



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA, EN
TELECOMUNICACIONES**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA**

**TEMA: EVALUACIÓN DE UN PROTOTIPO DE CENTRAL ASTERISK
IMPLEMENTADO SOBRE LA TARJETA RASPBERRY PI**

AUTORES: CHÁVEZ ARIAS, MARÍA JOSÉ

SALTOS RODRIGUEZ, DIANA JAQUELINE

DIRECTOR: ING. ACOSTA, FREDDY

CODIRECTOR: ING. LARA, ROMÁN

SANGOLQUÍ, 2015

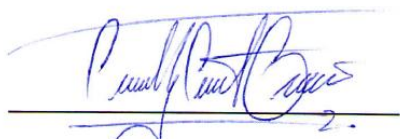
CERTIFICACIÓN

Certificamos que el proyecto titulado “EVALUACIÓN DE UN PROTOTIPO DE CENTRAL ASTERISK IMPLEMENTADO SOBRE LA TARJETA RASPBERRY PI”, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas.

Debido a que se trata de un trabajo de investigación se recomienda su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan ala Srta. Chávez Arias María José con C.I: 171760362-3 y ala Srta. Saltos Rodríguez Diana Jaqueline con C.I: 1723556765 que lo entreguen al Ing. Paul Bernal, en su calidad de Director de la Carrera.

Atentamente.



Ing. Freddy Acosta.

DIRECTOR



Ing. Román Lara

CODIRECTOR

DECLARACIÓN DE RESPONSABILIDAD

MARÍA JOSÉ CHÁVEZ ARIAS
DIANA JAQUELINE SALTOS RODRIGUEZ

DECLARAMOS QUE:

El proyecto de grado denominado “EVALUACIÓN DE UN PROTOTIPO DE CENTRAL ASTERISK IMPLEMENTADO SOBRE LA TARJETA RASPBERRY PI”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

En virtud a esta declaración, nos responsabilizamos del contenido, veracidad y alcance del proyecto de grado en mención.

Sangolquí, 17 Junio del 2015



Chávez Arias María José



Saltos Rodríguez Diana Jaqueline

AUTORIZACIÓN

MARÍA JOSÉ CHÁVEZ ARIAS
DIANA JAQUELINE SALTOS RODRIGUEZ

Autorizamos a la Universidad de las Fuerzas Armadas “ESPE” la publicación, en la biblioteca virtual de la institución el trabajo “EVALUACIÓN DE UN PROTOTIPO DE CENTRAL ASTERISK IMPLEMENTADO SOBRE LA TARJETA RASPBERRY PI”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 17 Junio del 2015



Chávez Arias María José



Saltos Rodríguez Diana Jaqueline

DEDICATORIAS

“La última derrota de la vida es ceder ante las dificultades”.

A Dios principalmente, a mis padres por su constancia y sacrificio, a mis hermanos quienes amo con el alma, a una persona muy especial Patricio Alejandro Guerra Vaca que siempre ha estado conmigo.

Saltos Rodríguez Diana Jaqueline

A Dios por haberme dado la fortaleza, sabiduría y bendiciones a lo largo de estos 5 años. A mis padres por su apoyo, comprensión y esfuerzo. A mi hermano por ser mi inspiración y a mi abuelita que desde el cielo me ha cuidado y quien fue parte de lo que soy ahora.

Chávez Arias Marías José

AGRADECIMIENTOS

A Dios principalmente, a mis padres, mis ángeles guardianes que siempre están conmigo dándome su apoyo y su amor, a mis hermanos a quienes añoro con el alma, son mis mejores amigos y mis confidentes, a una persona muy especial Patricio Alejandro Guerra Vaca, quien ha estado a mi lado durante toda mi carrera universitaria, apoyándome y animándome siempre, a mis abuelitos pilar fundamental de la familia de la que orgullosamente soy parte, a familiares y amigos, nunca me he sentido sola, siempre hay alguien tendiéndome su mano, a nuestros queridos ingenieros quienes han sido también amigos y nos han sabido a guiar en este camino. Gracias a todos por ser mi inspiración.

Saltos Rodríguez Diana Jaqueline

“Si tiene un gran sueño debes estar dispuesto a un gran esfuerzo para concretarlo porque solo lo grande alcanza a lo grande”. Facundo Cabral

A Dios y a la Virgen por haberme cuidado y protegido siempre, por colocarme pruebas de las cuales he sacado grandes enseñanzas, por haberme dado a mi motivación u motor, mis padres, a quienes les agradezco por su apoyo, por enseñarme a no rendirme ante las dificultades, y lo que soy el día de hoy es gracias a cada uno de los valores que me han inculcado. A mi hermano por estar siempre pendiente de mí, por ser mi inspiración de esfuerzo y determinación. A mi tío Patricio que me ha brindado su apoyo y motivación a lo largo de estos años.

A mis amigos con quienes nos enfrentamos a malas noches, semanas enteras sin dormir, momentos de estrés, y frustración de los cuales surgimos y llegamos a la metas. Finalmente a cada una de las personas, Ingenieros que más que maestros fueron amigos, que nos apoyaron de una u otra maneras, mil gracias por sus palabras de aliento, Dios los bendiga.

Chávez Arias Marías José

INDICE DE CONTENIDO

CERTIFICACIÓN	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
AGRADECIMIENTOS	vi
RESUMEN.....	xii
ABSTRACT	xiii
CAPÍTULO 1 : INTRODUCCIÓN.....	15
1.1. Presentación	15
1.2. Objetivos	16
1.2.1. General	16
1.2.2. Específicos.....	16
1.3. Justificación e importancia del proyecto	17
1.4. Alcance del proyecto.....	18
CAPÍTULO 2 : MARCO TEÓRICO.....	19
2.1. Introducción de la Telefonía IP.....	19
2.2. Definición	20
2.2.1. Características de los sistemas de Telefonía IP	21
2.2.2. Componentes del sistema de Telefonía IP	23
2.2.3. Ventajas de la Telefonía IP.....	25
2.2.4. Desventajas de la Telefonía IP	26
2.2.5. Arquitectura de la Telefonía IP.....	26
2.2.6. Escenarios de redes de telefonía IP	29
2.3. Comparación de la Telefonía Tradicional y la Telefonía IP	30
2.4. Protocolos VoIP	31
2.4.1. Protocolo H.323.....	32
2.4.2. Protocolo SIP	33
2.4.3. Protocolo IAX2.....	34
2.4.4. Protocolo MGCP	35
2.5. CODECS.....	35
2.5.1. G.711.....	36
2.5.2. G.726.....	37
2.5.3. G.723.1	37

2.5.4. G.729a	37
2.5.5. GSM	38
2.5.6. SPEEX	38
2.5.7. iLBC	38
2.6. SOFTWARE LIBRE ASTERISK.....	39
2.6.1. Introducción.....	39
2.6.2. Definición y características	40
2.6.3. Asterisk como voice-mail	44
2.6.4. Asterisk como ACD	45
2.6.5. Asterisk como Gateway.....	45
2.6.6. AGI (Asterisk Gateway Interface)	46
2.6.7. Arquitectura.....	47
2.6.8. Descripción de los APIs	47
2.6.9. Canales.....	48
2.6.10. Configuración.....	49
2.7. HARDWARE RASPBERRY PI.....	51
2.7.1. Introducción de Raspberry PI.....	51
2.7.2. Definición	52
2.7.3. Características.....	52
2.7.4. Raspberry PI vs Arduino.....	53
2.7.4.1. Hardware y software.....	55
2.8. DISEÑO DE ASTERISK COMO INTERFAZ GRÁFICA.....	56
2.8.1. PYTHON.....	56
2.8.1.1. Definición de Python.....	56
2.8.1.2. Características de Python	56
2.8.2. PHP	58
2.8.2.1. Definición de PHP	58
2.8.2.2. Características de PHP.....	59
2.8.2.3. Funcionamiento PHP	60
2.8.3. PHP VS PYTHON	61
CAPÍTULO 3 : IMPLEMENTACIÓN CENTRAL TELEFÓNICA IP	63
3.1. Topología.....	63
3.2. Selección de Hardware	63

3.2.1. Servidor.....	63
3.2.2. Enrutadores.....	64
3.2.3. Gateways.....	66
3.3. Selección del Software Asterisk.....	67
3.3.1. Instalación	67
3.3.2. Configuración de Asterisk.....	68
3.4. Archivos de Configuración de Asterisk	69
3.4.1. Extensiones.....	70
3.4.1.1. SIP.CONF	70
3.4.1.2. IAX.CONF	73
3.4.1.3. Comparación entre IAX y SIP	74
3.4.2. Buzón de Voz.....	75
3.4.2.1. VOICEMAIL.CONF.....	76
3.4.3. FOLLOWME.CONF	81
3.4.4. FEATURES.CONF	83
3.4.4.1. Transferencia de Llamadas.....	83
3.4.4.2. Parqueo de Llamadas	83
3.4.5. Distribución Automática de Llamadas.....	86
3.4.5.1. Call Center	86
3.4.5.2. AGENTS.CONF	88
3.4.5.3. QUEUES.CONF.....	89
3.4.5.4. Música en espera.....	91
3.4.5.5. MUSICONHOLD.CONF.....	92
3.4.6. Comunicación de centrales Asterisk	92
3.4.7. Comunicación con la PSTN.....	94
3.4.8. CDR.....	95
3.4.9. EXTENSIONS.CONF	97
CAPÍTULO 4 : EVALUACIÓN DEL PROTOTIPO.....	107
4.1. Calidad de Servicio para VoIP (QoS).....	107
4.2. Problemas que se presentan con la calidad de voz	107
4.2.1. Jitter.....	108
4.2.2. Latencia	108
4.2.3. Pérdida de paquetes.....	109

4.2.4. Eco.....	109
4.2.5. Ancho de banda insuficiente	109
4.2.6. Umbrales aceptables de la VoIP	110
4.2.7. MOS (Mean Opinion Score)	111
4.3. Llamadas simultáneas	112
4.3.1. Llamadas simultáneas considerando el ancho de banda.....	112
4.3.2. Llamadas simultáneas considerando el rendimiento de la Raspberry PI	114
4.3.2.1. SIPP	116
4.4. Análisis de QoS	120
4.4.1. SIP-SIP	121
4.4.2. 10 Llamadas simultáneas	123
4.4.3. 35 Llamadas simultáneas	127
4.5. Análisis de Costos	131
4.5.1. Costos implementación central	132
4.5.2. Ahorro mensual	134
CAPÍTULO 5 : CONCLUSIONES Y RECOMENDACIONES	135
5.1. Conclusiones.....	135
5.2. Recomendaciones.....	137
REFERENCIAS	138

ÍNDICE TABLAS

Tabla 1.....	41
Tabla 2.....	55
Tabla 3.....	96
Tabla 4.....	111
Tabla 5.....	111
Tabla 6.....	112
Tabla 7.....	115
Tabla 8.....	118
Tabla 9.....	119
Tabla 10.....	126
Tabla 11.....	130
Tabla 12.....	131
Tabla 13.....	132
Tabla 14.....	132
Tabla 15.....	134

ÍNDICE DE FIGURAS

Figura 1. Paquete IP	20
Figura 2. Red de Telefonía IP	21
Figura 3. Componentes de la Telefonía IP	24
Figura 4. Modelo TCP/IP para VoIP.....	27
Figura 5. Asterisk como PBX.....	39
Figura 6. Que es Asterisk?	40
Figura 7. Asterisk de forma modular.....	42
Figura 8. FXS, FXO	43
Figura 9. Funcionalidades de Asterisk.....	44
Figura 10. Asterisk como Gateway.....	46
Figura 11. Arquitectura de Asterisk	49
Figura 12. Raspberry PI	52
Figura 13. Características Raspberry PI	53
Figura 14. Funcionamiento PHP	60
Figura 15. Cómo trabaja PHP	61
Figura 16. Router Mikrotik.....	65
Figura 17. Gateway de Voz Grandstream.....	66
Figura 18. Configuración del CDR.....	95
Figura 19. Codec de Audio	113
Figura 20. Llamadas simultáneas vs Rendimiento	118
Figura 21. Porcentaje de llamadas Pérdidas.....	120
Figura 22. Tráfico TCP, RTP con una llamadas de SIP-SIP	121
Figura 23. Tráfico RTP cursado por la Central con una llamadas de SIP-SIP.....	122
Figura 24. Llamada de SIP-SIP	122
Figura 25. Trama de la llamada de SIP-SIP	123
Figura 26. Tráfico TCP, RTP con 10 llamadas simultáneas	123
Figura 27. Tráfico RTP con 10 llamadas simultáneas.....	124
Figura 28. 10 Llamadas realizadas	124
Figura 29. Trama de una llamada con 0% de paquetes perdidos.....	125
Figura 30. Trama de una llamada con más del 1% de paquetes perdidos.....	126
Figura 31. Tráfico TCP, RTP con 35 llamadas simultáneas	127
Figura 32. Tráfico RTP con 35 llamadas simultáneas.....	128
Figura 33. 21 Llamadas realizadas	129
Figura 34. Trama de una llamada con menos del 1,5% de paquetes perdidos.....	129
Figura 35. Trama de una llamada con más del 1,5% de paquetes perdidos	130
Figura 36. Escenario implementado.....	133

RESUMEN

Este trabajo busca diseñar, implementar y evaluar un prototipo de central Asterisk sobre 4 tarjetas Raspberry Pi para los laboratorios de Telefonía IP del Departamento de Eléctrica y Electrónica de la Universidad de Fuerzas Armadas ESPE. Se cuenta con un estudio del estado del arte de los temas relacionados con la arquitectura de Telefonía IP, protocolos de VoIP, software (Asterisk), especificaciones, funcionamiento y aplicaciones realizadas con Raspberry Pi. Para un manejo sencillo de la central sobre la Raspberry se diseñó e implementó una interfaz amigable al usuario en Python, permitiendo a los estudiantes realizar las configuraciones avanzadas de la central telefónica IP Asterisk como: buzón de voz, llamada en espera, IVR (InteractiveVoice Response), ACD (Automatic Call Distributor), Webmail, además de sistemas CTI para la gestión de la atención al cliente, donde el cliente pueda interactuar de diferentes formas con la persona, empresa, etc donde esté realizando la llamada. Además cuenta con un registro detallado de las llamadas que se realizan en la central (CDR), las cuales son almacenadas en una base de datos SQL obteniendo una facturación de las llamadas, obteniendo así un Call Center. Para finalizar se adicionó un dispositivo de entradas FXS, FXO para poder salir a la red PSTN. Se detalla la arquitectura, programación y estudio de los resultados de la aplicación.

PALABRAS CLAVES:

- **ASTERISK**
- **RASPBERRY PI**
- **ACD**
- **IVR**
- **CDR**
- **FXS**
- **FXO**

ABSTRACT

This project seeks to design, implement and evaluate a prototype central Asterisk of 4 cards Raspberry Pi for IP Telephony laboratories of the Department of Electronics, University of Armed ESPE. It has a state of the art study of issues related to the architecture of IP Telephony, VoIP protocols, software (Asterisk), specifications, performance and applications made with Raspberry PI. For easy operation of the center on the Raspberry was designed and implemented a user-friendly interface in Python, allowing students to perform advanced settings Asterisk IP PBX as (Interactive Voice voicemail, call waiting, IVR (Interactive Voice Response), ACD (Automatic Call Distributor), Webmail, plus CTI systems for managing customer service, where customers can interact in different ways with the person, company, etc where you are calling. It also has a detailed log of calls made in the central (CDR), which are stored in a SQL database obtaining an invoice for calls, so obtaining a Call Center. As the final part FXS device entries, FXO was added to get out to the PSTN. Architecture, programming and studying the results of the implementation is detailed.

- **ASTERISK**
- **RASPBERRY PI**
- **ACD**
- **IVR**
- **CDR**
- **FXS**
- **FXO**

EVALUACIÓN DE UN PROTOTIPO DE CENTRAL ASTERISK IMPLEMENTADO SOBRE LA TARJETA RASPBERRY PI

El presente proyecto abarca el proceso de evaluación de un prototipo de central telefónica IP con software libre (Asterisk) sobre hardware de bajo costo (Raspberry Pi).

Inicialmente en el primer capítulo se desarrolla los contenidos del proyecto como son: la presentación, resumen, objetivos, justificación e importancia y alcance.

En el segundo capítulo se describe los fundamentos teóricos que se requieren para poder conocer y comprender sobre la telefonía IP, sus características generales, arquitectura, etc, además se describe el software libre Asterisk, encargado de poner en funcionamiento la central sobre la Raspberry Pi. Para finalizar se describe los beneficios de la herramienta Python, la cual permitirá el diseño de la interfaz gráfica para la configuración de las funciones básicas y avanzadas de la central.

El tercer capítulo está dedicado a la implementación de las diferentes funcionalidades de Asterisk sobre la tarjeta Raspberry PI, considerando el escenario donde se evaluará cada una de ellas.

En el cuarto capítulo se presenta la evaluación de diferentes parámetros como jitter, paquetes perdidos, ancho de banda, eco, etc además de saturación de la central para determinar el número de llamadas simultáneas.

El quinto capítulo se presenta las conclusiones y recomendaciones del proyecto, además se incluye la información, respaldada en formato digital, necesario para la ejecución del proyecto, del prototipo de central Asterisk sobre la tarjeta Raspberry PI.

CAPÍTULO 1 : INTRODUCCIÓN

1.1. Presentación

Las comunicaciones de voz son de suma importancia en cualquier ámbito ya sea laboral, social o estudiantil. Para ello se desarrolló la telefonía fija, la cual permitió acortar distancias entre ciudades, países, empresas, etc.

La telefonía fija en primera instancia fue analógica, basada principalmente en la conmutación de circuitos, es decir, una comunicación se mantiene mientras dura la conexión, esto era posible a través de una central que en principio fue manipulada de forma manual, pero con el desarrollo tecnológico paso a ser automática.

Con el desarrollo tecnológico la telefonía evolucionó, contando actualmente con la telefonía IP, la cual es digital y en los últimos años ha tomado auge aumentando su número de abonados, este tipo de telefonía ofrece el servicio de transmisión de voz sobre una red de datos(internet, intranet, ISP) utilizando el protocolo IP (VoIP) [1].

La telefonía IP como una de sus principales ventajas ofrece disminuir costos de infraestructura, así como el costo tarifario por llamada. Además optimiza las líneas de comunicación, se puede obtener movilidad y acceso a funcionalidades avanzadas, por ejemplo: buzón de voz, IVR (Interactive Voice Response), ACD (Automatic Call Distributor), CTI (Computer Telephony Integration)[2].

La Universidad de las Fuerzas Armadas ESPE actualmente cuenta con una central 3com ASTERISK 3CR10551A la cual no abastece las necesidades de los estudiantes, esta tarjeta ofrecía flexibilidad y facilidad en las comunicaciones, sin embargo con el paso de los años se ha ido deteriorando y ha cumplido su tiempo de vida útil, adquirir un mayor número

de tarjetas se vuelve complicado debido al costo elevado de las mismas, por lo cual se contribuye a los estudiantes desarrollando una central con software libre y hardware de bajo costo que ofrezca las características necesarias de una central Asterisk.

1.2. Objetivos

1.2.1. General

Evaluar el desempeño de centrales telefónicas mediante la implementación de un prototipo basado en Asterisk sobre un controlador Raspberry PI.

1.2.2. Específicos

- Realizar, un detallado estudio del estado del arte sobre la telefonía VoIP que permita alcanzar un conocimiento profundo en cuanto a sus características generales, arquitectura, ventajas y desventajas.
- Diseñar, una interfaz gráfica de Asterisk amigable al usuario.
- Implementar, el prototipo de central telefónica, que servirá como material de apoyo para las prácticas de laboratorio de telefonía IP.
- Gestionar, de forma centralizada los dispositivos que formen parte de la telefonía VoIP.
- Evaluar, la funcionalidad y flexibilidad de la central telefónica mediante la utilización de la interfaz de usuario creada.
- Analizar, las capacidades de la tarjeta Raspberry PI como central telefónica en escenarios de inundación de central.

1.3. Justificación e importancia del proyecto

El evidente crecimiento de la telefónica IP hace necesaria una formación académica universitaria que incluya un conocimiento completo y útil para los ingenieros en Telecomunicaciones en cuanto a esta tecnología.

