que tengan volumen el MASTER, HEADPHONE(audífono) y MIC(microfono). si en lugar del 00 aparece una letra M en cualquiera de ellos , esta desactivado y tenemos que activarlos posicionándonos en él y presionando la letra m.

APÉNDICE B

Configuración de softphone.

Este apéndice esta dedicado a la configuración de algunos de los softphone's que se utilizaron en este trabajo de graduación. Los softphone a configurar son Twinkle, Ekiga y X-lite. Twinkle y Ekiga se puede instalar con apt-get, X-lite no es propio de debian por lo que se tiene que instalarse manualmente.

Los parámetros que se necesitaran Para configurar los tres softphone son:

- Username y secret de algún usuario (declarado en sip.conf)
- IP del servidor donde esta instalado Asterisk.

a) Twinkle:

Este es un softphone de protocolo SIP que solo puede se utilizado en sistemas operativos GNU/Linux.

Al iniciar el softphone nos pedirá el nombre de usuario para el perfil: editamos uno de los username creados en el sip.conf (cliente1), después editamos los parámetros del usuario.

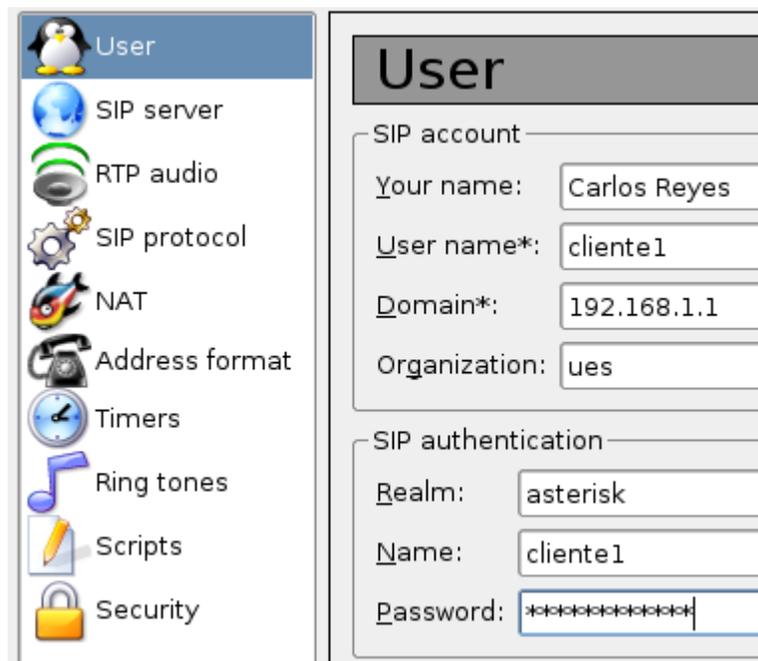


Figura B.1 Parámetros de configuración del softphone Twinkle.

Una vez configurado. aplicamos los cambio y mostrara la pantalla principal del softphone Twinkle. Listo para realizar pruebas.

b) Ekiga:

Al iniciar Ekiga por primera vez, aparecerán 10 pasos para configurar correctamente el softphone, esto incluye al final pruebas de sonido. No es necesario crear una cuenta ya que esto se hace solo cuando se utilizara el servidor @ekiga.net y para correr nuestras aplicaciones con Asteirsk el servidor sera una de los ordenadores que pertenecen al red LAN.



Figura B.2 Pantalla principal del softphone Ekiga.

Entramos a la opción Editar --> Cuentas --> Añadir.

Registrador: IP del Servidor donde se encuentra instalado Asterisk.
Usuario: *username* definido en sip.conf (cliente1).
Contraseña: *secret* definido en sip.conf (cliente1pass).

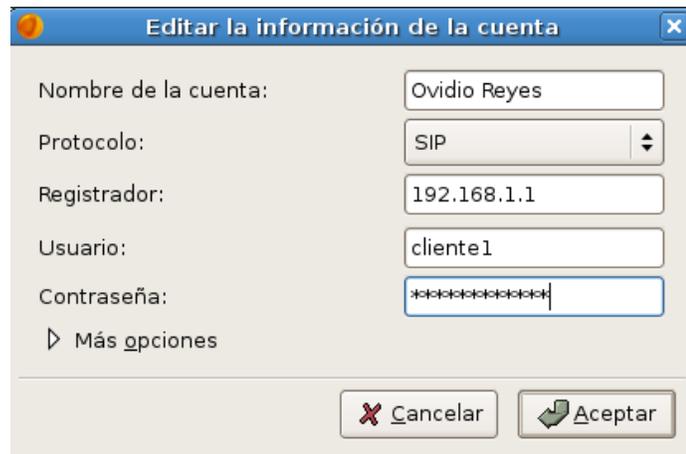


Figura B.3 Parámetros de configuración del softphone Ekiga.

c) X-lite:

Este es uno de los softphone mas amigables y esta disponibles para Window y Linux.