Los laboratorios de Electrónica de la Universidad de Fuerzas Armadas ESPE no cuenta con centrales que permitan realizar prácticas a los estudiantes utilizando la tecnología IP, por lo cual se debe acudir a otras herramientas para poder implementarlas, en ciertos casos son limitadas por incompatibilidad del software, además de consumir tiempo en instalación y ejecución del mismo, este tiempo podría ser utilizado para reafirmar conocimientos realizando un mayor número de prácticas y configuraciones con la central, para proporcionar Telefonía IP.

Asterisk es un software libre con una licencia GPL(General Public License) que se ejecutan en una PC estándar(arquitectura x86,x86_64, ppc) bajo GNU/Linux, BSD o MacOSX [3], que permite gestionar comunicaciones VoIP, sin embargo no existe gran flexibilidad en el manejo y estudio de este software de gestión debido a su interfaz poco amigable al usuario, por lo que se hace necesaria su integración con otros sistemas que faciliten su manejo.

Raspberry es un mini computador de placa reducida y bajo costo desarrollado para la enseñanza de computación en escuelas, este hardware permite la integración de Asterisk en su sistema operativo, de manera que los estudiantes universitarios que necesiten utilizar los grandes beneficios de este software puedan obviar la instalación del mismo, economizando así gran cantidad de tiempo.

Con la integración de Asterisk y Raspberry se busca implementar un prototipo que facilite la creación y configuración de centrales telefónicas IP mediante una interfaz gráfica amigable al usuario que permita colocar a cada dispositivo las funciones que deseen integrar en los mismos.

Una central IP debe ofrecer confiabilidad para cumplir con los objetivos que se desean lograr en la formación académica de los estudiantes universitarios, por tal motivo se hace necesaria la evaluación de los servicios que ofrece cada central telefónica así como también la verificación de los parámetros de calidad que ofrece la telefonía VoIP en el prototipo implementado.

1.4. Alcance del proyecto

El alcance del Proyecto incluye:

Estudio del estado del arte de la arquitectura de Telefonía IP, protocolos de VoIP, software (Asterisk), especificaciones, funcionamiento y aplicaciones realizadas con Raspberry Pi. Lo cual será el punto de partida para la instalación de Asterisk en 4 Raspberry para que funcionen como centrales de telefonía IP con las configuraciones de las diferentes aplicaciones avanzadas que provee este software.

Las configuraciones se realizarán mediante una interfaz que será diseñada e implementada en Python, entre las configuraciones que los estudiantes podrán configurar son opciones avanzadas como: buzón de voz, llamada en espera, IVR (Interactive Voice Response), ACD (Automatic Call Distributor), Webmail, además de sistemas CTI para la gestión de la atención al cliente, registro detallado de llamadas (CDR), para facturar llamadas, y la salida a la red PSTN con un dispositivo de entradas FXS, FXO. Finalizando con la evaluación del funcionamiento de la Central mediante pruebas y análisis de resultados de un central con saturación de llamadas.

CAPÍTULO 2 : MARCO TEÓRICO

2.1. Introducción de la Telefonía IP

En la década de los 80's, la VoIP se implementó para reducir el ancho de banda mediante compresión vocal aprovechando la compresión utilizada en los sistemas celulares, logrando reducir los costos en comunicación internacional, las primeras aplicaciones fueron implementados en RDSI (Red Digital de Servicios Integrados) sobre la red LAN, con el paso de los años se migro a redes WAN y finalmente las primeras aplicaciones generalizadas de VoIP aparecieron en los 90's con llamadas de voz gratuitas entre computadores con equipos especiales [4].

El avance de las Telecomunicaciones no se detiene, cada día el medio por el cual nos comunicamos a diferentes lugares, países, etc, está en constante evolución. El internet de las cosas, el fenómeno BYOD (Bring Your Own Device) orientado a mantenernos comunicados mediante cualquier dispositivo, cualquier contenido y de cualquier forma, nos lleva a buscar nuevas tecnologías enfocadas al intercambio de información por diferentes medios.

Por este motivo la telefonía evolucionó desde la conmutación de circuitos a lo que hoy es la telefonía IP. La conmutación de circuitos o la red PSTN aún sigue vigente siendo una central efectiva a lo que fue en sus inicios, pero al tener telefonía IP se cuenta con una red IP, por la cual se podrá enviar voz, video, datos llegando a ser una red más eficiente.

Cuando nos referimos a la telefonía IP se está hablando de una conmutación de paquetes y es una aplicación de la tecnología de VoIP (Voz sobre el Protocolo de Internet), los sistemas de VoIP son utilizados en redes de datos de área local LAN e Internet, se encuentran integrados con las redes tradicionales de telefonía. La transmisión de voz a través de redes basadas

en IP puede subdividirse en dos subconjuntos básico: transmisión conforme al protocolo VoIP y Telefonía Internet. La única diferencia es la naturaleza de la red IP subyacente: el VoIP se utiliza en redes basadas en IP privadas y gestionadas, mientras que la telefonía Internet utiliza básicamente la red Internet” [5].

2.2. Definición

Es una aplicación de la tecnología VoIP que permite que las señales de voz viajen a través de la redes IP(LAN, WAN) bajo el protocolo TCP/IP.

La telefonía IP maneja la comunicación mediante la conmutación de paquetes, es decir toda la información que se va a transmitir a través de la red se divide en paquetes de datos, cada paquete tiene un encabezado que contiene el origen y el destino, un número de secuencia, un bloque de datos y un código de comprobación de errores. Los enrutadores y servidores dirigen esos paquetes a través de la red hasta llegar al destino, cuando llegan a juntarlos en su orden original [6].

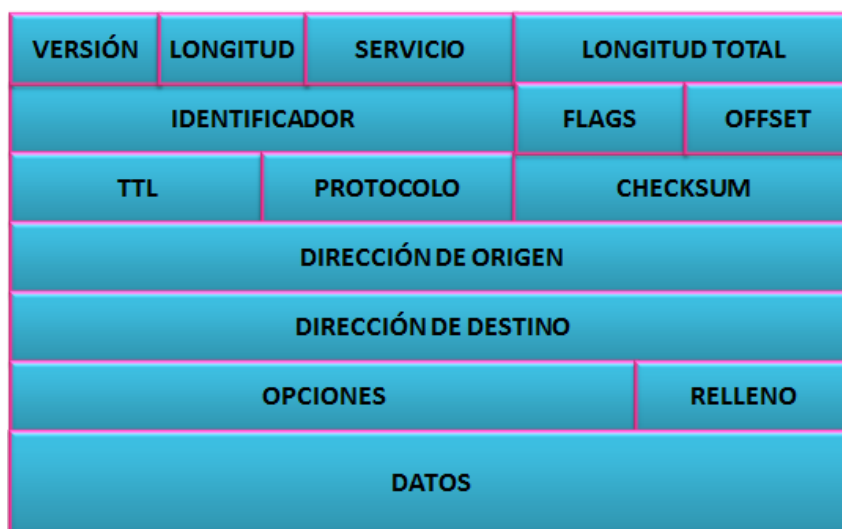


Figura 1. Paquete IP

Este tipo de telefonía se complementa con la tradicional, adicionándole ventajas económicas, satisfacción al cliente y ventajas tecnológicas.

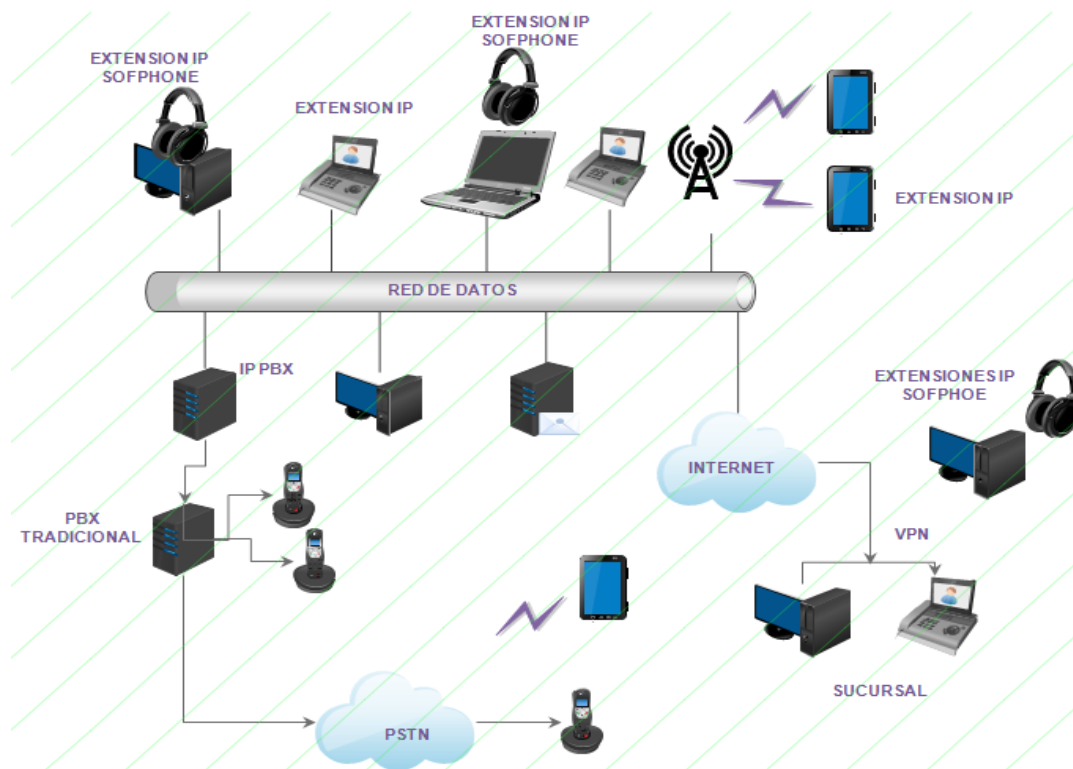


Figura 2. Red de Telefonía IP

2.2.1. Características de los sistemas de Telefonía IP

La telefonía IP permite integrar comunicaciones en una sola red económica y flexible, logrando que el desempeño de los miembros de una empresa sea eficiente y productivo.

Esta eficiencia se logra debido a que en una comunicación entre dos abonados mientras uno habla otro escucha, logrando que el archivo se reduzca a la mitad, de igual manera los espacios de silencio son eliminados, es decir la telefonía IP únicamente envía los paquetes que contenga información.

Se pueden utilizar teléfonos analógicos dentro de un sistema de telefonía IP, para ello es necesario utilizar tarjetas de interfaces analógicas, estas tarjetas convierten las señales de voz, de una señal analógica a una señal digital o viceversa. Estas tarjetas tienen puertos FXO (Foreign Exchange Office) permitiendo que un dispositivo se conecte a la Red PSTN, recibiendo y enviando la señal analógica, y FXS (Foreign Exchange Station) puerto que permite que un teléfono analógico se conecte a la central IP.

Permite un mayor número de funcionalidades con menor infraestructura que un Red Tradicional y logrando un ahorro económico. Algunas de las funciones que podemos realizar con la Telefonía IP son:

- Llamadas en espera
- Identificación de usuarios
- Buzón de voz
- Música en espera
- Videoconferencia
- Monitoreo de llamadas, etc.

Este sistema nos permite colocar llamadas con el Protocolo VoIP de las siguientes maneras:

ATA (Adaptador Telefónico analógico): permite conectar un teléfono normal a su computador o directamente a la conexión de internet para usar VoIP. El ATA es un convertidor analógico-digital que toma la señal análoga de su teléfono y la convierte en digital para transmitirla por internet [7].

Teléfonos IP: son teléfonos especiales, en lugar de usar conectores RJ-11 utilizan los RJ-45 para conectarse directamente en el Router, estos teléfonos tienen el hardware y software necesarios para manejar llamadas VoIP directamente en su tarjeta. Existen teléfonos Wi-Fi, los cuales se conectan y realizan llamadas VoIP a través de un punto inalámbrico [7].

Computador a Computador: se utiliza software denominado Softphone que se instalan en el computador, para lo cual es necesario una tarjeta de sonido, micrófono y conexión a internet ya sea por cable o un módem DSL, a excepción de la arquitectura que cuente con un ISP, las llamadas de computador a computador son gratis sin importar distancia.

2.2.2. Componentes del sistema de Telefonía IP

Si bien es cierto un sistema de Telefonía IP puede estar conformado por dos computadores con softphone instalados o dos teléfonos IP, es necesario considerar que se puede llegar a tener una escalabilidad, es decir que el lugar donde se esté dando este servicio vaya creciendo con un mayor número de abonados. Por ello para mantener en orden y en correcto funcionamiento la red se requiere de ciertos elementos que son fundamentales, como por ejemplo:

GateKeeper: es un protocolo que forma parte de VoIP, es el cerebro de la red de telefonía IP, controla el procesamiento de la llamada en protocolo H.323 y gestiona toda la comunicación de VoIP, además es el que permite la traducción de direcciones ip entre zonas. Ofrece servicios adicionales como: autorización de llamada, gestión de llamadas, gestión de ancho de banda y control de llamadas.

MGC (Media Gateway Controller) o Softswitch: es un software que contiene en su interior al GateKeeper y permite conectar a los abonados a través de la red de Telefonía Tradicional (PSTN) por internet, enviar y recibir paquetes de datos. Además permite conectar una gran variedad de servicios entre el emisor y receptor. Entonces el softswitch se construye con un servidor en cual este instalado un software, un ejemplo de software es Asterisk que es el motivo de estudio de este trabajo.

Gateway: es el dispositivo que permite conectar la red de telefonía IP hacia otros tipos de redes por ejemplo: redes H.323 con redes basadas en protocolo SIP, o la red de telefonía convencional (PSTN) o redes digitales de servicios integrados (ISDN). El Gateway es encargado de la conversión entre los diferentes formatos de transmisión y procedimientos de comunicación. Los terminales se comunican con los gateways utilizando los protocolos H.245 (Protocolo de control usado en el establecimiento y control de una llamada) y Q.931 (Protocolo que se define para la señalización de accesos RDSI básico) [8].

Terminales: son los dispositivos finales los cuales pueden ser hardware como los teléfonos IP, teléfonos analógicos con una conexión ATA y software con los denominados softphones, los softphones existen gratuitos y otros con licencias pagadas.

Estos dispositivos fundamentales se complementan con otros como por ejemplo switches y routers, permitiendo que todos los equipos se conecten a la red en el caso de los switches, mientras que el router permite la comunicación entre redes WAN.

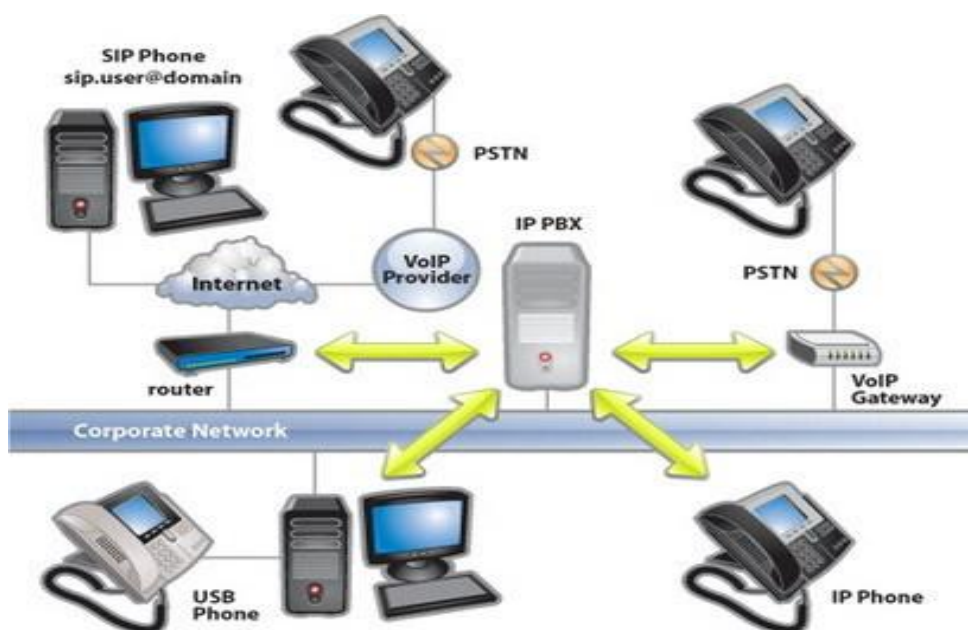


Figura 3. Componentes de la Telefonía IP

2.2.3. Ventajas de la Telefonía IP

- Mueve el tráfico multimedia por cualquier red que utilice IP. Esto ofrece a los usuarios flexibilidad en los medios físicos (líneas PSTN, Xdsl, ISDN), líneas dedicadas, cable coaxial, satélite y par trenzado y flexibilidad en la ubicación física [6].
- La conmutación de paquetes permite que varias llamadas telefónicas cursen simultáneamente por el mismo espacio que ocuparía una sola llamada con circuitos conmutados.
- Permite la reutilización y escalabilidad de las redes ya existentes, logrando instalar nuevos servicios con mayor rapidez.
- Administración inteligente ya que el usuario puede decidir qué hacer con la llamada entre varias funciones como por ejemplo: redirigir la llamada a otro número, tono de ocupado, etc.
- Los costos económicos se reducen a corto y largo plazo debido a factores como el hecho de que se puede utilizar la misma red para transmitir voz, video y datos.
- Movilidad a los usuarios ya que mediante un AP pueden conectarse a la red con su teléfono y recibir o realizar la llamada desde su teléfono celular sino se encuentra en su lugar de trabajo.
- Las llamadas entre sedes son gratuitas.
- Interoperabilidad con las redes telefónicas actuales [9].
- Calidad de Servicio Garantizada a través de una red de alta velocidad. Con un relativo ancho de banda (con 512Kbps ya se garantiza una buena calidad de voz) [9].

- Servicios de Valor Agregado(videoconferencia) [9]..

2.2.4. Desventajas de la Telefonía IP

- La voz puede escuchar distorsionada si no se cuenta con una red con calidad de servicio, debido a que se puede tener latencia o pérdida de paquetes. Este problema es más frecuente en enlaces WAN pero también está presente en enlaces LAN.
- La calidad de la transmisión se puede ver afectada en el caso que se cuente con softphones ya que al estar instalados en un computador dependerá del rendimiento de la misma.
- Este sujeta a potenciales vulnerabilidades como Spam, virus y Gusanos, pérdida de privacidad, Hackers.
- Depende de la energía eléctrica, sin estos los dispositivos dejan de funcionar a comparación de la telefonía tradicional donde aún existen teléfonos con los cuales sin electricidad se pueden recibir las llamadas.
- Al utilizar número telefónico con dirección IP se complica la comunicación para llamadas de Emergencia ya que no se puede asociar la zona geográfica, por ello si la persona que llama no le puede decir donde se encuentra, el operador del 911 no podrá saber dónde se encuentra [7].

2.2.5. Arquitectura de la Telefonía IP

La tecnología bajo la cual se maneja la Telefonía IP es la VoIP como se ha mencionado en temas anteriores, en este caso se va analizar como viajan los paquetes de VoIP por la red utilizando el modelo TCP/IP.

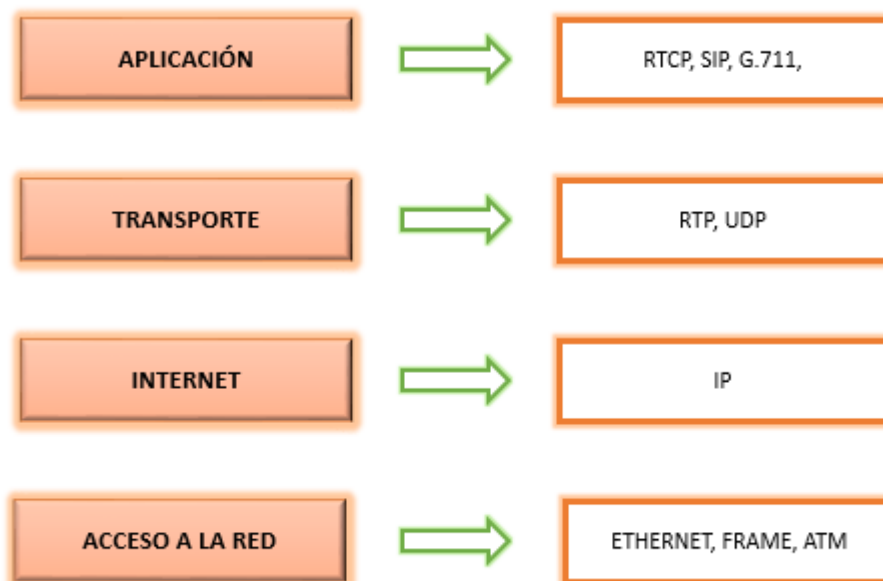


Figura 4. Modelo TCP/IP para VoIP

Capa Acceso a la Red: es la encargada de transmitir bits de información a través de las redes internas o externas convirtiendo los paquetes a señales eléctricas u ópticas, se encarga de las conexiones físicas entre los equipos hacia la red, de las velocidades de transmisión, etc. Además se ocupa del direccionamiento físico, de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control del flujo.

Red: es la capa encargada de que los paquetes lleguen desde su origen a su destino, en esta capa los paquetes son los denominados PDU, además de que utiliza el protocolo IP que es utilizado tanto en el origen como en el destino para comunicarse a través de Internet. Es decir es en donde se añade la dirección IP al paquete, en nuestro caso cada teléfono o PC tendrá una dirección IP.

Transporte: es la capa donde se verifica que los datos lleguen correctamente a su destino, recibe datos de las capas superiores, los divide y los pasa a la capa de red. Los protocolos de transporte son TCP y UDP.

UDP (User Datagram Protocol): es el protocolo de transporte utilizado en la Voip, en la mayoría de aplicaciones se utiliza el protocolo de transporte TCP ya que utiliza más tiempo de entrega de paquetes en el destino para asegurar que lleguen correctamente, pero en nuestro caso se selecciona el protocolo UDP ya que al hablar de aplicaciones en tiempo real lo que nos interesa es la velocidad de entrega de los paquetes mas no lleguen todos los paquetes.

Aplicación: permite la posibilidad de acceder a los servicios de las demás capas, además de definir los protocolos que utilizan las aplicaciones para intercambiar datos como por ejemplos: POP, SMTP, FTP. Además en esta capa también se cuenta con protocolos importantes utilizados para VoIP que son: RTP (Real Time Transport Protocol) Protocolo de transporte en tiempo real, RTCP (Real Time Control Transport Protocol) Protocolo de control de transporte en tiempo real y NTP.

RTP(Protocolo de transporte en tiempo real): es el protocolo encargado de transmitir en tiempo real tanto la voz como el video, funciona sobre el protocolo UDP, entre las funciones que podemos mencionar son:

- Identificar el tipo de información transmitida [10].
- Agregarle marcadores temporales y números de secuencia a la información transmitida [10].
- La medición de tiempo y el reporte de la calidad [10].

RTCP(Protocolo de control de transporte en tiempo real): monitoriza la entrega de la señal de voz y proporciona funciones mínimas de control para asegurar la entrega de paquetes [10].

2.2.6. Escenarios de redes de telefonía IP

Una llamada sobre la tecnología VoIP puede ocurrir en diferentes escenarios, dependiendo del tipo de red, el software y hardware a utilizar, es decir la infraestructura sobre la cual se monta este tipo de tecnología, entonces:

La VoIP funciona sobre redes IP. Existen tres tipos de redes IP:

- **Internet:** para el tráfico de la voz, la red de telefonía pública solo brinda una llamada por línea, mientras que los proveedores de telefonía por Internet pueden tener llamadas simultáneas restringidas solamente por el ancho de banda de Internet, por lo tanto la limitante es la falta de un ancho mínimo de banda estable, lo que se podría solucionar con paquetes muy pequeños.
- **Red IP públicas:** ciertos operadores de VoIP ofrecen a las empresas la conectividad necesaria para interconectar sus LAN y asegurar un ancho de banda constante, menos retardo, mayor seguridad, etc.
- **Intranet:** es la red de la propia empresa; puede tratarse de varias LAN conectadas por WAN. Ésta topología es idónea para VoIP.

Dentro del Hardware y Software de una infraestructura de la telefonía IP se puede encontrar con sistemas implementados únicamente en computadores o teléfonos celulares con software denominados softphones, sistemas en los cuales se utilizan los teléfonos IP, teléfonos IP y softphones y la adaptación de la infraestructura de la telefonía tradicional con la utilización de los denominados ATA para convertirlos en una infraestructura de telefonía IP.

2.3. Comparación de la Telefonía Tradicional y la Telefonía IP

La PSTN está construida sobre una infraestructura en la cual sólo los fabricantes de los equipos desarrollan aplicaciones para esos equipos, la Telefonía IP tiene una infraestructura más abierta donde varios fabricantes pueden proveer de un mayor número de aplicaciones logrando soluciones más creativas para los usuarios.

La PSTN no permite converger datos, voz y video con la infraestructura que se encuentra al momento debido a que no es lo suficientemente flexible, lo cual se convierte en un problema debido a que los datos actualmente se han convertido en un punto de suma importancia a nivel empresarial y residencial con el desarrollo del Internet orientados a banda ancha.

La telefonía fija tradicional utiliza conmutación de circuitos, ya que la central telefónica establece una conexión permanente entre dos usuarios, conexión que se utiliza para llevar las señales de voz, mientras que la telefonía IP utiliza conmutación de paquetes ya que los paquetes de datos, que contienen la señal de voz digitalizada y comprimida se envían a través de Internet a la dirección IP del destinatario. Cada paquete puede utilizar un camino diferente para llegar [11].

Los CoDecs de voz en la telefonía digital por lo general utilizan un menor ancho de banda, consumiendo tan solo 8kbps con lo que se logra producir una calidad de voz aceptable comparada con el ancho de banda necesario para la red de telefonía tradicional en el cual se utiliza un ancho de banda alrededor de los 64kbps.

Con el inconveniente que nos podemos encontrar en la telefonía IP es cuando debe circular por la red de Internet ya que puede tener problemas de confiabilidad y calidad del sonido debido a las limitaciones del ancho de banda como se ha mencionado anteriormente.

También podemos encontrar problemas de interoperabilidad entre protocolos de codificación y compresión de la voz [12]

Otros factores importantes a considerar dentro de la VoIP que se deben controlar son las demoras, pérdida de paquetes, ancho de banda y eco, temas que serán analizados con mayor profundidad en próximos capítulos.

La VoIP depende de energía eléctrica, los teléfonos actuales trabaja con alimentación fantasma proporcionada sobre la línea de la oficina central, incluso si hay fallo con el servicio eléctrico, el teléfono aún funciona [7]. Con VoIP, sin electricidad no hay teléfono por lo cual necesita una fuente de energía estable.

2.4. Protocolos VoIP

Los protocolos de VoIP son de suma importancia ya que de ellos depende la manera como se comunicaran los equipos, codecs con la red, para la eficacia y complejidad de la comunicación. Por lo tanto podemos decir que son el conjunto de reglas mediante el cual un terminal IP podrá establecer una llamada entrante, saliente, o si la línea está ocupada entre otros.

Los protocolos de señalización pueden ser propietarios o libres a continuación vamos a nombrar los principales en orden de antigüedad:

- H.323 – Protocolo definido por la ITU-T
- SIP – Protocolo definido por la IETF(hacia las redes Privadas).
- IAX – Protocolo originario de las PBXsAsterisk (actualmente se encuentra en su versión 2 –IAX2).
- H.248 (Megaco) – Protocolo de Control
- MGCP para el control de la gateway a la red pública PSTN (propietario Cisco)
- SkinnyClient Control Protocol – Protocolo propiedad de Cisco [13].

- MiNet - Protocolo propiedad de Mitel [13]
- CorNet-IP - Protocolo propiedad de Siemens [13].
- Skype – Protocolo Propietario peer-to-peer utilizado en la aplicación Skype [13].
- Jingle –Protocolo abierto utilizado en tecnología Jabber [13].

Además cabe mencionar que la señalización SS7 se utiliza hacia la red pública PSTN y el protocolo Q.931 se utiliza para el establecimiento de la llamada en H.323. A continuación vamos a describir los protocolos más utilizados actualmente y que son de uso libre.

2.4.1. Protocolo H.323

Es un protocolo creado por la International Telecommunication Union (ITU) fue el primero en desarrollarse con la idea de posibilitar la comunicación multimedia sobre redes IP principalmente para lo que es video conferencia. Es un protocolo robusto con una gran capacidad de interoperabilidad y escalabilidad para añadir nuevas funciones.

H.323 incluye un gran número de protocolos individuales desarrollados para aplicaciones específicas el problema es que no fue desarrollado específicamente para VoIP.

Es un protocolo para aplicaciones multimedia que hace referencia a los terminales, equipos y servicios estableciendo un señalización en redes IP, ofrece control y señalización de las llamadas, además es muy rápida comparada con SIP debido a que los mensajes de H.323 son binarios comparados con los mensajes que envía SIP [11].

Inicialmente fue creado para redes LAN pero con el tiempo se expandió a redes WAN, además permite interactuar con redes tradicionales.

2.4.2. Protocolo SIP

Session Initiation Protocol (Sip) es un protocolo más lineal y flexible desarrollado por Internet Engineering Task Force (IETF) específicamente para aplicaciones VoIP, con ciertas dificultades de interoperabilidad pero nacido desde, por y para Internet [14].

Funciona a nivel de la capa de aplicación utilizando el puerto 5060 para señalización con UDP o TCP y 2 puertos RTP para cada conexión de audio. Permite la creación, modificación y término de sesiones multimedia en donde interviene voz, video y mensajería instantánea entre uno o más dispositivos, o usuarios participantes.

La gran mayoría de los teléfonos IP así como los softphones soportan este tipo de protocolo, además de que posee la capacidad de saber en todo momento en donde se encuentra y el estado del usuario.

Entre los inconvenientes que puede presentar este tipo de protocolo es debido a que tiene problemas con el NAT (Network Address Translation), los datos y señalización viajan de forma separada y suele necesitar un servidor STUN (Session Traversal Utilities for NAT) para resolver este problema [15]. Otro problema que se puede presentar se da debido a que los protocolos utilizados alrededor del mundo no siempre son compatibles.

SIP está basado en cliente-servidor es decir tiene un mensaje de Request y Response. Los clientes envían peticiones a un servidor, el cual recibe una respuesta estableciendo la comunicación sin elementos intermediarios entre las terminales SIP [11].

2.4.3. Protocolo IAX2

IAX (Inter-Asterisk exchange) es un protocolo creado por Mark Spencer diseñador de Asterisk y en principio fue desarrollado para la comunicación entre otros servidores Asterisk, cabe mencionar que está en absoluto limita a Asterisk ya que permite la comunicación con cualquier otro tipo de servidores que utilicen este protocolo.

Además este protocolo apareció para solucionar ciertos inconvenientes presentados en el protocolo SIP como por ejemplo los procesos de NAT y firewalls, utiliza un solo puerto UDP que el 4569 tanto para canales de señalización como de datos de todas las llamadas.

Es un estándar abierto que puede ser modificado a los requerimientos del usuario, permite manejar una gran variedad de codecs y streams logrando así poder transportar cualquier tipo de dato.

Otra de las ventajas que nos ofrece este protocolo sobre SIP es que utiliza un menor ancho de banda debido a que los mensajes son codificados de forma binaria a comparación de los mensajes SIP que como ya se ha mencionado anteriormente son de texto.

Con el problema que nos podemos encontrar es que no se encuentra estandarizado y no es un protocolo que se va a encontrar en los dispositivos hardware.

IAX permite un enlace troncal denominado IAX Trunked y lo que permite es que el protocolo utilice una única cabecera para el paso de varias llamadas, es decir lo que se logra es poder pasar varias llamadas por el mismo ancho de banda por el cual pasaría una llamada con el protocolo SIP logrando así un ahorro importante del el ancho de banda. Lo que evitamos con este tipo de enlace troncal es el overhead de las capas IP y transporte UDP excesivo. El

inconveniente en este caso se presenta debido a que solo puede ser habilitado entre servidores Asterisk.

Actualmente IAX se encuentra en su segunda versión denominado IAX2 y fue creado y estandarizado en el 2004 por la empresa Digium.

2.4.4. Protocolo MGCP

Es un protocolo de la capa de aplicación desarrollado por el grupo IETF, MGCP (Media Gateway Control Protocol) se basa en texto para el establecimiento y control de las llamadas. Se maneja bajo una arquitectura maestro-esclavo, es decir permite la comunicación entre el controlador de Gateway MGC (Call Agent) con las Gateway de telefonía GW(hacia la PABX o PSTN), otro elemento importante es el SG (Signaling Gateway).

La función de Gateway tradicional es la de ofrecer conectividad y la traducción para que dos redes diferentes se puedan entender, en este caso el Gateway cumple la función de conversión del flujo de datos, conversión de señalización bidireccionalmente.

Por tanto el MG está encargado de la conversión del contenido multimedia, el MGC realiza el control de señalización del lado IP y el SG el control de la señalización del lado de la conmutación de circuitos.

La sesión de MGCP puede ser punto a punto o punto multipunto, además soporta un control de señalización de llamada escalable integrando QoS en el Gateway.

2.5. CODECS

Un CODEC hace referencia a un codificar/decodificador es decir a una Compresión/Descompresión, y la función que cumple es la de codificar audio

o video en un extremo de la comunicación para luego decodificarla en el otro extremo con el fin de lograr un sonido o imagen útil.

El objetivo del códec es transformar la señal analógica a una señal digital para poder transportar la voz en paquetes IP por una red de datos, por lo cual dependiendo del tipo de códec que utilicemos ocupará más o menos ancho de banda, además de otras consideraciones para su elección como: la calidad del sonido, requisitos computacionales, entre otros.

En cuanto a las aplicaciones los equipos que interviene en la VoIP utilizan diferentes tipos de codecs es por ellos que necesitan negociar cuál de ellos intervendrá en la comunicación. Algunos de los CODECS más utilizados para VoIP tenemos: G.711, G.726, G.723.1, G729, GSM, SPEEX, ILBC.

2.5.1. G.711

Es un Codec estandarizado por UIT por lo cual se encuentra en cualquier dispositivo VoIP, es fundamental en la PSTN, el más utilizado en ambientes LAN y muy conveniente en ambientes de gran ancho de banda. Es denominado también u-law (ley u) que codifica cada 14 muestras en palabras de 8 bits y es utilizado en Norte América y Japón y a-law (ley a) que codifica cada 13 muestras en palabras de 8 bits en el resto del mundo.

Es el códec que usa mayor velocidad con respecto a los demás proporcionando un flujo de datos de 64Kbits/s, de mayor calidad de audio, es óptimo y el consumo es moderado. Este codec requiere un procesamiento muy lento, necesita un mínimo de 128 Kbps para una comunicación bidireccional y fue la base para desarrollar los demás códec que existen en la actualidad y no requiere licencia para su utilización.

2.5.2. G.726

Es un Codec estandarizado por UIT y es conocido como ADPCM (Adaptive Differential Pulse Code Modulation), permite operar con diferentes tasas de transferencia logrando conseguir un ancho de banda de 16Kbps, 24Kbps y 32Kbps con la ventaja de disminuir el ancho de banda sin incrementar el costo computacional. Esto debido a que en lugar de enviar el resultado de medida de cuantización, envía únicamente suficiente información para describir la diferencia entre la muestra actual y la anterior a ella.

Comúnmente utiliza un ancho de banda de 32Kbps para troncales internacionales en la red telefónica, además de ser el códec estándar para teléfonos inalámbricos y no se necesita licencia para ser utilizado.

2.5.3. G.723.1

Es un Codec requerido por el protocolo H.323, utilizado para una comunicación de baja velocidad de transferencia protegido por patentes y el cual requiere una licencia para la aplicación comercial. Opera con velocidades de transmisión de 5.3Kbps y 6.3 Kbps.

2.5.4. G.729a

Es un Codec desarrollado por varias empresas privadas por lo cual requería una licencia para su utilización, actualmente existe una versión libre. Requiere un ancho de banda de 8Kbps con una carga computacional elevada. Suministra tasas de 6.4Kbit/s y 11.8Kbit/s para una peor o mejor calidad respectivamente y utiliza el algoritmo CS-ACELP para la compresión de voz, soportado por muchos teléfonos y sistemas.

2.5.5. GSM

Conocido por ser utilizado en redes GSM, su nombre original es Regular Pulse Excitation-Long Term Prediction (RPE-LTP), es muy popular en sistemas de VoIP. Codifica a 13Kbps con una carga computacional media, esto debido a que utiliza muestras anteriores para predecir la actual. Es un códec que no necesita licencia para su utilización y es soportado por varias plataformas de hardware y software.

2.5.6. SPEEX

Es un Codec de velocidad variable VBR(Variable Bitrate), su función es regular dinámicamente la tasa de transmisión de bits en función de los cambios de las condiciones de la red, generalmente el ancho de banda oscila entre 2.15Kbps a 22.4Kbps. Existen dos versiones que son el de banda ancha y banda angosta, la utilización de cada una de ellas dependerá de las calidad de voz requerida en los terminales. Es un códec de distribución libre, únicamente es necesario registrarse en <https://www.speex.org>.

2.5.7. iLBC

Es un Codec de baja velocidad de transferencia de Internet desarrollado por Global IP Solutions, libre no necesita ser licenciado para su distribución, utiliza algoritmos complejos lo que exige altos niveles de capacidad computacionales. Utilizado en aplicaciones de VoIP para streaming de audio y mensajería. Muestrea a 18Khz y utiliza una licencia PLC y certifica a 15.2Kbps o 13Kbps.

2.6. SOFTWARE LIBRE ASTERISK

2.6.1. Introducción

Existen algunos software que brindan soluciones para VoIP y que son de distribución libre, sin embargo Asterisk es la mejor opción ya que puede trabajar con líneas RDSI (Red Digital de Servicios Integrados), VoIP, terminales analógicos, teléfonos digitales, softphones etc., que pueden trabajar de forma independiente o en conjunto.



Figura 5. Asterisk como PBX

Asterisk puede ofrecer los servicios de telefonía como: transferencia de llamadas, música en espera, llamada tripartita, buzón de voz, web mail, desvío de llamadas, followme, sistema de control de colas, operadora virtual, IVR, videoconferencia, grabación de llamadas etc., ya sea a pocas líneas o a varios clientes.

Permite conectar un determinado número de líneas para realizar llamadas entre sí o para conectarse a un proveedor VoIP. Soporta distintos protocolos

y es el único software que permite tener las funcionalidades de una PBX a un costo reducido, a diferencia de otros sistemas PBX propietarios que son sumamente costosos, por tal motivo Asterisk se ha convertido en una de las herramientas más utilizadas en el mercado del manejo y solución de las telecomunicaciones.

Este proyecto fue desarrollado por el ingeniero Mark Spencer, miembro fundador de la compañía Digium, principal desarrolladora de Asterisk. Originalmente desarrollado para el sistema operativo GNU/Linux, Asterisk actualmente también se distribuye en versiones para los sistemas operativos BSD, MacOSX, Solaris y Microsoft Windows, aunque la plataforma nativa (GNU/Linux) es la mejor soportada de todas [16].

2.6.2. Definición y características

Es un software libre diseñado para integrar y unificar todos los sistemas de comunicaciones hasta el momento existentes, permite controlar y gestionar comunicaciones de cualquier tipo, ya sean analógicas, digitales o VoIP



Figura 6. Que es Asterisk?

mediante todos los protocolos VoIP que implementa.

Según [17] Asterisk es una aplicación servidor que permite que terminales “clientes” se conecten a él, una vez conectados, los usuarios pueden transmitir

voz y video en tiempo real utilizando cualquiera de los protocolos y códecs soportados por Asterisk.

Asterisk es un PBX híbrido que integra tecnologías como Multiplexación por división de tiempo y telefonía IP con funcionalidad de unidades de respuesta automática y distribución automática de llamadas [17].

Tabla 1.

Protocolos y Códecs soportados por Asterisk.

PROTOCOLOS	CODECS AUDIO
IAX2	G.729
SIP	GSM
Skinny	ILBC/Speech
MGCP	G.722/G.723
H.323	G.711a/G.711u

Asterisk es realmente una gran herramienta útil para el estudio y desarrollo de la telefonía IP, tiene la gran ventaja de tener varios usos, puede ser desarrollado para uso doméstico, uso empresarial, operadores de telefonía IP en pequeñas o grandes empresas, como sistemas para call-centers etc.

Asterisk es capaz de convertir una computadora común en un servidor de comunicaciones, además es muy modular ya que nos permite ajustarlo a nuestras necesidades, sin embargo el buen funcionamiento de Asterisk depende en gran parte del rendimiento de la CPU que esté utilizando.