Figura B.4 Pantalla principal del softphone X-lite.

Seleccionamos el botón superior izquierdo y dentro de este seleccionamos la opción: *Sip Account settings... --> Add*

Account Voicemail Topology Presence **Advanced**

User Details

Display Name **Ovidio Reyes**

User name **cliente2**

Password *****

Authorization user name **cliente2**

Domain **192.168.1.1**

Domain Proxy

Register with domain and receive incoming calls

Send outbound via:

domain

proxy Address

target domain

Dialing plan #1\|a|.T;match=1;prestrip=2;

OK Cancel Apply

Figura B.5 Parámetros de configuración del softphone X-lite.

Aplicamos los cambios y concluimos con la configuración de este softphone.

Nota Importante:

Cuando se desee ingresar al buzón de voz es necesario introducir los dígitos de nuestro buzón y clave con intervalos de 1seg. como mínimo entre cada tecla, de lo contrario el softphone no reconoce el valor del DTMF que se introdujo. Dando como resultado no permitir que entremos a nuestro buzón.

En el caso de Ekiga es necesario introducir los dígitos de nuestro buzón y clave desde el propio teclado de Ekiga de lo contrario no reconoce los DTMF.

APÉNDICE C

Sentencias SQL

SQL (Structured Query Language) [26,27,28], es un lenguaje declarativo que utilizan todas las bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

Para poder crear bases de datos, tablas, ingresar los registros, modificarlo, consultar el valor de un registro de un campo específico o borrarlos un registro, en Mysql, es necesario conocer algunos de las sentencias más básicas.

En las siguientes secciones se explicará como crear una base de datos y realizar consultas sobre ella, SQL (no hace diferencias entre mayúsculas y minúsculas).

En primer lugar establezca una conexión en primer plano con Mysql. (como root)
Corinto·#mysql

- Asignar un clave al usuarios root.
mysql>SET password=password('root'); 'root' sera la clave del usuario root.

- Ver las bases de datos existentes.
mysql>SHOW DATABASES;

- Crear una base de datos.
mysql>CREATED DATABASE central;

- Ingresar a una base de datos.
mysql>USE central;

También lo podemos hacer al iniciar mysql
Corinto·#mysql -uroot -proot central

- Crear una tabla que pertenece a la base de datos central
mysql>USE central;
mysql>CREATE TABLE `usuarios` (
`id` int(10) unsigned NOT NULL auto_increment,
`nombre` varchar(60) NOT NULL default "",
`apellido` varchar(60) NOT NULL default "",
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

- Ver las tablas de una base de datos.
mysql>USE central;
mysql>SHOW TABLES;

Recordar que siempre que se va trabajar con las tablas de una base de datos, debe encontrarse adentro de esas base de datos.

- Ver la estructura de una tabla.

```
mysql>DESCRIBE usuarios;
```

- Ingresar datos a una tabla.

```
mysql>INSERT INTO usuarios VALUE(' ','Willian Ovidio','Reyes Medrano');  
mysql>INSERT INTO usuarios VALUE(' ','Carlos Alfredo','Reyes Medrano');
```

- Ver todos los registros de una tabla.

```
mysql>SELECT*FROM usuarios;
```

- Recuperar un campo con un registro especifico.

```
mysql>SELECT nombre FROM usuarios WHERE id=1;
```

- Ordenar una tabla por orden alfabético de su nombre.

```
mysql>SELECT*FROM usuarios ORDER BY nombre;
```

- Modificar el registro de una tabla.

```
mysql>UPDATE usuarios SET nombre='Raul Emigdio' WHERE id = 1;
```

- Eliminar un registro de una tabla.

```
mysql> DELETE FROM usuarios WHERE name='Willian Ovidio';
```

- Eliminar una tabla.

```
mysql>DROP TABLE usuarios;
```

- Eliminar una base de datos.

```
mysql>DROP DATABASE central;
```

Estas son algunas de las sentencias que nos pueden ayudar a verificar los registros de la base de datos.

APÉNDICE D

Practica desarrollada con la herramienta de configuración.

La siguiente practica se realizo para proporcionar una guía de laboratorio de VoIP a la EIE. Con la implementación de la practica se puso a prueba la herramienta de configuración desarrollada en este trabajo de graduación.