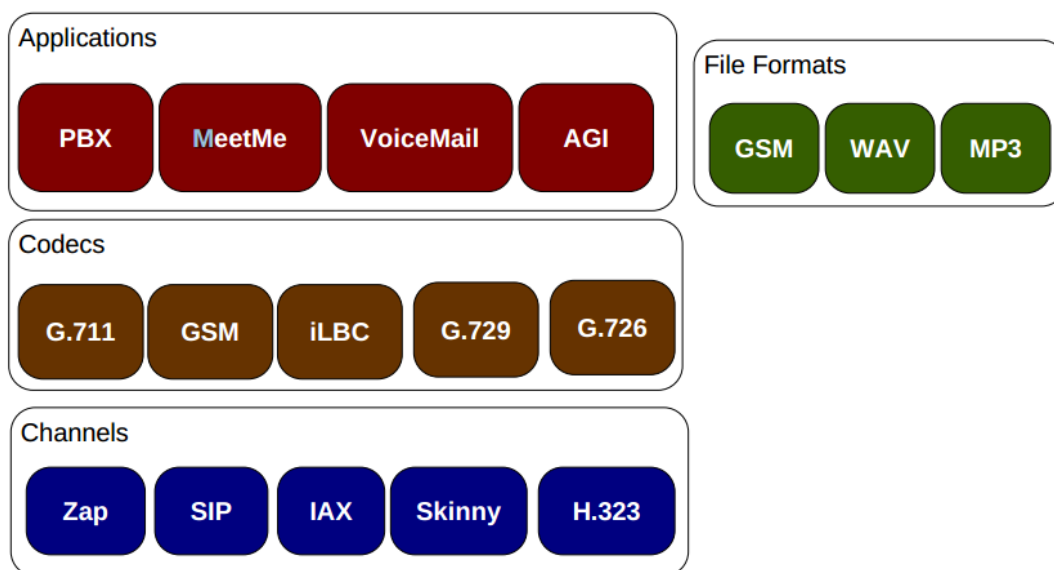


Figura 7. Asterisk de forma modular

Es recomendable preservar al máximo la CPU de Asterisk, se debe correr siempre en una maquina dedicada y probar el dimensionamiento antes de implantarlo. Asterisk siempre debe ser implementado en una VLAN (Red Virtual) específica para VoIP (Voz sobre IP). Cualquier tempestad de Broadcast causados por virus, puede comprometer su funcionamiento debido al uso de la CPU, cuando este fenómeno se da [17].

Su instalación es un poca larga, y su configuración un poco compleja, se necesita conocer la funcionalidad de cada módulo de Asterisk antes de empezar a configurarlo y programarlo.

Asterisk tienen una gran cantidad de funciones que se pueden crear a partir de éste software y que se utilizan de forma combinada con la finalidad de tener una mayor riqueza en las soluciones de telecomunicaciones.

Asterisk como PBX permite que todas las llamadas internas sean conmutadas directamente sin necesidad de salir por la red pública de telefonía es decir un sistema PBX evita que se tenga que conectar cada uno de los

dispositivos de una oficina de manera separada a la red de telefonía pública, con esto cuando se establece comunicación interna en una oficina no es necesario emplear la salida de llamadas hacia la central telefónica que regresan nuevamente para establecer comunicación interna, esto es una gran ventaja ya que se ahorra una gran cantidad de cargos mensuales, implementar un sistema PBX tiene un costo elevado pero con Asterisk el costo de implementación se reduce considerablemente.

Para funcionar con voz sobre IP no se necesita de ningún hardware adicional, ahora para interconectar con la telefonía tradicional se requiere de tarjetas especiales que se instalan en el computador y que son de muy bajo costo como las conocidas tarjetas FXO y FXS [18].

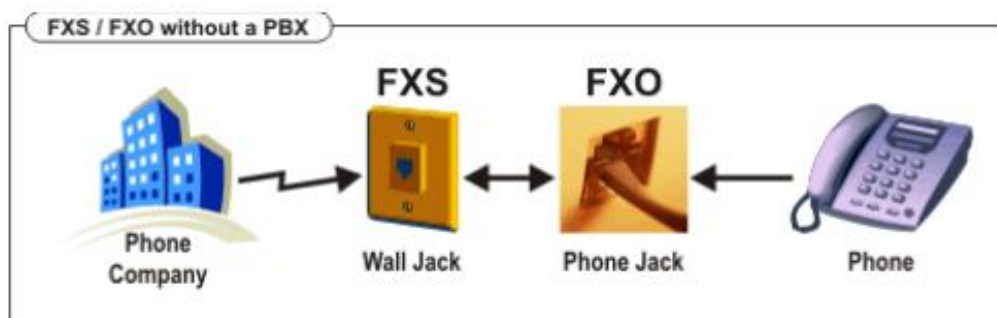


Figura 8. FXS, FXO [18]

2.1.1. Recursos comunes de Asterisk.

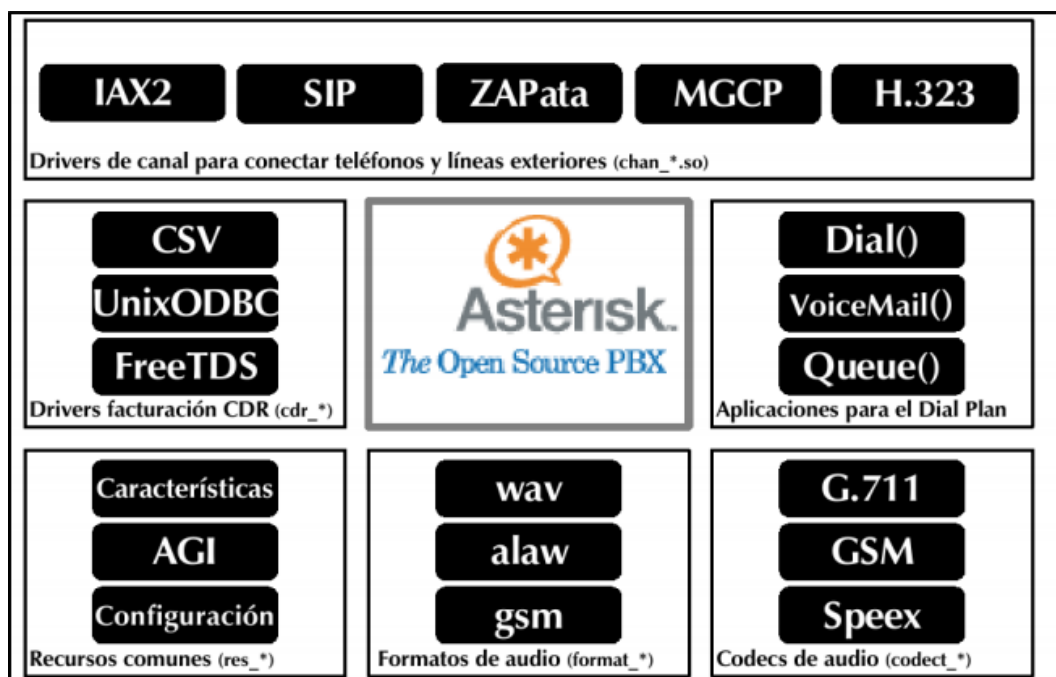


Figura 9. Funcionalidades de Asterisk

Entre los principales recursos que ofrece Asterisk como central telefónica están:

- Funcionalidades básicas sobre llamadas: Transferencia Ciega, Música en espera, Timbres Distintos, Pickup de llamadas, Llamada en espera, Conferencias, Buzón de Voz, Colas de llamadas, Colas con Prioridad, Caller ID en espera.
- Funcionalidades avanzadas como IVR, ACD, voicemail, conferencia.

2.6.3. Asterisk como voice-mail

La aplicación de correo de voz integrado de Asterisk permite sustituir los sistemas de correo de voz propietarios heredados por soluciones mucho más baratas, flexibles y personalizadas. El correo de voz de Asterisk puede ser implementado como soporte básico del sistema o puede actuar como cliente

a un sistema de mensajería, el almacenamiento de mensajes utilizando es IMAP8 u ODBC9 [16].

Asterisk IVR: Respuesta de Voz Interactiva (IVR, Interactive Voice Response) en un sistema telefónico capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Es un sistema automatizado orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones [16].

Asterisk voicemail utiliza los recursos de Asterisk IVR, cuando un usuario se comunica, un sistema de respuesta contesta la llamada con mensajes previamente grabados en archivos de audio con un menú de opciones entre las que el usuario puede elegir ya sea presionando una tecla o por reconocimiento de voz.

2.6.4. Asterisk como ACD

ACD (Automatic Call Distributor) permite a las empresas tele operadoras gestionar un sinnúmero de llamadas, por ejemplo en un sistema de callcenter de telefonía móvil se pueden encaminar las llamadas de acuerdo al plan contratado: pos pago, prepago etc., éstos sistemas son sumamente caros, sin embargo con Asterisk su costo de implementación únicamente se reduce al coste del servidor y de los teléfonos.

2.6.5. Asterisk como Gateway

Un gateway (puerta de enlace) es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su propósito es traducir la información del protocolo utilizado en una red, al protocolo usado en la red de destino [18].

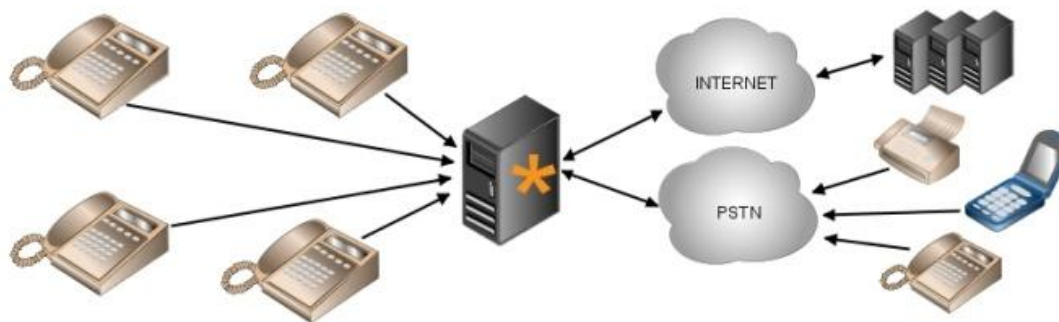


Figura 10. Asterisk como Gateway

Al tener Asterisk la funcionalidad de un Gateway puede convertir las señales analógicas (FX0, FXS) o digitales (IDSN) que vienen de la central telefónica o de VoIP y transmitir por la red corporativa de datos [17].

2.6.6. AGI (Asterisk Gateway Interface)

Muy parecido al uso de Aplicaciones, disponemos de esta pasarela para poder ejecutar servicios en los canales, pero basados en otros lenguajes de programación diferentes a la estructura de Asterisk en sí.

Existen librerías AGI prácticamente para todos los lenguajes más populares como pueden ser, PHP, Python, Java, etc.

Con esto tenemos un potencial ilimitado para crear o entrelazar aplicaciones ya existentes con nuestro sistema Asterisk, y aumentar con esto las funciones que posee Asterisk.

2.6.7. Arquitectura

Este software está formado por un núcleo principal encargado de gestionar todo el sistema PBX. Sus funcionales principales son: [16]

- Interconectar de forma automática cada llamada ente los usuarios participantes teniendo en cuenta el tipo de protocolo utilizado por cada terminal.
- Lanzar los servicios de valor añadido cuando sean requeridos.
- Traducir y adaptar los códecs a cada terminal involucrado en la comunicación.
Gestionar el sistema para que funcione de forma óptima en diversas condiciones de carga.

Para cumplir éstas funciones éste núcleo se apoya por 4 APIs, un API es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Usando este sistema basado en API, la base de Asterisk no tiene por qué preocuparse por detalles como ¿Que llamada se está realizando? O ¿Que Códecs se están utilizando? Entre otras [18].

2.6.8. Descripción de los APIs

- API de canales de Asterisk: Maneja el tipo de conexión por el cual el cliente está llegando, sea una conexión SIP, H.323, RDSI, etc.
- API de aplicaciones Asterisk: Permite a varios módulos de tareas cumplir varias funciones, multiconferencias, lista de directorios, buzones de voz, aplicaciones personalizadas, etc.

- API de traducción de códecs: Carga módulos Códices para apoyar varios tipos de audio, codificando y decodificando formatos tales como G.711, G.729, GSM, etc.
- API de formato de ficheros Asterisk: Maneja la lectura y escritura de varios formatos de archivos para el almacenaje de datos en el sistema de archivos.

Usando estos APIs Asterisk alcanza una completa abstracción entre sus funciones básicas y las diferentes tecnologías y aplicaciones relacionadas.

Ahora bien, las interconexiones existentes en la estructura Asterisk son denominadas canales, a continuación algunas definiciones utilizadas en esta estructura [18]:

2.6.9. Canales

Conexión que conduce una llamada entrante o saliente en el sistema Asterisk. La conexión puede venir o salir hacia telefonía tradicional analógica o digital o VoIP. Por defecto, Asterisk soporta una serie de canales, los más importantes son:

Protocolos: son utilizados para identificar conexiones y determinar puntos de destino y cuestiones relacionadas a la señalización de telefonía como el tono y el tiempo de campanilla.

Protocolos soportados por Asterisk:

- IAX2
- SIP
- SKINNY
- MGCP
- H.323

Códices y conversores (Coder/DECoder): Los Códices cumplen varias funciones entre las cuales se destacan la codificación de las llamadas para que ocupen menos espacio de banda.

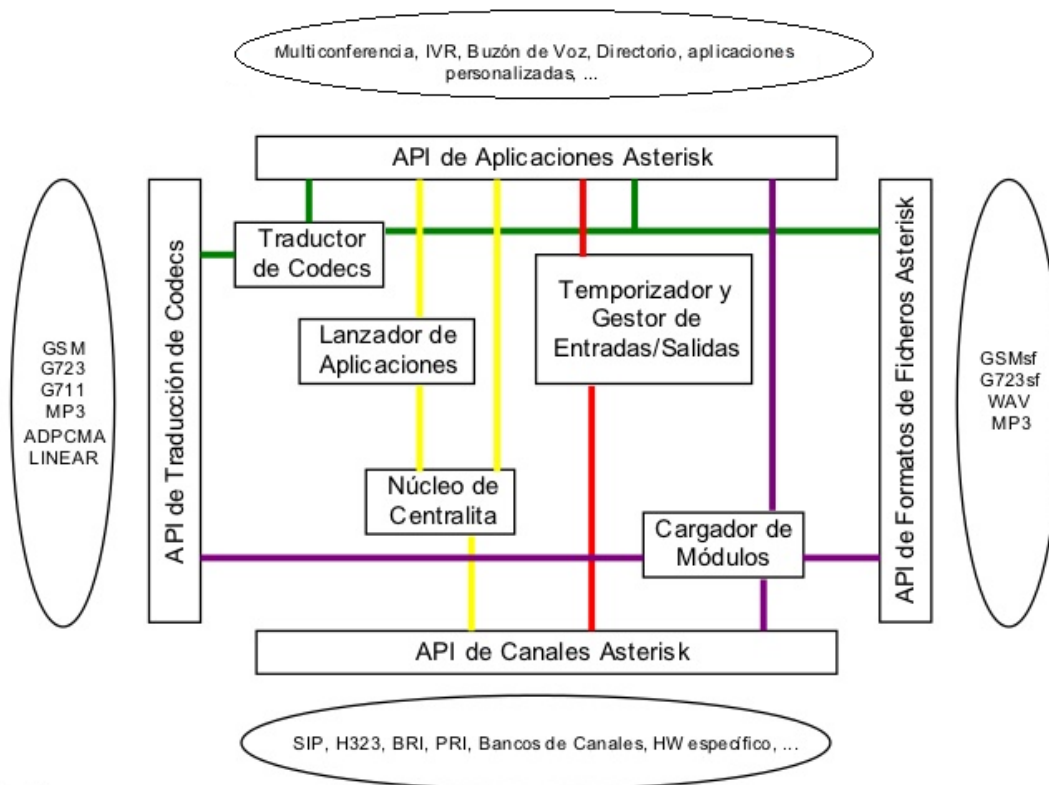


Figura 11. Arquitectura de Asterisk

2.6.10. Configuración

Asterisk tiene un conjunto de archivos que se pueden modificar de acuerdo a la funcionalidad que el usuario desee, éstos archivos se encuentran en distintos directorios:

- /etc/asterisk: directorio donde se almacenan los archivos de configuración que controlan Asterisk.

- `/usr/lib/asterisk/modules`: directorio donde se encuentran los módulos compilados en extensión `.so`.
- `/usr/share/asterisk`: directorio donde se guardan las claves RSA, los archivos para reproducir de música en espera, los script AGI en una instalación desde paquetera para distribuciones Debian y Ubuntu.
- `/var/lib/asterisk`: directorio donde se guardan las claves, la música en espera y los sonidos del sistema en una instalación desde las fuentes.
- `/var/spool/asterisk`: directorio donde se guardan las conferencias salvadas por `MeetMe()`, los textos que son leídos por `Dictate()` y los mensajes de voz si no utilizamos una BD.
- `/var/run/asterisk`: directorio donde se encuentra el archivo PID3 de Asterisk que se está ejecutando.
- `/var/log/asterisk`: en este directorio se almacenan los logs y notificaciones realizado mediante por ejemplo la aplicación `Verbose()` o `NoOp()` se encuentran aquí, aunque también son dirigidos al demonio `syslog`.

En el directorio `/etc/asterisk` encontraremos los archivos necesarios que se deben configurar para que Asterisk funcione como centralita.

Los archivos más importantes a configurar son:

- ***sip.conf***: Se configuran los usuarios Sip correspondientes a cada una de las extensiones.
- ***extensions.conf***: Se desarrolla el dialplan o plan de marcado que se encargará de gestionar todas las llamadas que se realicen, así

como de establecer los privilegios de los que dispondrán cada uno de los usuarios.

Éstos ficheros son los que se deben configurar para implementar un escenario VoIP sencillo con clientes SIP que se comuniquen entre sí, sin embargo para la implementación del presente trabajo se necesitará modificar otros ficheros, los cuales se profundizarán más adelante.

2.7. HARDWARE RASPBERRY PI

2.7.1. Introducción de Raspberry PI

Un ordenador portátil es realmente una herramienta hoy en día muy necesaria en la vida diaria, es una herramienta fundamental de trabajo, donde se desarrollan todo tipo de archivos, documentos, e inclusive aplicaciones especialmente para quienes se encuentran inmersos en el mundo de la programación, sin embargo si se necesita que un ordenador esté dedicado a la ejecución de una aplicación en particular resulta muy costoso destinar para éste fin una computadora portátil de las que hoy en día se conocen en el mercado (Tochiba, HP, Dell etc), además de consumir un gran espacio y desaprovechar gran cantidad de los recursos que ofrecen dichos ordenadores, es necesario una herramienta que ofrezca las características principales de un ordenador, de tamaño reducido, bajo coste y velocidad de procesamiento suficiente para el desarrollo de diversas aplicaciones, para solventar ésta necesidad existe hoy en día un hardware que cumple con las características antes mencionadas, la tarjeta RaspberryPI que además ofrece una velocidad de procesamiento sumamente conveniente para cualquier tipo de aplicación desarrollada en softwares como Python, PHP, Borland C entre otras.

2.7.2. Definición

Raspberry Pi es un ordenador de tamaño de una tarjeta de crédito, y de un coste reducido, tiene conexión de red y HDMI, puede comunicarse con componentes electrónicos, pensado para facilitar la enseñanza de la informática en general.

Raspberry Pi es ideal para personas que se interesan por la programación, sistemas embebidos o simplemente para aquellos que son curiosos. Además el puerto GPIO de la Raspi permite conectar periféricos extras o componentes electrónicos.

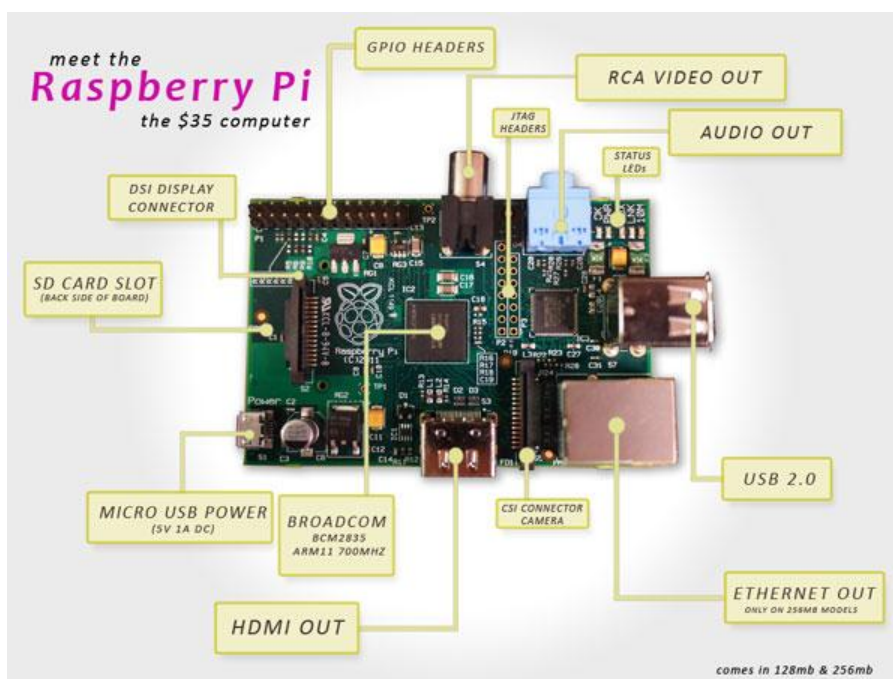


Figura 12. Raspberry PI

2.7.3. Características

- Necesita de una tarjeta SD dónde va instalado su sistema operativo (Raspbian) para iniciar el sistema.
- La memoria RAM es fija, no se puede cambiar.
- El tamaño mínimo recomendado es de 2 GB.

- No tiene WIFI pero se puede insertar un módulo WIFI USB ya que si dispone de puerto USB.
- Se puede aumentar la cantidad de puertos USB mediante un HUB con alimentación independiente.
- No cuenta con un reloj de tiempo real.
- Tiene un puerto Ethernet que permite tener conexión a Internet.
- Tiene salida HDMI que permite la conexión a una pantalla con puerto HDMI y tener así una mini computadora.
- Bajo costo.

CPU	Arquitectura ARM11 a 700MHz (posible OC a 1GHz)
GPU	Broadcom VideoCore IV (capacidad 1080@30fps)
Memoria	256MB (modelo A) 512Mb (modelo B)
Almacenamiento	Tarjeta SD
Entradas/Salidas	2xUSB 2.0 Conector MIPI CSI (cámara propia) Video: Conector RCA, HDMI 1.4, Interfaz DSI (LCDs) Audio: jack 3,5 mm Internet: Ethernet RJ-45 GPIO,SPI,I2C,UART
Alimentación	Puerto micro-USB, consumo 700mA (3,5W)

Figura 13. Características Raspberry PI [23]

2.7.4. Raspberry PI vs Arduino.

Aunque Raspberry PI y Arduino pueden ser utilizadas en ocasiones para dar soluciones a problemas similares son totalmente distintas, Raspberry es un ordenador completo con todas las características de una computadora, soporta un todo sistema operativo, mientras que Arduino es un

microcontrolador, el cual es un componente de una computadora y no soporta un sistema operativo.

Tanto Raspberry PI como arduino son tarjetas pequeñas y baratas sin embargo cuando a velocidad se refiere Raspberry PI es el ganador, aproximadamente 40 veces más rápido que arduino, lo cual es muy entendible si se considera que se está comparando una computadora con un microcontrolador.

Raspberry PI puede ejecutar varias aplicaciones en segundo plano, es decir puede estar funcionando como un servidor PBX y a su vez como un servidor VPN mientras que arduino lleva a cabo un proceso único por un largo periodo de tiempo.

A pesar de que la Raspberry cuenta con mucho poder de cómputo en comparación a Arduino, esta llega a ser muy compleja para proyectos básicos de electrónica.



Se debe tomar en cuenta que la configuración del PI se puede dañar por desconectarlo sin un apagado adecuado.

Cuando se trata de aplicaciones de software como Asterisk, Raspberry PI es muy conveniente sin embargo Arduino por su simplicidad es mucho mejor para proyectos de hardware ya que tiene una capacidad analógica y en tiempo real lo que le permite trabajar perfectamente con cualquier tipo de sensor.

2.7.4.1. Hardware y software

Tabla 2

Raspberry pi vs Arduino

	Arduino 	Raspberry Pi modelo B 
Precio en dólares	\$30	\$35
Tamaño	7.6 x 1.9 x 6.4 cm	8.6cm x 5.4cm x 1.7cm
Memoria	0.002MB	512MB
Velocidad de reloj	16 MHz	700 MHz
OnBoard Network	Ninguna	10/100 wired Ethernet RJ45
Multitarea	No	Sí
Voltaje entrada	7 a 12 V	5 V
Memoria Flash	32KB	Tarjeta SD (2 a 16G)
Puertos USB	Uno	Dos
Sistema Ope.	Ninguno	Distribuciones de Linux
Entorno de desarrollo integrado (IDE)	Arduino	Scratch, IDLE, cualquiera con soporte Linux

2.8. DISEÑO DE ASTERISK COMO INTERFAZ GRÁFICA

2.8.1. PYTHON

2.8.1.1. Definición de Python

Python es un lenguaje de programación fácil de entender y de aprender, dispone de una serie de librerías que pueden distribuirse fácilmente y que facilitan el desarrollo rápido de aplicaciones en cualquier tipo de plataforma. Es muy útil en el manejo de archivos: lectura, escritura, configuración etc., es un software libre al alcance de todos y muy fácil de integrar con otras herramientas de programación y desarrollo.

Es fácil ampliar el intérprete Python con nuevas funciones y tipos de datos implementados en C y C++. Python es adecuado como lenguaje de extensión para aplicaciones adaptables al usuario.

Es un lenguaje de programación muy versátil, fuertemente tipado, imperativo y orientado a objetos, aunque contiene también características que lo convierten en un lenguaje de paradigma funcional [19].

La sintaxis de Python es realmente sencilla con un modo interactivo, se puede combinar con una librería gráfica para realizar interfaces gráficas inclusive para otros softwares, como es el caso de Asterisk.

2.8.1.2. Características de Python

- Simple y sencillo de aprender, libre y fuente abierta, orientado a objetos.
- Lenguaje de muy alto nivel: Cuando escribes programas en Python nunca debes preocuparte por detalles de bajo nivel, como manejar la memoria empleada por tu programa.

- Permite separar el programa en módulos reutilizables en otros programas en Python.
- Tiene integrados una gran variedad de módulos estándar que facilita la programación.
- Python es un lenguaje interpretado, lo cual puede ahorrarte mucho tiempo durante el desarrollo ya que no es necesario compilar ni enlazar [20].
- Los programas en Python son más cortos que sus programas equivalentes escritos en C++.
- Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola instrucción [20].
- No es necesario declarar variables o argumentos.
- El intérprete Python se puede enlazar con un programa escrito en C, Si necesitas que una pieza de código se ejecute muy rápido, puedes lograrlo escribiéndola en C y después combinarla con el programa de Python.
- Python ofrece mucho más chequeo de error que C.
- El intérprete de Python y la extensa biblioteca estándar puede distribuirse libremente.
- Python es portable, ésta característica es posible gracias a que Python funciona en varias plataformas sin requerir cambio alguno en el programa, siempre y cuando no se utilicen características que dependan del sistema operativo.

- **Incrustable:** Puedes insertar Python dentro de tu programa en C/C++ para ofrecer las facilidades de "scripting" dentro del mismo [21].
- **Librerías extendidas:** La librería estándar de Python es de hecho muy amplia. Puede ayudarte a hacer varias cosas que involucran: expresiones regulares, generación de documentos, evaluación de unidades, pruebas, procesos, bases de datos, navegadores web, CGI, ftp, correo electrónico, XML, XML-RPC, HTML, archivos WAV, criptografía, GUI(graphicaluser interfaces/interface gráfica del usuario) usando Tk, y también otras funciones dependientes del Sistema. [21].

2.8.2. PHP

2.8.2.1. Definición de PHP

PHP (Personal Home Page) es un lenguaje de programación libre, utilizado especialmente en el desarrollo web y que puede ser incrustado en HTML, simple de utilizar aunque también ofrece características avanzadas.

El desarrollo de PHP está centrado en el desarrollo de aplicaciones del lado del servidor pero tiene otras funcionalidades, además cuenta con un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

2.8.2.2. Características de PHP

- Permite modificar dinámicamente el contenido de una página WEB.
- Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas.
- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, SybasemSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en

formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

- Un sitio con páginas webs dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.

2.8.2.3. Funcionamiento PHP



Figura 14. Funcionamiento PHP [22]

A diferencia de Java o JavaScript que se ejecutan en el navegador PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap) [22].

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP.

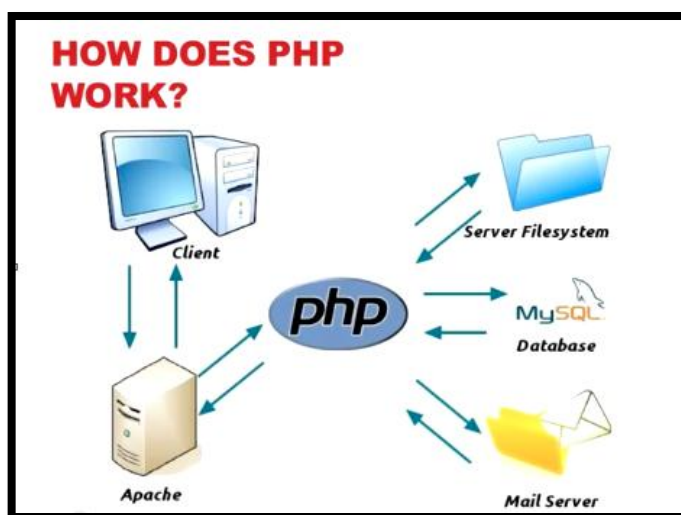


Figura 15. Cómo trabaja PHP

2.8.3. PHP VS PYTHON

- Python es muy fácil de aprender, de muy alto nivel, su curva de aprendizaje es inclusive más suave que la de PHP y Java.
- Tanto Python como PHP implementan una serie de librerías que facilitan la programación y permiten desarrollar interesantes aplicaciones.
- La documentación acerca de Python es escasa en comparación con la documentación que se encuentra acerca de PHP.
- En cuanto a velocidad Python se ejecuta con bytecode lo que lo hace muy veloz, colocándose entre java y PHP en cuanto a velocidad se refiere.

- Implementar Python en Web resulta más difícil que implementar PHP en Web.

Básicamente lo que hace que Python sea el programa favorito en el diseño de una interfaz gráfica para Asterisk es su simplicidad de su sintaxis, es muy fácil de entender y aprender, además que ofrece características muy similares a PHP en cuanto a velocidad, dispone de una serie de módulos que lo hace fácil de programar y sobre todo permite un fácil manejo de ficheros, necesario para la configuración de Asterisk.

CAPÍTULO 3 : IMPLEMENTACIÓN CENTRAL TELEFÓNICA IP

3.1. Topología

En el presente proyecto se presenta una topología de cuatro Raspberry Pi en las cuales tienen instalado Asterisk, funcionando en conjunto como centrales telefónicas.

Cada central está conectada a un router permitiendo que los equipos como: teléfonos, sofphones, se puedan conectar a las mismas. Entre las configuraciones con las que cuenta las centrales son: extensiones (SIP, IAX), voicemail, followme, musiconhold, ivr, features, transferencia de llamadas (ciega, atendida), registro de llamadas(CDR),facturación de las llamadas, call center (ACD), configuración para la comunicación entre las extensiones de dos centrales y para finalizar una de las centrales con ayuda de un Gateway de voz podrá recibir y realizar llamadas desde y hacia la red PSTN, todas estas funciones serán desarrolladas en el transcurso de este capítulo.

Además cabe mencionar que para acceder a la interfaz de la central donde se configuran las funciones antes mencionadas se utilizará un computador que se encuentre conectado a la misma de red de la central y se podrá acceder mediante el escritorio remoto.

3.2. Selección de Hardware

3.2.1. Servidor

El servidor se implementa en un computador de bajo costo denominado Raspberry Pi, de tamaño reducido ya que se encuentra construida en una placa del tamaño de una tarjeta de crédito y que cuenta con las siguientes características:

- Utiliza un chip Broadcom BCM2835 con un procesador ARM1176JZF a 700MHz.
- Un procesador gráfico (GPU).
- VideoCore IV
- 512 MB de memoria RAM (modelo B).

Este mini computador trabaja con un sistema operativo GNU/Linux, con varias distribuciones adaptadas para Raspberry Pi entre los que encontramos:

- NOOBS, el cual es una opción para los que deseen probar varios sistemas operativos, ya que en su primer arranque ofrece un menú donde permite seleccionar con que sistema operativo se desea trabajar, de acuerdo a nuestras necesidades. Además se puede adquirir una tarjeta SD con esta distribución ya instalada.
- Raspbian que es una distribución libre basada en Debian, optimizada para Raspberry Pi, es de los sistemas operativos más antiguos para este hardware, pero es una de las versiones ecomendadas para utilizarlo con Asterisk. Esta versión se la puede descargar desde la página oficial de Raspberry Pi.

Para la instalación de Raspbian es necesario una tarjeta SD con un tamaño mínimo de 4 GB. Los pasos para el proceso de instalación se detallan en la sección de anexos.

3.2.2. Enrutadores

Para la topología especificada en este proyecto se utilizó dos router Mikrotik RB951Ui-2HnD con las siguientes especificaciones técnicas:

Router con un CPU Atheros de nueva generación.

Cuenta con:

- Cinco puertos Ethernet
- Un puerto inalámbrico WIFI de 2,4 GHz de potencia 1 W 802.11b/g/n
- RouterOS como sistema operativo basado en un Kernel de Linux
- 60 MB en disco
- 128 MB de memoria RAM
- La función de entrada/salida PoE en un puerto.



Figura 16.Router Mikrotik

Los cuales nos permitirán simular la comunicación vía internet entre dos central es que se encuentren remotamente conectadas en redes LAN diferentes. Para seleccionar el tipo de router a utilizar se debe considerar equipos que soporten VPN (Site-to-site) o túneles para la transmisión segura entre los dos servidores.

VPN:es un túnel mediante el cual se puede transmitir información a través de internet de manera segura y de forma transparente para el cliente, esto

debido a que los paquetes son encriptados. La configuración de la VPN se detalla en la sección de anexos.

3.2.3. Gateways

Una de las funciones que se realizó en este proyecto fue la conexión de la central Asterisk con la red PSTN, para lo cual se utilizó un Gateway de voz Grandstream HT503 que cuenta con las siguientes especificaciones:

- Adaptador de alimentación, 12VDC, 0.5^a
- Puerto LAN (RJ-45)
- Puerto WAN (RJ-45)
- PHONE (RJ-11)
- LINE (RJ-11)
- LED de encendido
- LED del puerto LAN, WAN, PHONE y LINE

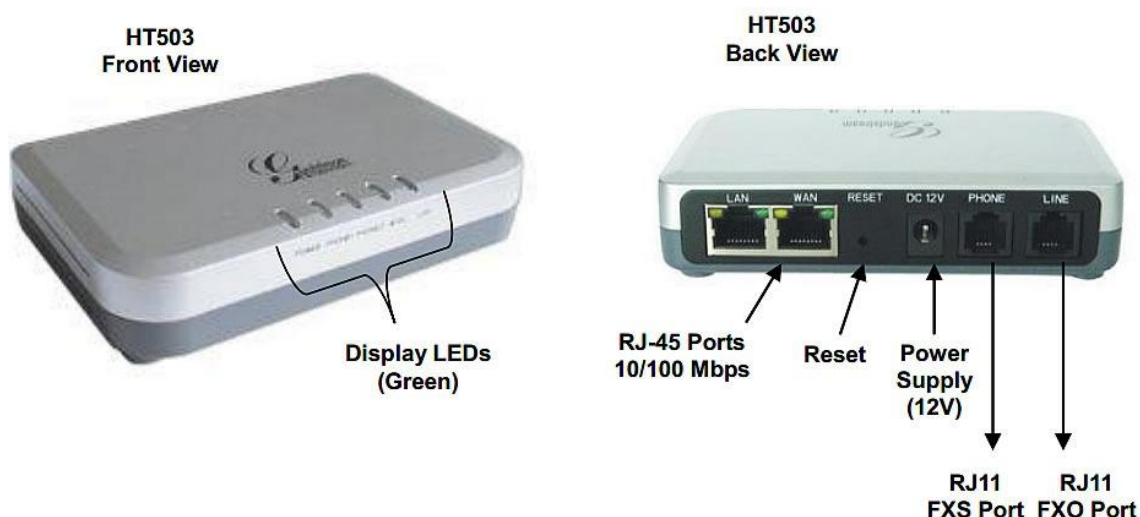


Figura 17. Gateway de Voz Grandstream

Este equipo cuenta con un puerto FXS que es el puerto PHONE y por el cual se podrá conectar un teléfono analógico a la central Asterisk, un puerto FXO que es el puerto LINE y donde se conecta la línea analógica para que la central se conecte a la red PSTN. Los dos puertos pueden tener cuentas SIP lo cual lo convierte en una ventaja ya que soporta llamadas simultáneas en ambos puertos.

Para la conexión a la red PSTN también existen tarjetas PCI las cuales cuentan con puertos FXS y FXO, tanto los gateways como las tarjetas cumplen la misma función y se obtiene los mismos resultados, la principal diferencia es que la tarjetas se conectan a los buses internos de los equipos, además no necesita de enchufes adicionales y sobre carga menos a la red IP, pero no es una alternativa cuando se trabaja con hardware de bajo costo como lo es raspberry pi.

Por ello se seleccionó un Gateway que es dispositivo externo al cual se conectan las líneas o teléfonos y además cuenta con una interfaz Ethernet que permite su configuración.

3.3. Selección del Software Asterisk

3.3.1. Instalación

Asterisk es un software que incorpora todas las funcionalidades de una centralita convencional y además incorpora funciones avanzadas a un bajo costo. Asterisk cuenta con varias versiones y la selección de una de ellas dependerá del entorno de trabajo si se desea estabilidad o nuevas funcionalidades.

El tipo de versión define durante cuánto tiempo va a ser soportada, por tanto tenemos:

LTS (Long TermSupport):son soportadas por 4 años, por lo tanto es recomendable en casos donde se requiera una central para trabajar con abonados masivos y si se busca estabilidad sin fallos en el funcionamiento y con mayor seguridad.

Estándar: están soportadas por un periodo más corto entre 6 a 8 semanas, donde existirán releases que incluirán cambios para corregir errores, además de actualizaciones de las funcionalidades y novedades, recomendable para proyectos de investigación o laboratorios.

Por lo tanto la versión seleccionada para este proyecto es 11.12.o una versión estándar y que nos ofrece servicios de valor añadido sin dejar a un lado la estabilidad y seguridad, su instalación se desarrollará en la sección de anexos.

3.3.2. Configuración de Asterisk

Actualmente existen varias distribuciones de Asterisk que incluyen su propia interfaz gráfica para la configuración de las funcionalidades que el cliente o usuarios requiera, por ejemplo: Elastix, Trixbox, etc, incluso existe una versión para raspberry denominada uelastix. El objetivo primordial de este proyecto es conocer desde sus archivos base como se configuran las funcionalidades, para partir de ello y diseñar una interfaz que ayude a los estudiantes a comprender como funciona Asterisk y en caso de requerir ajustarlo o acomodarlo a las necesidades del estudiante lo pueda realizar.

Por lo tanto para una central más dinámica y con un correcto funcionamiento se debe considerar lo siguiente:

Música en espera: en el caso de querer reproducir una canción diferente a las que Asterisk nos proporciona se debe convertir la misma en dos formatos que son alaw y G.722, esto debido a que cuando se reciba una llamada de la

PSTN se reproduzca en formato alaw, y cuando las llamadas se realicen entre las extensiones internas se reproduzca en formato G.722 para evitar el transcoding y evitar el consumo inútil de procesador.

Codecs: el objetivo de la voz sobre ip es explotar la calidad que esta nos ofrece, por ello para proporcionar una mejor experiencia al usuario al realizar las llamadas es recomendable utilizar el códec G.722, que es un códec wideband que ofrece mejor calidad, acentúa las frecuencias altas transmitiendo mayor información con el mismo ancho de banda que un códec alaw.