La practica se diseño y realizo conjuntamente con un amigo y compañero, Roberto Carlos Romero Castro. Esta guía también esta incluida en el trabajo de graduación: *“Laboratorio de comunicaciones de voz sobre protocolo de Internet y redes inalámbricas”*, Presentada por: Roberto Carlos Romero Castro.

TEMA:

“Implementar una central telefónica VoIP en una PYME, haciendo uso de la herramienta de configuración desarrollada en EIE-UES”

REQUERIMIENTOS:

1) Servicio de Internet.

2) 4 computadoras personales:

Una PC que sera nuestra central telefónica y debe tener instalado el sistema operativo Debian GNU/Linux, las siguientes 3 serán nuestro clientes.

1. las 3 PC restantes serán nuestros clientes:

Deberán tener instalado cualquier softphone que soporten el protocolo SIP (Twinkle, Ekiga o x-Lite (Windows)).

La practica consistirá en asignar extensiones a tres empleados de nuestra PYME, donde cada uno de ellos laboran en área diferente (Gerencia, Administración y Bodega) y se debe cumplir lo siguiente:

Empleado	Area de Trabajo	Hacer Llamadas	Recibir Llamadas
carlos	Gerencia	A todas las áreas	Solo de Administración
roberto	Administración	A todas las áreas	Solo de Gerencia
ovidio	Bodega	A nadie	De Todas las áreas

TABLA D.1 *Condiciones a cumplir de una central telefónica VoIP, implementada en una PYME.*

Esquemáticamente el flujo de llamadas seria como se muestra en la Figura D.1

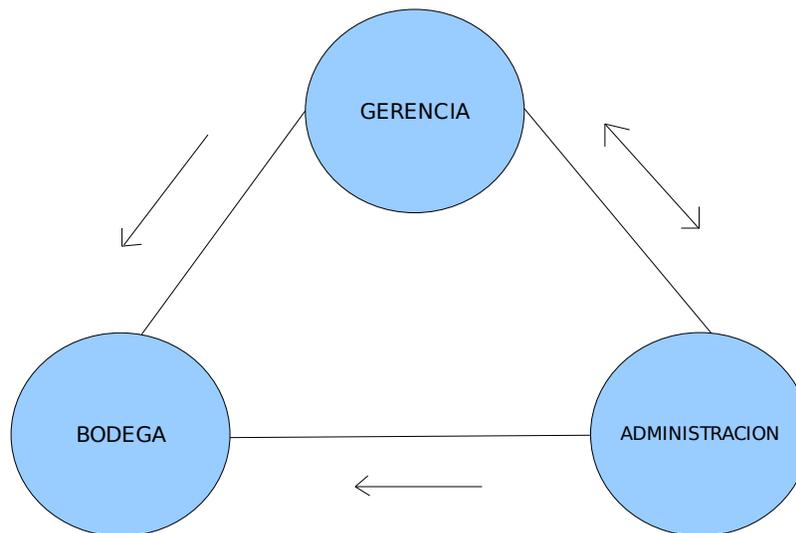


Figura D.1 *Flujo de llamadas disponibles en la PYME.*

En una aplicación real, cada área de trabajo tendrá mas de un persona. Para realizar la practica solo se considera un persona por área.

Para que la central telefónica funciones como se indica en la Tabla D.1 se debe configurar de la siguiente forma:

Usuario	Protocolo	Extensión	VoiceMail	Contexto
carlos	sip	124	111	Gerencia
roberto	sip	125	222	Administración
ovidio	sip	126	333	default

TABLA D.2 *Contextos, extensiones y protocolos a utilizar en la central telefónica de la PYME.*

1) Primero iniciamos los servidores usando el script startpbx.sh (como root), todo esto lo hacemos desde la maquina que funcionara como central telefónica.

CORINTO:/home/ovidio# /usr/local/startpbx.sh

```
*****
* CONGRATULATION PBX IS READY TO BE USED *
*****
```

2) Abrimos un navegador y editamos la URL

<http://localhost:8080/central>

usuario: admin

clave: admin

3) Navegamos a la opción *Contextos* dentro de este *Crear Nuevo Contextos* y creamos los contextos **Gerencia** y **Administración**, el contexto default ya esta incluido por defecto y no lo tenemos que crear

4) Ya creados los contextos vamos a la opción *Contextos* dentro de este *Incluir Contextos Dentro de Otros* e incluimos.