Es decir al utilizar los codecs alaw y ulaw la calidad de audio que proporcionan es la misma que cualquier otro teléfono. Además cabe mencionares que estos dos códec son idénticos, ulaw es igual a alaw pero desplazado, en el caso de continentes como Europa, Asia, África o Latinoamérica se usa el códec ulaw, pero reduciríamos el consumo del procesador utilizando alaw ya que Asterisk realiza el transcoding.

3.4. Archivos de Configuración de Asterisk

Después de haber culminado la instalación de Asterisk podemos observar todos los archivos de configuración con los que cuenta, ingresando a la carpeta *configs*, estos archivos contienen ejemplos de cómo y qué se puede configurar en cada uno de ellos. Por lo tanto se procede a copiar los archivos con los cuales se desea trabajar en la carpeta */etc/asterisk*, y es desde aquí que se comienza a crear las reglas de marcación (*dialplan*), configuración de las opciones avanzadas y configuración de las conexiones telefónicas (canales IAX y SIP).

De los archivos antes mencionados los más importantes son: *extensions.conf*, *sip.conf*.

3.4.1. Extensiones

Las extensiones son las que gestionan las llamadas y comunican a la líneas internas, además estas toman su comportamiento de acuerdo a como se le configure su contexto. En este proyecto se va a contar con extensiones iax y sip las cuales serán definidas en el archivo *iax.conf* y *sip.conf*.

3.4.1.1. SIP.CONF

En este archivo se configura el canal para el protocolo SIP y está conformado por los siguientes bloques:

[general]

Es donde se determina las opciones predeterminadas para todas las extensiones definidas dentro del archivo. Algunos de los parámetros que se define en esta sección son: el puerto de comunicación, la dirección IP de conexión al servidor, codecs de audio permitidos, etc, la definición de cada uno de estos parámetros dependerá de cómo queremos que funcione la extensión en nuestra central.

[extensión]

El nombre que se define dentro de los [] puede variar, depende de cómo el usuario desea identificar a las extensiones de su central, pueden ser nombres de los usuarios de las extensiones o números. Los parámetros que se configuran en esta sección se los describe a continuación pero se puede mencionar algunos como: contraseña de conexión al servidor, host desde donde se conecta, identificador de llamadas y el contexto asociado a dicha extensión en el dialplan.

Una configuración básica del archivo sip.conf, es como se muestra a continuación, en donde se define una de las extensiones configuradas en este proyecto.

[general]

updbindaddr=

0.0.0.0:5060

; define la interfaz de red donde se puede especificar la IP o "0.0.0.0" para que escuche todas y el puerto UDP (5060) utilizado conectar las extensiones SIP al servidor Asterisk.

allowguest = no

; parámetro de seguridad, no permite llamadas de extensiones SIP remotas sin contraseña.

srvlookup = yes

; permite hacer búsquedas de registros DNS para envío de emails, se lo debe activar cuando se utiliza proveedores externos.

alwaysauthreject = yes

; parámetro de seguridad cuando un usuario se intenta autenticar con credenciales erróneas.

lenguaje = en

; idioma que usara la extensión, depende de si se ha instalado locuciones de más de un idioma

[7001]

username = 7001

; usuario para conectar a la extensión

secret = 7001

; contraseña de la cuenta

callerid = 7001

; es opcional, permite que cuando se llame internamente en la pantalla figure la extensión y el nombre sino no lo tiene en la agenda.

; la extensión del buzón definido en el voicemail.

mailbox = 4001@default

; permite la interacción del usuario con el sistema y se definen como "user":autentica

type = friend

llamadas entrante, "peer" auténticallamadas

nat = no	<i>; se utiliza para convertir direcciones ip dentro de una red cuando se encuentra detrás de un corta fuegos como un router o firewall, SIP puede dar problemas con el uso de NAT.</i>
host = dynamic	<i>; especifica el host o IP desde la que el usuario se comunica con el servidor, si el usuario se puede conectar desde varias lps se lo coloca en "dynamic".</i>
context = extensiones- internas	<i>;contexto del dialplan asociado con este usuario.</i>
canreinvite = no	<i>;con "no" forzamos a que el streaming de audio de la llamada pase por Asterisk y no directamente entre los terminales, evitando los problemas de tener sonido en una dirección o no se tenga ningún sonido.</i>
qualify = yes	<i>; con "yes" se puede determinar cuando el dispositivo puede ser alcanzado. Asterisk por defecto considera que el dispositivo está presente cuando tiene una latencia menor de 2000 ms, por lo tanto se puede colocar "yes" o 2000ms.</i>
disallow = all	<i>;deshabilita todos los codecs de audio.</i>
allow = ulaw	<i>; habilita un códec, se puede poner varios codecs en un mismo usuario.</i>

salientesy "friend":para definir ambos tipos de llamadas.

3.4.1.2. IAX.CONF

En este archivo se configura el canal iax para el protocolo iax2 y las extensiones que usará dicho canal, la configuración es similar a la de una extensión sip incluso está definida por los mismos bloques.

Una configuración básica del archivo iax.conf, es como se muestra a continuación, en donde se define una de las extensiones configuradas en este proyecto.

[general]

updbindaddr = 0.0.0.0:4569

; define la interfaz de red donde se puede especificar la IP o "0.0.0.0" para que escuche todas y el puerto UDP (4569) utilizado conectar las extensiones IAX al servidor Asterisk.

srvlookup = yes

delayreject = yes

; mejora la seguridad contra ataques de fuerza bruta, retrasando los mensajes de autenticación.

alwaysauthreject = yes

autokill = yes

; utilizado para verificar el estado de las conexiones, cuando se encuentra en "yes" y no recibe un ACK en 2000ms se cancela la conexión, útil para terminar una llamada cuando no se tiene respuesta.

[5001]

username= 5001

secret = 5001

callerid= 5001

mailbox= 6001@default

type = friend
host = dynamic
context = extensiones-internas
canreinvite = no
qualify = yes
nat = no
disallow = all
allow = ulaw

3.4.1.3. Comparación entre IAX y SIP

En capítulos anteriores ya se describieron ciertas ventajas de IAX sobre SIP las cuales las vamos ampliar de mejor manera una vez que ya se conoce los parámetros que se configura en cada uno de los archivos.

Una de las ventajas que se menciona fue el **ancho de banda**, esto debido a que IAX utiliza un menor ancho de banda al codificar los mensajes en binario y al reducir las cabeceras de los mismos, mientras que SIP envía los mensajes en forma de texto. Probablemente en este punto cuando se hable de servidor SIP esto llegue a ser una ventaja ya que la información de señalización pasa por el servidor, pero la información de audio viaja de extremo a extremo sin pasar por el mismo, lo que no ocurre en IAX y a su vez en servidores IAX su principal desventaja es que al viajar todo en conjunto debe pasar por el servidor y esto aumenta el ancho de banda que se utiliza.

En cuanto al parámetro **NAT** se mencionó que se tiene ciertos inconvenientes en el flujo de audio cuando se lo configura en SIP, esto debido a que tanto la señalización como los datos viajan de manera separada requiriendo un servidor STUN para solucionar estos problemas, lo que no ocurre en IAX ya que la señalización y datos viajan conjuntamente.

IAX únicamente utiliza un solo **puerto** utilizando multiplexión o trunking para mandar la información de la señalización y los datos de las llamadas que se generen, siendo este puerto el 4569, mientras que SIP utiliza un puerto para la señalización que es el 5060 y 2 puertos RTP para cada conexión de audio. Por ejemplo para 50 llamadas simultáneas SIP utilizaría 100 puertos RTP más un puerto de señalización (5060). IAX utilizaría un solo puerto para todo, el 4569.

Para concluir, SIP es un protocolo utilizado con mayor popularidad debido a que ya se encuentra estandarizado por la IETF, por lo tanto la mayoría de los equipos cuentan con este protocolo, mientras que IAX aún no lo es, lo que hace complicado encontrar este tipo de protocolo en los equipos.

3.4.2. Buzón de Voz

El buzón de voz es similar a las antiguas contestadoras que el usuario debía adquirir para poder almacenar los mensajes cuando una llamada no era contestada, en la actualidad se puede decir que es similar al correo electrónico para almacenar correos de texto alojados en nuestro servidor Asterisk.

Por lo tanto el buzón de voz se encarga de almacenar mensajes de voz para posteriormente enviarlos a un correo electrónico avisando de los mismos al usuario destinatario.

La principal característica de las centrales modernas es este tipo de sistemas, y Asterisk cuenta con uno muy flexible, entre sus características podemos mencionar:

- Cuenta con saludos predeterminados y personalizados.
- Se puede asociar varias extensiones con un buzón y varios buzones con una extensión
- Contiene diferentes carpetas para organizar los mensajes de voz.
- Saludos de acuerdo al estado de usuario (ocupado o disponible)
- Se puede generar cuentas ilimitadas protegidas por contraseñas.
- Notificación de nuevo mensaje de voz vía correo electrónico con la posibilidad de adjuntar el archivo de voz.

En Asterisk este sistema puede funcionar de tres diferentes maneras que son:

- Con la ayuda de un menú implícito con una serie de funciones como lectura, manipulación y eliminación de los mensajes después de haber accedido al mismo.
- La segunda manera y la cual será desarrollada en este proyecto es re direccionando al correo electrónico, después de llegar al buzón personal dependerá del usuario como proceder con el correo.
- Finalmente mediante un sistema de almacenamiento específico, que puede ser servidor SQL, IMAP o servidores de buzón de voz compatibles.

Para poder configurar el buzón de voz se debe trabajar con el archivo ***voicemail.conf*** el cual lo analizaremos a continuación.

3.4.2.1. VOICEMAIL.CONF

Como se mencionó anteriormente, es en este archivo donde se configura el buzón de voz. Cuando se recibe una llamada y no es contestada o la línea

se encuentra ocupada, la llamada se re direcciona al buzón de voz donde se podrá guardar el mensaje y al culminar esta conexión el mensaje se enviara vía mail al correo.

Asterisk no envía emails ya que es una aplicación como cualquier otra que corre bajo Linux, lo que Asterisk hace es conectarse a un servidor SMTP instalado en el servidor, es este el encargado de enviar los emails. Otra posibilidad de realizar el envío de emails es utilizando un servidor remoto con una cuenta, el procedimiento es exactamente el mismo.

Un servidor SMTP para el envío de emails es SendMail, con el paso del tiempo y debido a su complejidad, poca fiabilidad surgieron otros servidores SMTP más ligeros. El principal problema de estos servidores se da al incremento de correos SPAM, debido a que los servidores y los proveedores verificaran si el correo proviene de un servidor oficial o de un servidor que se dedica al envío de SPAM es decir, no se puede considerar a esta aplicaciones como servidores originales debido a la falta de ciertas características como: MX en DNS dirección ipdinámica, etc.

Existen algunos sustitutos a sendmail como exim o postfix, en este proyecto se trabajó con exim que es un servidor que viene por defecto con Debian. Se sugiere que para evitar el problema antes mencionado se debería utilizar una cuenta de un servidor ya verificado y seguro que se lo configuró como se muestra en la sección de anexos.

El archivo voicemail.conf consta de tres bloques que son:

[general]

En este bloque se configura los parámetros generales del buzón de voz como cantidad de mensajes, tiempo para cada grabación, número de reintentos etc.

[zonemessages]

Aquí se define las zonas horarias y su formato, la hora para distintos usuarios no es la misma y para poder indicar la hora de recepción del mensaje es necesarios fijar distintas zonas horarias.

[default]

Dentro de este contexto se define los usuarios para el buzón de voz, todos los usuarios pueden estar en el mismo contexto o contar con más de una contexto.

La configuración básica del archivo voicemail.conf, se muestra a continuación.

[general]

format = wav	<i>; tipo de formato que se guardaran los mensajes de voz, puede ser gsm o wav</i>
serveremail =asterisk@voztovoice.org	<i>; nombre mediante el cual se identificara el por correo electrónico el remitente.</i>
attach = yes	<i>;indica si el mensaje de voz se envía vía correo electrónico</i>
maxmsg = 100	<i>;número máximo de mensajes en el buzón de voz.</i>
maxsecs = 300	<i>; duración máxima en segundos de cada mensaje de voz.</i>
minsecs = 3	<i>;Valor mínimo en segundos que debe durar la grabación para ser reconocida y enviada al destinatario.</i>

maxgreet = 60	<i>; duración máxima en segundos de duración del mensaje de bienvenida.</i>
maxsilence = 10	<i>; valor en segundos del silencio para que el servidor corte la llamada al buzón de voz, si se desea un valor infinito se debe colocar 0</i>
silencethreshold = 128	
maxlogins = 3	<i>; número de intentos permitidos para ingresar al buzón de voz</i>
moveheard = yes	<i>; mueve los mensajes escuchados a la carpeta old.</i>
userscontext=default	<i>; contexto definido para los usuarios.</i>
pbxskip=yes	<i>; omite la abreviación pbx en el remitente del mensaje</i>
fromstring=VozToVoice	<i>; nombre del remitente que aparecerá en el correo electrónico</i>
mailcmd = /usr/sbin/sendmail-t	<i>; programa predefinido para el envío de mensajes</i>
tz=Colombia	<i>; Huso horario definido para indicar la fecha y hora del correo de voz recibido.</i>
attachfmt = wav	<i>; formato de audio con que se enviaran el mensaje de voz adjunto al correo electrónico.</i>
saycid = yes	<i>; parámetro que permite escuchar los datos del usuario que dejo el mensaje (número de teléfono y nombre).</i>

sayduration = no	<i>; indicar o no la duración del mensaje de voz</i>
saydurationm = 2	<i>; duración mínima del correo de voz para que se anunciada.</i>
listen-control-forward-key = #	<i>; tecla para adelantar el mensaje que se está escuchando.</i>
listen-control-reverse-key = *	<i>;tecla para ir atrás en el mensaje que se está escuchando.</i>
listen-control-pause-key = 0	<i>; tecla para poner en pause el mensaje</i>
listen-control-restart-key = 2	<i>; tecla para volver a escuchar el mensaje</i>
listen-control-stop-key = 13456789	<i>;teclas para parar el mensaje y volver al menú del contestador.</i>
backupdeleted = 100	<i>;número de mensajes que se podrán almacenar en la carpeta Deleted.</i>
[zone messages]	
colombia=America/Bogota vm-received' aebY 'digits/at' HM	<i>;definición de zonashorarias</i>
eastern=America/New_York vm-received' Q 'digits/at' IMp	
central=America/Chicago vm-received' Q 'digits/at' Imp	
central24=America/Chicago vm-received' q 'digits/at'H N 'hours'	
military=Zulu vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'	
european=Europe/Copenhagen vm-received' a d b 'digits/at' HM.	
[default]	<i>;extensión, contraseña y correo de usuario.</i>
4001 => 1204,Majo,mj@gmail.com	

Se puede agregar el número de cuentas que se desee, una vez terminada la configuración, en el dialplan se configura una extensión la cual permita al usuarios llamar para escuchar los mensajes, borrarlos, etc.

3.4.3. FOLLOWME.CONF

Es una aplicación de Asterisk que permite configurar a un grupo de extensiones, celulares o fijos para que se encuentren relacionados a una extensión. Es decir cuando a esta extensión le ingresa una llamada y esta no puede ser contestada la llamada automáticamente se re direcciona grupo de extensiones, celular o fijo que se la haya configurado. Para que la llamada sea atendida primero ingresará a un menú donde se le indicara quien es el remitente y si desea atender o no la llamada. Si la llamada no ha sido atendida seguirá el proceso normal de cualquier otra llamada que será invocar al viciemail para dejar un mensaje de voz.

Al igual que archivos antes descritos, esta cuenta con dos bloques en donde el contexto general se configura los parámetros generales, seguido del contexto con el cual queremos identificar a la extensión o extensiones que se desea contar con esta aplicación.

La configuración básica del archivo followme.conf, se muestra a continuación.

[general]

featuredigittimeout => 5000	<i>; tiempo en milisegundos que se tiene para aceptar o rechazar la llamada.</i>
takecall =>1	<i>; para aceptar la llamada se deberá digitar el número 1.</i>
declinecall =>2	<i>; para rechazar la llamada se deberá digitar el número 2.</i>

call-from-prompt =>followme/call-from	<i>; voz predefinida que se escuchara anunciándole quien está llamando.</i>
norecording-prompt =>followme/no-recording	<i>;voz predefinida en caso de que el llamante no registre su nombre o si la opción para registrarlo no está activada.</i>
options-prompt =>followme/options	<i>; voz que anunciara que debe digitar 1 para aceptar la llamada o 2 para rechazarla.</i>
pls_hold_prompt=>followme/pls-hold-while-try	<i>;voz que se escuchara para indicar al llamante que espera en línea mientras se transfiere la llamada.</i>
status_prompt=>followme/status	<i>; voz que anunciara el status de la persona llamada.</i>
sorry-prompt =>followme/sorry	<i>;voz que anunciará si no se ha podido contactar a la persona buscada después de todos los desvíos.</i>
[3001]	<i>;número de la extensión que se desea configurar.</i>
context = extensiones-internas	<i>;contexto que usará la aplicación followme para localizar el llamado.</i>
number = 3003,10	<i>;definición de la lista de números que se intentará llamar para localizar al destinatario, si el destinatario no contesta a la extensión 3001 se desviara a la extensión 3003 por 10 segundos.</i>
number = 90987009999,10	<i>;número celular para desviar la llamada, iniciando con prefijo para la salida a la PSTN que se analizara en archivos posteriores.</i>

3.4.4. FEATURES.CONF

Este archivo permite personalizar los códigos para proveer a la centralita de opciones con un valor añadido. Algunas de las configuraciones que se puede realizar son: trasferencias de llamadas, parqueo de llamadas, grabación de llamadas, sala de conferencias, etc. De las cuales desarrollaremos transferencia de llamadas y parqueo de llamadas.

3.4.4.1. Transferencia de Llamadas

Una gran parte de los teléfonos incorporan teclas que sirven para realizar transferencias de llamadas de forma transparente para Asterisk, pero en ciertas ocasiones los teléfonos no cuentan con estas opciones y es recomendable que ya vengan configurados en la central.

Dentro de las transferencias se tiene dos tipos que son las transferencias ciegas y atendidas.

Transferencia ciega: este tipo de transferencia se presenta cuando se desvía una llamada y no se anuncia de la llamada a la extensión de destino.

Transferencia atendida: se presenta cuando se desvía una llamada hacia una extensión destino, pero primero es anunciada para ver si la extensión puede atender o no la llamada.

3.4.4.2. Parqueo de Llamadas

Existe ocasiones en las que es necesario dejar en espera una llamada debido a que se pueden presentar escenarios como: llamadas con mayor prioridad o si necesita levantarse por algún motivo. Por lo tanto lo que permite esta función es dejarla estacionada a la llamada para que no se pierda y después de un tiempo recuperarla.

Estas aplicaciones son configuradas en el archivo `features.conf`, conformada por tres contextos que son:

[general]

En este contexto se define los parámetros generales de parqueo, almacenamiento, transferencia de llamadas, etc.

[featuremap]

Contexto en el cual se define la combinación de teclas a pulsar para realizar cualquiera de las opciones antes mencionadas. Para que funcione esta configuración, Asterisk debe permitir un puente entre las llamadas, es decir debe estar definido en el `extensions.conf` y en la aplicación `Dial()`, la cual será desarrollada posteriormente. La definición de esta configuración se presenta con las siguientes letras:

- Transferencia atendida: To t
- Transferencias ciegas: T o t
- Poner la llamada en espera: K o k.

[applicationmap]

En el contexto final se puede añadir funcionalidades que se puede utilizar durante una llamada.

La configuración básica del archivo `features.conf`, se muestra a continuación.

[general]

`parkext => 700`

; extensión en la cual se parqueara una llamada.

`parkpos => 701-710`

; extensiones reservadas para el parqueo de las llamadas.

`context => parkedcalls`

; contextopara el extensions.conf

parkingtime => 45	<i>; tiempo en segundos que una llamada puede estar parqueada.</i>
comebacktoorigin = yes	<i>;con “yes” después del tiempo del parkingtime la llamada regresa a la extensión que la parqueo.</i>
parkedplay = caller	<i>;a quien enviar el beep cuando se llama a la extensión parqueada, parked(canal parqueado), caller (llamante) y both(ambos).</i>
parkedcalltransfers = caller	<i>; habilita o deshabilita la secuencia de tonos para transferir la llamada cuando ya ha sido transferida.</i>
transferdigittimeout =>5	<i>;tiempo en segundos de espera entre dígitos cuando se transfiere una llamada.</i>
xfersound = beep	<i>;sonido que indicara que la llamada con transferencia atendida se completó.</i>
xferfailsound = beeper	<i>;sonido que indicara que la transferencia falló.</i>
featuredigittimeout= 2000	<i>; tiempo en segundos entre dígitos para activar las funcionalidades que se describen en el siguiente contexto.</i>
atxfernoanswertimeout=15	<i>;tiempo máximo para que una llamada se contestada con el método atendido.</i>
atxferdropcall = no	<i>; “no” si se transfiere una llamada con el método atendido cuelga antes de ser transferida la llamada, asterisk devuelve la llamada a quien la estaba transfiriendo.</i>

atxferloopdelay = 10	<i>; tiempo en segundos de espera antes de devolver la llamada con atxferdropcall = no.</i>
atxfercallbackretries = 2	<i>; número de veces que se intentara devolver la llamada.</i>
[featuremap]	
blindxfer => #	<i>; tecla que se debe digitar seguido del número de la extensión para realizar una transferencia ciega.</i>
disconnect => *	<i>; tecla para terminar la llamada</i>
atxfer => *2	<i>; tecla que se debe digitar seguido del número de la extensión para realizar una transferencia atendida.</i>
parkcall => *7	<i>; secuencia de teclas para parquear una llamada, se puede digitar de esta forma o directamente la extensión 700 con #700.</i>

3.4.5. Distribución Automática de Llamadas

3.4.5.1. Call Center

Es un centro de Atención de llamadas frecuentemente utilizado por empresas para crear un nexo con cliente, donde el propósito principal es atender un gran número de llamadas entrantes y salientes, estas llamadas pueden tener diversos objetivos como: soportes técnicos, empresas de telemarketing, departamentos de atención al cliente, etc.

El grupo humano encargado de atender las llamadas son los denominados **Agentes**, los cuales son evaluados bajo ciertos parámetros en conjunto con la calidad de las llamadas, estos son: número de llamadas recibidas, número de llamadas realizadas, duración de las llamadas, tiempos medios, tiempos

de respuesta, disponibilidad de agentes. Estos parámetros ayudaran a verificar si el trabajo se está realizando correctamente obteniendo un cliente satisfecho lo que traerá ganancias a la empresa.

Asterisk cuenta con sistema de distribución de llamadas que permite configurar diferentes algoritmos para una mejor gestión de las llamadas dependiendo de las necesidades del cliente. Un componente importante de un call center son las denominadas colas, que cumplen con las siguientes funciones:

- Las llamadas entrantes son colocadas en la cola de espera en el caso de que ningún agente puede atender dicha llamada.
- Permite la reproducción de música en espera mientras el usuario se encuentra encolado hasta que la llamada sea atendida, colocar música en espera es darle una toque diferente a la central debido a que escuchar el tono de timbrado por un cierto tiempo llega a ser cansado para el usuario.
- Existen diferentes algoritmos de distribución que serán seleccionados dependiendo del número de agentes disponibles, tamaño de cola, tiempos de espera a clientes, etc.
- Para que las llamadas sean atendidas, un grupo de agentes debe registrarse, este registro puede ser de manera estática o dinámica. **Estática:** el agente solo puede ingresar al sistema desde la misma extensión o equipo terminal, y **dinámica:** los agentes se pueden loguear desde diferentes extensiones o equipos terminales.

Por lo tanto los archivos de Asterisk que serán configurados para crear un call center son: *queues.conf*, *agents.conf* y *extensions.conf*, además se debe contar con un grupo de extensiones configuradas en el *sip.conf*.

Si se desea configurar agentes estáticos no es necesario configurar el archivo *agents.conf*, en el caso de agentes dinámicos se debe configurar como se muestra a continuación. Al igual que en archivos antes mencionados cuenta con un contexto general donde se configurará parámetros generales, seguido por el contexto *agents*, donde se configura los agentes que se utilizaran en la cola.

3.4.5.2. AGENTS.CONF

[general]

multiplelogin=yes ;permite a una misma extensión conectarse como agente múltiple.

[agents]

maxlogintries = 5 ;número de intentos máximo que el agente tiene para loguearse.

autologoff = 15 ; tiempo máximo que tiene un agente para contestar, sino automáticamente se desconectara.

autologoffunavail = no ; si el agente se vuelve no disponible, el agente se desconectar automáticamente.

endcall = yes ; los agentes terminan la llamada con *

ackall=yes ; los agentes se registran con #.

acceptdmf=# ; tonotdmf de # para el registro.

enddtmf=* ; tonotdmf de * para finalizar.

;updatecdr=no ; para que se registre en el cdr.

savecallsin=/var/calls ; directorio donde se guardaran las grabaciones de las llamadas.

recordedagentcalls=yes ; permite grabar las llamadas de un agente.

agent => ; se define los agentes que se van a utilizar en la cola, seguidos de la contraseña y el nombre del agente.
1001,4321,MariaJoseChavez

Una vez configurado los agentes, se procede a configurar las colas, en este punto se mostrará la configuración de las colas para agentes dinámicos (contexto colas), y las colas para los agentes estáticos (contexto colas1). Al igual que en la mayoría de los archivos consta de un contexto general donde se definirán los parámetros generales. Para la configuración de las colas se trabajará en el archivo *queues.conf*.

3.4.5.3. QUEUES.CONF

[general]

`persistentmembers = yes` ; se coloca "yes" para guardar la configuración cuando se trabaja con agents dinámicos, es decir cuando Asterisk arranque nuevamente colocara a cada agentes en su respectiva cola.

`autofill = yes`

; este parámetro permite que los usuarios sean atendidos de forma paralela hasta que haya agentes disponibles, sino se lo define como "yes" la forma de trabajar será que un usuario debe esperar estar primero en la cola para ser atendido aun si existe agentes disponibles.

[colas]

`strategy = ringall`

; contexto en donde se definirá el comportamiento de la cola.

; se define el tipo de estrategia que se va utilizar para atender las colas, entre las opciones tenemos: **ringall**: llama a todos los agentes a la vez y el primero que conteste atenderá la llamada, **leastrecent**: asigna la siguiente llamada al agente que más tiempo lleve sin atender una llamada, **fewestcalls**: asigna la siguiente llamada al agente que menos llamadas haya atendido,

	<i>random: asigna la siguiente llamada aleatoriamente a cualquier agente disponible,</i>
	<i>rrmemory: distribuye las llamadas por turnos entre los agentes disponibles y recuerda el último agente que le intento llamar, wrandom: asigna la llamada aleatoriamente usando una métrica basada en penalidades.</i>
timeout = 15	<i>; tiempo en segundos en el cual permanece timbrando antes de considerar que la llamada no va a ser contestada.</i>
retry = 5	<i>; tiempo en segundos que se debe esperar antes de volver a timbrar a otro agente.</i>
wrapuptime = 0	<i>; tiempo en segundos que debe expirar después de terminar una llamada y pasar otra llamada al agente.</i>
maxlen = 0	<i>; número máximo de llamadas que se puede colocar en una cola, con 0 no tiene ningún límite.</i>
announce-	<i>; tiempo en el cual se le anunciará al usuario en qué posición se encuentra en la cola.</i>
frequency=0	<i>; anuncia junto con la posición el tiempo estimado de espera.</i>
announce-holdtime=	<i>; con "no" no se timbrará a los agentes que se encuentren ocupados.</i>
no	<i>; configuración de agentes dinámicos, que formarán parte de esta cola.</i>
ringinuse = no	<i>; contexto donde se definen agentes estáticos. La diferencia entre colas1 y colas es que aquí se definen las extensiones que formarán parte de la cola.</i>
member=>Agent/1001	
member=>Agent/1002	
[colas1]	
member=>SIP/7001	
member=>SIP/7002	

La cantidad de colas que se desee configurar dependerá de la aplicación para la cual se prestará el call center, en el caso de los agentes dinámicos para que las extensiones pasen a ser agentes se debe loguear ingresando el número del agente con el cual se va a registrar y la contraseña, al contrario de los agentes estáticos, que en la configuración de la cola ya se define que extensiones permanentemente forman parte de ella.

3.4.5.4. Música en espera

Para que el cliente se sienta en un ambiente cómodo esperando hasta poder ser atendido por un agente se sugiere colocar música en espera, que será el siguiente archivo que se configurará.

La aplicación de Asterisk se llama MusicOnHold(MOH) y utiliza ficheros en formatos MP3 y WAV, si a la central se le configura para que funcione como call center se debe considerar que trabajará con una gran cantidad de llamadas y esto implica que aumente su capacidad de procesamiento.

En este caso debido a que Asterisk buscará el formato que menos coste le supone al códec que se está usando se sugiere transformar los formatos a SLIN, que es la opción más barata.

Para transformarla a SLIN se necesita de sox que es una de las librerías que se instaló cuando se instaló Asterisk y se ejecuta el siguiente comando: `sox music.wav -t raw -r 8000 -s -2 -c 1 music.sln`

En el caso del presente proyecto no se manejará un gran número de llamadas, por tal motivo se continuo trabajando con formato WAV. Los archivos que se utilizan para esta aplicación es: *musiconhold.conf* y *extensions.conf*.

3.4.5.5. MUSICONHOLD.CONF

```
[default]
mode=files
directory= /var/lib/asterisk/moh
```

Solo es necesario definir la dirección a la carpeta donde se encuentra la música.

3.4.6. Comunicación de centrales Asterisk

En este proyecto se presenta la configuración de una comunicación entre dos centrales Asterisk que pueden encontrarse remotamente ubicadas, la comunicación se puede realizar mediante un enlace dedicado de un proveedor de VoIP, pero en este caso se va a simular una comunicación mediante internet con la ayuda de una VPN.

Una VPN es un túnel por donde se puede transportar información utilizando como medio de transporte Internet, los paquetes viajan de forma segura ya que son encriptados.

Además se utilizará un enlace troncal bajo protocolo IAX por las ventajas que este nos ofrece. Un ejemplo en la cual se podría realizar esta configuración se presenta cuando es necesario comunicar una central matriz de un lugar específico con sus sucursales.

Cada una de las centrales cuenta con su propia red LAN con la ayuda de dos routers Mikrotik, en dichos routers se configurará una VPN la que se puede observar en los anexos.

La configuración de la troncal se la realizara en el archivo iax.conf como se muestra a continuación:

Red de la Central Principal: 192.168.2.9

Red de la Central Sucursal: 192.168.3.6

Central Principal

```
[servidor4]
username= servidor3
type=friend
secret=123
host=192.168.3.6
permit= 192.168.3.6
auth=md5
disallow=all
allow=ulaw
context= extensions-internas
peercontext=extensions-internas
trunk=yes
```

Central Sucursal

```
[servidor3]
username= servidor4
type=friend
secret=123
host=192.168.2.9
permit= 192.168.2.9
auth=md5
disallow=all
allow=ulaw
context= extensions-internas
peercontext=extensions-internas
trunk=yes
```

Como se observa los parámetros configurados son similares a los configurados en una extensión normal. Las diferencias son:

- Se debe contar con la misma contraseña en ambas centrales para poder establecer la comunicación
- La IP configurada debe ser la de la central contraria.
- Se debe configurar el parámetro **trunk** "yes", para activar la función de troncal.
- Se debe definir el contexto de la central contraria que se define en el extensions y no necesariamente debe ser la misma.

El *username* depende de la persona que lo configure, en este se definió con el nombre de la troncal contraria para considerar que es aquí donde se la define para comunicar con la propia central, y en *auth* se utilizó una encriptación md5 para la comunicación.

3.4.7. Comunicación con la PSTN

La PSTN es la Red telefónica conmutada o sus siglas en inglés *Public Switched Telephone Network* que se encuentra totalmente desplegada en el país al contrario de la VoIP, pero la VoIP nos permite un ahorro notorio especialmente en empresas que cuentan con varias extensiones, ya que permite la comunicación entre ellas sin la necesidad de contratar líneas tradicionales. El número de líneas tradicionales que se debería contratar disminuye o tan solo se necesitaría una y esto debido a que la central no puede comunicarse con números celulares, provinciales, etc por el despliegue de la red antes mencionada.

Para la comunicación es necesario configurar el Gateway de voz Grandstream y en Asterisk se realice una configuración sobre el archivo *sip.conf* ya que el Gateway soporta este protocolo. Con la línea tradicional conectada al Gateway las llamadas a esta línea serán recibidas en la extensión creada.

En sip.conf:

[200]	; nombre de la extensión que se definirá en extensions y en el Gateway de voz
username = 200	
context= extensiones-internas	; el contexto en donde se definirá la extensión en el extensions.
secret = 123	; contraseña configurada en el fxoport del Gateway de voz.
type = peer	; host del Gateway de voz
host = 192.168.2.10	
port=5062	; modo de los tonos dtmf que se definirá en el Gateway de voz.
dtmfmode=rfc2833	
canreinvite=no	
disallow=all	

allow = alaw

El resto de parámetros configurados son similares a los antes mencionados en la configuración de extensiones en el sip.conf.

3.4.8. CDR

Se configura los siguientes archivos de configuración de ASTERISK:

1.- En el archivo **Modules.conf** agregamos al final:

```
load => cdr_csv.so
```

El módulo configurado cdr_csv es muy simple, permite registrar el CDR en valores separados por comas es decir un archivo CSV, dicho archivo lo podemos encontrar en: /var/log/asterisk/cdr-csv/Master.csv, pero antes se debe configurar el archivo CDR.conf.

2.- Configuramos el archivo **CDR.conf** con lo siguiente:

```
[general]
enable = yes
unanswered = no
batch = no
size = 100
time = 300
scheduleronly = no
safeshutdown = yes

[csv]
usegmtime = no
loguniqueid = no
loguserfield = no
accountlogs = yes
```

Figura 18. Configuración del CDR

Tabla 3

Parámetros del CDR

Parámetro	Valores	Descripción
enable	yes/no	Activa o desactiva la utilización del CDR.
unanswered	yes/no	Registrar las llamadas no atendidas
batch	yes/no	Permite almacenar los datos de las llamadas en un buffer, para luego enviarlos a donde quisiéramos almacenar. OJO! que activando esta opción, si bien se liberaría un poco de procesamiento a Asterisk, puede ocasionar posibles pérdidas de datos, en caso que Asterisk se detenga.
size	Número entero	En caso que se active la opción de Batch, se define la cantidad de registros de CDR que se acumularán en el buffer, antes de ser enviados al medio de almacenamiento.
time	Número entero	En caso que se active la opción de Batch, se define la cantidad de tiempo máximo que deberán estar los registros almacenados en el buffer.
scheduleronly	yes/no	Asterisk utiliza un "scheduler" interno, para determinar en qué momentos se envían los registros al medio de almacenamiento. Dicho envío se puede hacer en el mismo "thread" donde está el "scheduler" o bien crear un nuevo "thread" para realizar esta tarea. En caso que se configure un proceso Batch con pocos registros, el envío puede estar en el "thread" del "scheduler" (Configurar este parámetro en "yes"). Ahora si se configuró un Batch de mayor tamaño, por ejemplo size=10, es recomendable realizar esta tarea en un nuevo "thread" (Configurar este parámetro en "no").
usegmttime	no	Utiliza GMT en lugar de la hora local. El valor predeterminado es no.
loguniqueid	no	Variable Identificación Única del CDR, El valor predeterminado es no.
loguserfield	no	variable Campo de usuariodelCDR, El valor predeterminado es no.
accountlogs	yes	Crear un archivo CSV separado para cada valor distinto de la variable CDR accountcode, El valor predeterminado es sí

Una vez realizadas dichas configuraciones se va a obtener el archivo Master.csv mencionado anteriormente con toda la información de nuestro CDR separada por comas, Aunque Asterisk tiene módulos que permiten obtener el CDR en base de datos para el presente proyecto se ha utilizado el archivo Master.csv para posteriormente cargarlo en una base de datos SQLITE, esto con la finalidad de no utilizar mayor cantidad de recursos de la tarjeta RASPBERRY PI pues para obtener la información del CDR directamente en una base de datos es necesario la instalación de nuevos módulos que resultan innecesarios tomando en cuenta la facilidad de transformar un archivo .CSV en una base de datos SQLITE.

3.4.9. EXTENSIONS.CONF

Este archivo es el principal a la hora de hacer funcionar la central. Unavez que se configura cada una de las funciones en los diferentes archivos se procede a definir el plan de marcado o dialplan que realizará la central para cada contexto y cada usuario.

El formato del dialplan para definir extensiones, aplicaciones, etc, es el siguiente:

exten =>patrón, índice, acción(parámetros)

Donde:

exten : todas la líneas del dialplan comienzan con esta palabra, y hace referencia a la definición de una extensión seguida de la acción que se va a realizar, a su vez se puede utilizar la palabra **“same”** que realiza la misma función pero simplifica la configuración, ya que únicamente se puede definir una sola vez la extensión seguido de la acción a realizar.

Patrón: es donde se define la extensión que realizará la acción configurada, la extensión se le puede definir de forma numérica como “7001”

o combinaciones complejas que abarquen varias extensiones a la vez como se muestra a continuación:

Los patrones comienzan con guion bajo “_” y seguido de:

- X acepta números del 0 a 9.
- Z acepta números del 1 a 9
- N acepta números del 2 a 9
- [x-y] acepta cualquier número de “x” a “y”
- [xyz] representa las cifras “x”, “y” o “z”
- . representa una o más repeticiones
- ! representa cero o más repeticiones

Índice: es el orden en el que se van a realizar las acciones, en Asterisk se debe definir este orden, y puede ser mediante un orden explícito con números, es decir : acción 1, acción 2 y así sucesivamente. Otra manera de definir las prioridades es colocando de arriba hacia abajo las acciones pero se lo define con el valor “n” para que se realicen de forma consecutiva.

Acción (parámetros): donde se define la acción a ejecutar, estas serán definidos en los contextos que se detallan a continuación.

Una vez que se conoce bajo que formato se va a configurar, se procede a definir cada contexto configurado en el *extensions.conf*. Al igual que los archivos antes definidos, se le puede configurar diferentes contextos teniendo en común el **[general]**.

En realidad estas configuraciones no son de mayor importancia para nuestro plan de marcado por lo cual se puede mantener los valores por defecto.

[default]

exten => _X.,1,Hangup(21)

exten => s,1,Hangup(21)

Este contexto se lo define por seguridad, ya que si existe algún usuario anónimo que no tiene ninguna contexto definido se colgará la llamada, evitando que un usuario son autenticar pueda realizar llamadas con costo a través de nuestra central.

[parqueo]

exten => 701, hint, park:701@parkedcalls

exten => 702, hint, park:702@parkedcalls

exten => 703, hint, park:703@parkedcalls

El contexto *parqueo* es donde se define las extensiones que servirán para dejar a una llamada parqueada o en espera, por lo tanto lo que se observa es como se relaciona la extensión con el contexto definido *parkedcalls* del *features.conf* para que realice la acción de *parqueo*.

El siguiente contexto es *extensiones-internas* donde se define las extensiones sip, iax anteriormente configuradas en sus respectivos archivos, además se configura la comunicación hacia la PSTN y entre centrales.

[extensiones-internas]

Plan de marcado desde y hacia la PSTN

exten => 200,1,Goto(ivr,s,1)

exten => _9X.,1,Dial(SIP/200/\${EXTEN:1})

La primera línea se define para recibir una llamada en la central, por lo tanto la extensión que se encuentra como patrón es la configurada en el

archivo sip.conf y su vez en el Gateway de voz, una vez que la llamada ingrese será recibida en dicha extensión con prioridad 1 y direccionada al contexto ivr a la extensión “s” y prioridad 1. La extensión “s” es una extensión especial de Asterisk que indica inicio ya que viene de start.

La siguiente línea permitirá realizar llamadas hacia la PSTN, el patrón está definido para números que empiecen con 9 seguido de la cantidad de números entre 0 a 9 dependiendo de las cifras del número celular, provincial, convencional, etc. El número 9 es el prefijo que tendrán que marcar las extensiones para conectarse a la PSTN, viene definido con prioridad 1, seguido de la aplicación Dial que permite colocar y conectar una llamada al canal definido, en este caso un canal SIPj unto a `_${EXTEN}:1` que contiene el número marcado quitándole el primer dígito es este caso el 9 que es el prefijo, por lo tanto este número será enviado mediante la extensión sip al gateway para que se conecte a la PSTN.