- * Incluir el contexto **default** dentro de **Gerencia**
- * Incluir el contexto **Administración** dentro de **Gerencia**
- * Incluir el contexto **default** dentro de **Administración**
- * Incluir el contexto **Gerencia** dentro de **Administración**

5) Para ver los contextos que hemos creado e incluido seleccionamos *Ver Todos los Contextos*.

6) Posteriormente procedemos a crear los tres usuarios con los siguientes parámetros

Parametros	Usuarios		
	carlos	roberto	ovidio
Tipo de usuario	friend	friend	friend
Extensión	124	125	126
CallId	Gerente	Administrador	Bodeguero
Extensión de MailBox	111	222	333
Clave de MailBox	111	222	333
Contexto	Gerencia	Administración	default
**Extensión de Musica en Espera*	5050	5050	5050
Extensión para escuchar tus Voice Mail**	8080	8080	8080
* solo tiene que copiar un archivo de musica en el directorio /usr/share/astersik/mohmp3 ** La extension de Musica en Espera y Voice Mail ya estan incluidos por defecto en la Opcion Parametros Generales			

Nota: El resto de parámetros los dejamos con la opción por defecto.

TABLA D.3 Parámetros de los usuarios de la central telefónica.

Una vez creado los usuarios, seleccionamos la opción *Reload* que se encuentra en la pagina principal. Y nos aparecerá un mensaje **Asterisk Fue Configurado!!**. Si hubo algún problema en la configuración nos dará el mensaje **Error --- En la Configuración !!**

7) Solo nos hace falta configurar los softphone de las 3 maquinas que serán los usuarios, se necesitaran los siguientes parámetros.

- a) IP de la maquina que representa nuestra central telefónica
- b) El nombre de usuario y su clave.

RESULTADO DE LAS PRUEBAS					
Empleado	Área de trabajo	Hacer llamadas	Recibir llamadas	Musica en espera	Correo de voz
Carlos	Gerencia	Administración: SI__NO__	Administración: SI__NO__	SI__NO__	SI__NO__
		Bodega: SI__NO__	Bodega: SI__NO__		
Roberto	Administración	Gerencia: SI__NO__	Gerencia: SI__NO__	SI__NO__	SI__NO__
		Bodega: SI__NO__	Bodega: SI__NO__		
Ovidio	Bodega	Gerencia: SI__NO__	Gerencia: SI__NO__	SI__NO__	SI__NO__
		Administración: SI__NO__	Administración: SI__NO__		

TABLA D.4 Checklist: Comprobamos la configuración de la central telefónica.

BIBLIOGRAFÍA

- [1] Jim Van Meggelen, Jared Smith, and Leif Madsen. *Asterisk The Future of Telephony*. O'Reilly Media, 2005.
- [2] David Gomillion y Barrie Dempster. *Building Telephony Systems with Asterisk*. Packt Publishing, 2005.
- [3] Mark Spencer, Mack Allison y Christopher Rhodes, The Asterisk Documentation Team. *The Asterisk Handbook*. Digium, 2003.
- [4] Flavio E. Gonçalves. *Asterisk PBX, Guía de la Configuración*. Flavio E. Gonçalves, 2007.
- [5] James Elliott. *Hibernate: A Developer's Notebook*. O'Reilly Media, 2004.
- [6] Dave Minter and Jeff Linwood. *Beginning Hibernate From Novice to Professional*. Apress, 2006.
- [7] James Elliott. *Getting Started with Hibernate 3*. O'Reilly Media, 2006.
- [8] Francesc Rosés Albiol. *Introducción a Hibernate*. Francesc Rosés Albiol, 2004.
- [9] Christian Bauer, Gavin King, foreword by Linda DeMichiel. *Java Persistence with Hibernate* Manning Publications, 2007.
- [10] Héctor Suárez Gonzáles. *Manual de Hibernate*. Héctor Suárez Gonzáles, 2003.
- [11] Patrick Peak y Nick Heudecker. *Hibernate Quickly*. Manning Publications, 2006.
- [12] Jonathan Knudsen y Pat Niemeyer. *Learning Java, 2nd Edition*. O'Reilly Media, 2002.
- [13] Sharon Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser y Mark Hoeber. *The Java™ Tutorial Fourth Edition: A Short Course on the Basics*. Addison Wesley Professional, 2006.
- [14] Jorge Sánchez. *Java2 incluye Swing, Threads, programación en red, JavaBeans, JDBC y JSP / Servlets*. Jorge Sánchez, 2004.

- [15] Javier García de Jalón, José Ignacio Rodríguez, Iñigo Mingo, Aitor Imaz, Alfonso Brazález, Alberto Larzabal, Jesús Callejas y Jon García. *Aprenda Java como si estuviera en primero*. TECNUM, 2000.
- [16] Erik Hatcher y Steve Loughran. *Java Development with Ant*. Manning Publications, 2003.
- [17] Jesse Tilly y Eric Burke. *Ant: the Definitive Guide, First Edition*. O'Reilly Media, 2002.
- [18] Peter den Haan, Lance Lavandowska, Sathya Narayana Panduranga y Krishnaraj Perrumal. *Beginning JSP 2: From Novice to Professional*. Apress, 2004.
- [19] By H. M. Deitel - Deitel & Associates, Inc., P. J. Deitel - Deitel & Associates, Inc. *Java™ How to Program, Sixth Edition*. Prentice Hall, 2004.
- [20] By Ken Arnold, James Gosling y David Holmes. *THE Java™ Programming Language, Fourth Edition*. Addison Wesley Professional. 2005.
- [21] Brian Keeton, Chuck Cavaness y Geoff Friesen. *Special Edition Using Java 2 Standard Edition*. Que, 2000.
- [22] James Turner. *MySQL™ and JSP™ Web Applications: Data-Driven Programming Using Tomcat and MySQL*. Sams Publishing, 2002.
- [23] Giulio Sambo y Michael Sekler. *Beginning JSP™, JSF™ and Tomcat, Web Development From Novice to Professional*. Apress, 2007.
- [24] Ángel Esteban. *Tecnologías de Servidor con Java: Servlets, JavaBeans, JSP*. Grupo Eidos, 2000.
- [25] Hans Bersten. *JavaServer Pages*. O'Reilly Media, 2000.
- [26] Ian Gilfillan. *La Biblia MySQL*. Anaya Multimedia.
- [27] MySQL AB. *MySQL Query Browser*. MySQL AB, 2008.
- [28] MySQL AB. *MySQL 5.0 Reference Manual*. MySQL AB, 2008.
- [29] Michael “Monty” Wideniues, David Axmark y MySQL AB. *MySQL Reference Manual Documentation from the Source*. O'Reilly Media, 2003.

- [30] Carlos Eduardo Aníbarra Zelaya. *Manual Basico de HTML: Creación y Estructuración de Paginas WEB*. Carlos Eduardo, 2001.
- [31] Chuck Musciano, Bill Kennedy. *HTML & XHTML: The Defenitive Guide Fourth Edition, Second Edition*. O'Reilly Media, 2000.
- [32] Jason Brittain, Ian F. Darwin. *Tomcat The Definitive Guide*. O'Reilly Media, 2007.
- [33] Alberto Escudero Pascual y Louise Berthilson. *VoIP para el desarrollo: Una guía para crear una infraestructura de voz en regiones en desarrollo*. It+46, 2006.
- [34] Theodore Wallingford. *Switching to VoIP*. O'Reilly Media, 2005.
- [35] Equipo de instalador de debian. *Guía de instalación de Debian GNU/Linux*. Debian GNU/Linux, 2007, <http://www.debian.org/releases/stable/i386/index.html.es>
- [36] Equipo de Instalador de debian. *Obteniendo Debian Etch*, http://cdimage.debian.org/debian-cd/4.0_r5/i386/bt-dvd/
- [37] *Pagina Oficial del Sistema Operativo Debian GNU/Linux* <http://www.debian.org/>
- [38] Terry Dawson y Joshua Drake. *Redes en Linux Como*. Debian GNU/Linux, 1997.
- [39] Gustavo Noronha Silva y Hugo Mora. *APT HOWTO*. Debian GNU/Linux ,2003. <http://www.debian.org/doc/manuals/apt-howto/index.es.html>
- [40] Historia de VoIP. <http://voipex.blogspot.com/2006/04/historia-de-voip.html>.
- [41] Codec utilizados para trasmitir sonido en comunicaciones por Voip: <http://www.digium.com/en/products/g729codec.php>
- [42] VoIP Codec Compare: <http://www.brujula.net/speedtest/>
- [43] Tutorial de Asterisk: <http://www.asteriskguru.com/>
- [44] Bill MaCcarty. *Learning Debian GNU/Linux, 1st Edition*. O'Reilly Media, 1999.
- [45] Zapata Telephony Proyect: <http://www.zapatatelephony.org/>

[46] Foro de Voip: <http://www.voip-info.org/wiki/view/Asterisk+codecs>

[47] Pagina Oficial de Tomcat: <http://tomcat.apache.org/>

[48] Pagina Oficial de Digium: <http://www.digium.com>