Plan de marcado entre centrales.

```
exten => _5XXXX,1,Dial(IAX2/servidor4/${EXTEN},90,tr)
```

```
exten => _5XXXX,2,Hangup
```

Inicialmente se cuenta con un patrón, como se mencionó en la configuración para la comunicación entre centrales, cada una cuenta con su prefijo el cual deberá ser marcado para conectar a la central adjunta, se utiliza un canal IAX2 ya que se creó una troncal de este tipo, definida en el contextoServidor4, con `_${EXTEN}` se toma la extensión marcada, además se configuro parámetros en el Dial como: un timeout de 90 (s), t que permite a la persona a la cual se le llamo pueda transferir la llamada, y r permite definir un tono especialmente cuando se marca múltiples canales.

Las siguientes líneas de configuración que se detallaran a continuación permiten colocar en el plan de marcado las extensiones SIP tanto para

comunicación interna de la central como para la comunicación hacia la central adjunta.

Para la comunicación interna en la central.

exten => 3001,1,Dial(SIP/3001,20,Tt)

exten => 3001,2,Followme(3001,san)

exten => 3001,n,VoiceMail(4001@default)

Se define la extensión seguido de las aplicaciones que debe realizar, cada una con su prioridad para que tenga un orden de ejecución, en este caso se cuenta con un patrón numérico, se empieza con prioridad 1 la aplicación Dial que permite colocar la llamada en un canal, en este caso SIP, seguido de los parámetros T, t que permiten al usuario llamado y llamante transferir la llamada pulsando “#” respectivamente.

La siguiente aplicación con prioridad 2 es followme, definida en apartados anteriores, la función cuenta con los siguientes parámetros: contexto (3001) configurado en followme.conf, “s” anunciará al llamante el estado de conexión de la extensión antes de empezar a direccionar la llamada a otros números, “a” permite grabar el nombre de la persona que establece la llamada para ser anunciado al receptor cuando conteste y pueda aceptar o rechazar la llamada, “n” si después de haber llamado a todos los números asignados que se buscara al receptor no se logra localizar a la persona, se escuchará una voz indicando que no se ha podido contactar al destinatario.

Para finalizar a la extensión se le agrego el buzón de voz con la aplicación voicemail, dentro de la función se define la extensión creada para el buzón relacionada con el contexto default, donde se encuentran las características con las que contara el buzón en el voicemail.conf

Para la comunicación entre centrales

exten => 53001,1,Dial(SIP/3001,20,Tt)

exten => 53001,2,Followme(3001,san)

exten => 53001,n,VoiceMail(4001@default)

La diferencia con el caso anterior, se establece debido a que la extensión va precedida por el prefijo configurado para la comunicación entre las centrales como se mencionó en apartado anteriores.

Extensiones iax

Para la comunicación entre centrales e internamente con el protocolo iax la configuración es similar como en el caso anterior, únicamente se debe cambiar el canal que permitirá establecer la comunicación, en este caso IAX2 para las extensiones definidas en iax.conf.

exten => 2001,1,Dial(IAX2/2001,20,Tt)

exten => 2001,n,VoiceMail(6001@default)

exten => 32001,1,Dial(IAX2/2001,20,Tt)

exten => 32001,n,VoiceMail(6001@default)

Dentro del contexto extensiones-internas se debe incluir los contextos de otras aplicaciones con las que se desea que cuenten las extensiones, para lo cual se determina de la siguiente manera:

include => Aplicaciones

include => call-center

include => parkedcalls

include => ivr

Una de las aplicaciones que se define dentro de *extensiones-internas* es *call-center*. En este caso se definirá en *extensions* la función de un call center con agentes estáticos como se explicó anteriormente.

[call-center]

exten => 33333,1,Answer()

exten => 33333,n,Ringing

exten => 33333,n,SetMusicOnHold(default)

exten => 33333,n,Queue(colas1)

exten => 33333,n,Hangup

La configuración inicia con un patrón numérico que dependerá del usuario, tomando en cuenta que el número con que inicia debe ser diferente al prefijo que se utiliza para la conexión entre centrales y hacia la PSTN, a continuación se debe colocar las aplicaciones que va a realizar de forma consecutiva con ayuda de la prioridad “n”. Inicialmente *Answer()* permite contestar la llamada, *Ringing* da un tono de marcado, *SetMusicOnHold* se escuchará la música en espera seleccionada en el contexto *default*, y la parte principal de call center es definir a la cola que se utilizara para la atención a la llamadas en *Queue*, finalmente con *Hangup* se cuelga la llamada.

En el caso que se desea realizar una configuración de agentes dinámicos, se puede realizar de la siguiente manera:

exten => 1000,1,Agentlogin(\${EXTEN})

exten => 1000,n,Hangup

Se define un patrón numérico que trabaja en conjunto con *AgentLogin* para que el agente se pueda logear con una extensión definida para la cola con agentes dinámicos,

exten => 3081,1,Answer

exten => 3081,n,Ringing

```
exten => 3081,n,SetMusicOnHold(default)
```

```
exten => 3081,n,Queue(colas)
```

```
exten => 3081,n,Hangup
```

La definición de la cola que atenderá las llamadas es igual que el caso anterior, únicamente dentro de la función Queue se debe colocar el contexto de la cola correspondiente configurado en el queues.conf.

Otro de los contexto configurados en el extension.conf es *Aplicaciones*, en este contexto se configurará un IVR que es un menú interactivo con el usuario, además se podrá realizar la grabación del menú, y la configuración de la extensión para poder llamar a consultar los mensajes dejados en el buzón de voz.

[Aplicaciones]

Acceso para escuchar los mensajes del Buzón de Voz

```
exten => 9500,1,Answer()
```

```
exten => 9500,n,VoiceMailMain(@default)
```

Cuenta con un patrón numérico seguido de la prioridad, y la aplicación *Answer()* para que la llamada sea contestada, a continuación con la aplicación *VoiceMailMain* se re direcciona al menú del voicemail asociado al contexto definido en voicemail.conf. El menú del voicemail es propio de Asterisk y nos permitirá escuchar mensajes nuevos y antiguos, borrar mensajes, etc,

Aplicación para grabar IVR

```
exten => 9991,1,Answer()
```

```
exten => 9991,n,Wait(1)
```

```
exten=> 9991,n,Record(bienvenida.gsm)
```



```

exten => 9991,n,Wait(1)
exten => 9991,n,Playback(bienvenida)
exten => 9991,n,Hangup()

```

Como se ha observado a lo largo de la configuración del archivo `extensions.conf` es importante contar con un patrón, en este caso numérico, seguido de las aplicaciones que se van a realizar con prioridad “n” para que sean consecutivas, la aplicación *Answer()* contesta la llamada, *Wait()* permite dar un tiempo de espera en segundos, *Record()* permite realizar la grabación asignándole un nombre (e.j. *bienvenida*) en formato gsm, *Playback()* reproduce la grabación para verificar si el mensaje grabado es satisfactorio para el usuario o no, y *Hangup* cuelga la llamada como ya se explicó en apartados anteriores.

Finalmente se configura el menú para que el usuario tenga la opción de dirigirse a una extensión o aplicación en específico, este menú contara con el número de opciones que la necesidad lo requiera.

```

[ivr]
exten => s,1,Answer()
exten => s,n,Wait(1)
exten => s,n,Background(bienvenida-menu)
exten => s,n,WaitExten(5)

```

En este caso se define una extensión “s”, como se mencionó en configuraciones anteriores permite iniciar el contexto, `ivr` cuenta con aplicaciones que se realizaran de forma consecutiva, *Background* reproduce la grabación de bienvenida, *WaitExten* tiempo de espera en la cual se ingresa la opción seleccionada.

```

exten => 1,1,Goto(extensiones-internas,7003,1)
exten => 2,1,Goto(extensiones-internas,2002,1)

```

```
exten => 3,1,Goto(Aplicaciones,9999,1)
exten => 4,1,Goto(call-center,33333,1)
exten => *,1,Goto(s,1)
```

Para configurar el menú se requiere de un patrón que será el que el usuario digite para acceder a la opción correspondiente, *Goto* permitirá dirigir al contexto deseado, sea extensión o una aplicación, “*” permite repetir el menú dirigiéndose a la opción 1 nuevamente.

```
exten => t,1,Playback(vm-goodbye)
exten => t,n,Hangup()
```

```
exten => i,1,Playback(pbx-invalid)
exten => i,n,Goto(s,1)
```

Playback reproduce audios, en este caso los sonidos son propios de Asterisk, en el primer caso reproduce un audio de despedida y el siguiente un audio de haber digitado una opción invalida, el parámetro “t” indica que no se ha seleccionado ninguna opción se despide y cuelga, “i” indica una opción invalida y regresa nuevamente al inicio del menú.

CAPÍTULO 4 : EVALUACIÓN DEL PROTOTIPO

Una vez que se ha puesto en funcionamiento la central, después de haber pasado por una selección de hardware y software para determinar las características apropiadas para el correcto funcionamiento de la misma, en este capítulo se realizó un análisis de los posibles problemas que se pueden presentar en cuanto a la calidad de Voz y que soluciones se pueden desarrollar para enfrentar estos problemas. Es decir se analizara lo que es QoS para VoIP.

4.1. Calidad de Servicio para VoIP (QoS).

Calidad de servicio o Quality of Service en inglés, es la capacidad de poder ofrecer un buen servicio al momento de transmitir voz y video, es decir su objetivo es garantizar la trasmisión de cierta información en un tiempo dado o denominado throughput.

En capítulos anteriores se mencionó las grandes ventajas de la telefonía IP, como el reaprovechamiento de los recursos, disminución en costos de implementación y ejecución, pero podría presentarse ciertos problemas debido a la utilización de redes como intranet, internet, etc, debido a que depende de la velocidad, ancho de banda de la red. Estos inconvenientes se podrán ir resolviendo con el paso del tiempo, avance de la tecnología y principalmente de las redes.

4.2. Problemas que se presentan con la calidad de voz

A lo largo del tiempo mientras avanzaba la evolución la telefonía IP, a la par se presentaron diversos problemas en la calidad de voz, los más comunes son: latencia, jitter, pérdida de paquetes, ancho de banda insuficiente, eco que serán analizados a continuación.

4.2.1. Jitter

El jitter es un parámetro que se encuentra presente en redes no orientadas a la conexión y que se basa en la conmutación de paquetes, de suma importancia cuando se habla de calidad de voz. Los paquetes en la comunicación viajan por rutas distintas para llegar al destino, en la red se puede encontrar con diferentes circunstancias las cuales hacen que el paquete llegue con retardo, lleguen en desorden o no lleguen.

Por lo tanto el jitter se define como la variación en el tiempo de la llegada de los paquetes causada por la congestión de la red, pérdida de sincronización o por las diferentes rutas que sigue el paquete hasta llegar al destino. En las comunicaciones en tiempo real, en este caso la transmisión de voz no se puede tener un jitter alto ya que esto provocaría pérdida de paquetes, es decir la llamada se escucharía entrecortada.

Una de las posibles soluciones es la creación de prioridades de colas (e.j.FIFO) en los dispositivos de la red (routers), además de enlaces con velocidades mayores y reserva de ancho de banda.

4.2.2. Latencia

Se define como el tiempo que le lleva a un paquete llegar desde su origen a su destino, este inconveniente no se presenta únicamente en la VoIP, sino es un problema general de todas las redes de telecomunicaciones.

Probablemente es uno de los menos sencillos de resolver, debido a que este retardo depende de los equipos que se manejen dentro de la red, en el caso de que el problema se encuentre dentro de una red interna, y se tenga el control de la misma, se podrá reservar un ancho de banda específico para el envío de tráfico en tiempo real.

4.2.3. Pérdida de paquetes

La pérdida de paquetes se presenta cuando un paquete no llega a su destino, las causas pueden ser diversas, por ejemplo, pérdidas en el medio de transmisión por cables mal ponchados, exceso de ruido en el ambiente, etc.

En una red de datos la pérdida de paquetes es bastante tolerante al contrario de una red por donde se transmite voz, ya que si hay una gran pérdida de paquetes durante una conversación telefónica la comunicación se hará muy difícil de entender. Una posible solución para evitar la pérdida de paquetes es no transmitir silencios para evitar congestionar la red con información no relevante.

4.2.4. Eco

Es un fenómeno que se produce por la conversión de 2 a 4 hilos en los sistemas telefónicos, generalmente provoca mayores molestias cuando el valor del retardo aumenta, especialmente en la red tradicional.

Para suprimir los problemas de eco se utiliza supresores de eco, permiten que el canal pase de full-duplex a half-duplex, evitando que la comunicación en un sentido retorne en sentido contrario, otra solución es utilizar un cancelador de eco, cumple la función un filtro, en este caso detecta en la señal de retorno información similar y procede a cancelar dicha información.

4.2.5. Ancho de banda insuficiente

El ancho de banda se define como la cantidad de información que puede viajar a través de una conexión de red, es decir está estrechamente relacionado con la capacidad de la red, en las comunicaciones de voz el ancho

de banda se puede encontrar limitado por diferentes aplicaciones como correo electrónico, descarga de archivos, etc.

En la VoIP el ancho de banda depende de los códecs que se utilicen para la comunicación, en este punto se puede comparar dos de los códecs más utilizados, el códec G.711 aproximadamente utiliza un ancho de banda de 80Kbps al contrario del códec G.729 que utiliza 24Kbps para establecer una sola comunicación.

Los protocolos que se utilice para la comunicación también provocan una variación en el ancho de banda de la comunicación debido a que un protocolo SIP utiliza mayor ancho de banda que un protocolo IAX.

4.2.6. Umbrales aceptables de la VoIP

Se define como los valores máximos de jitter, latencia y pérdida de paquetes para medir la calidad de la voz. En estos umbrales también nos permitirán obtener un valor del MOS (medida subjetiva) que se analizará posteriormente.

El jitter es el parámetro que más influye en la calidad de la voz y es el promedio de las diferencias de latencias, latencia es el retardo en el arribo de paquetes, la pérdida de paquetes es el porcentaje de paquetes perdidos por lo tanto tenemos los siguientes valores.

Tabla 4

Parámetros de QoS [24]

Calidad VoIP	Excelente	Bueno	Aceptable	Pobre
Jitter(ms)	$t < 10$	$10 \leq t < 20$	$20 \leq t < 50$	$t \geq 50$
Latencia(ms)	$t < 50$	$50 \leq t < 150$	$150 \leq t < 300$	$t \geq 300$
Perdida paquetes (%)	$p < 0,1$	$0,1 \leq p < 0,5$	$0,5 \leq p < 1,5$	$p \geq 1,5$

4.2.7. MOS (Mean Opinion Score)

Es la medida de la calidad de voz de manera subjetiva según la recomendación ITU-T P.800, y es determinado por una calificación que el usuario asigna entre un valor de 1-5 para indicar que tan aceptable estuvo la llamada.

Tabla 5

Calificación MOS [24]

MOS	CALIFICACIÓN
EXCELENTE	4,6 - 5
BUENO	4,1 - 4,5
ACEPTABLE	3,6 - 4
BAJO	3 - 3,5
INACEPTABLE	1 - 3

Por lo tanto la calificación del MOS de los usuarios dependerá de la calificación que se tenga de los umbrales de jitter, latencia y pérdida de paquetes, como se puede observar en la Tabla 6.

Tabla 6

Parámetros de QoS y MOS [24]

MOS	Jitter	Latencia	Perdida Paquetes
Excelente	Excelente	Excelente	Excelente
Bueno	Excelente	Bueno	Bueno
Bueno	Bueno	Excelente	Bueno
Bueno	Bueno	Bueno	Excelente
Aceptable	Bueno	Bueno	Bueno
Aceptable	Bueno	Bueno	Aceptable
Aceptable	Bueno	Aceptable	Bueno
Aceptable	Aceptable	Bueno	Bueno
Pobre	Aceptable	Bueno	Aceptable
Pobre	Aceptable	Aceptable	Bueno
Pobre	Bueno	Aceptable	Aceptable
Pobre	Aceptable	Aceptable	Aceptable
Pobre	Pobre	Aceptable	Aceptable
Pobre	Aceptable	Pobre	Aceptable
Pobre	Aceptable	Aceptable	Pobre
Malo	Pobre	Pobre	Aceptable
Malo	Pobre	Aceptable	Pobre
Malo	Aceptable	Pobre	Pobre
Inaceptable	Pobre	Pobre	Pobre

4.3. Llamadas simultáneas

Las llamadas simultáneas es uno de los parámetros importante para determinar qué tan eficiente es nuestro equipo en comparación con centrales que se ofrecen en el medio

4.3.1. Llamadas simultáneas considerando el ancho de banda.

El ancho de banda es punto importante de considerar debido a que no es recomendable transmitir voz en enlaces menores a 1Mbpsa menos que se adopten técnicas de QoS.

Es decir se ha determinado el número de llamadas simultáneas que soportará el escenario en cual se desarrolló este trabajo, como se ha

mencionado anteriormente se trabajó con un enlace de 10/100 Mbps debido a los elementos de red que se utilizaron, los cuales fueron detallados en capítulos anteriores, principalmente por los routers Mikrotik.

Por lo tanto:

Uno de los principales parámetros a considerar para realizar es el tipo de Codec que se va a utilizar, al encontrarnos en Ecuador y lo que Latinoamérica se utiliza el códec G.711 cómo se describió en capítulos anteriores.

Codec Information				Bandwidth Calculations					
Codec & Bit Rate (Kbps)	Codec Sample Size (Bytes)	Codec Sample Interval (ms)	Mean Opinion Score (MOS)	Voice Payload Size (Bytes)	Voice Payload Size (ms)	Packets Per Second (PPS)	Bandwidth MP or FRF.12 (Kbps)	Bandwidth w/cRTP MP or FRF.12 (Kbps)	Bandwidth Ethernet (Kbps)
G.711 (64 Kbps)	80 Bytes	10 ms	4.1	160 Bytes	20 ms	50	82.8 Kbps	67.6 Kbps	87.2 Kbps
G.729 (8 Kbps)	10 Bytes	10 ms	3.92	20 Bytes	20 ms	50	26.8 Kbps	11.6 Kbps	31.2 Kbps
G.723.1 (6.3 Kbps)	24 Bytes	30 ms	3.9	24 Bytes	30 ms	33.3	18.9 Kbps	8.8 Kbps	21.9 Kbps
G.723.1 (5.3 Kbps)	20 Bytes	30 ms	3.8	20 Bytes	30 ms	33.3	17.9 Kbps	7.7 Kbps	20.8 Kbps
G.726 (32 Kbps)	20 Bytes	5 ms	3.85	80 Bytes	20 ms	50	50.8 Kbps	35.6 Kbps	55.2 Kbps
G.726 (24 Kbps)	15 Bytes	5 ms		60 Bytes	20 ms	50	42.8 Kbps	27.6 Kbps	47.2 Kbps
G.728 (16 Kbps)	10 Bytes	5 ms	3.61	60 Bytes	30 ms	33.3	28.5 Kbps	18.4 Kbps	31.5 Kbps
G722_64k(64 Kbps)	80 Bytes	10 ms	4.13	160 Bytes	20 ms	50	82.8 Kbps	67.6Kbps	87.2 Kbps
ilbc_mode_20(15.2Kbps)	38 Bytes	20 ms	NA	38 Bytes	20 ms	50	34.0Kbps	18.8 Kbps	38.4Kbps
ilbc_mode_30(13.33Kbps)	50 Bytes	30 ms	NA	50 Bytes	30 ms	33.3	25.867 Kbps	15.73Kbps	28.8 Kbps

Figura 19. Codec de Audio[25]

Primero se debe calcular el tamaño de la trama.

Tamaño de trama= Payload + Capa 2 + Capa 3 + Capa 4 + encabezado de capa de enlace

Tamaño de trama= 160+40+60 = 206 bytes

El payload depende del códec y podemos observar en la figura 19.

Sabemos que la trama del códec es de 8 bits por lo tanto:

$$206 \text{ bytes} = \frac{8 \text{ bis}}{1 \text{ byte}} = 1648 \text{ bits/trama}$$

Codec g 711 tiene 50 tramas x Segundo (figura 19.) por lo tanto:

$$BW \text{ requerido} = 1648 \frac{\text{bits}}{\text{trama}} \times \frac{50 \text{ tramas}}{\text{seg}} = 82,400 \frac{\text{bits}}{\text{seg}} \approx 80 \text{ kbps}$$

Tomando en cuenta las características del router 10/100 Mbps

$$\text{Número máximo de llamadas simultáneas} = \frac{100 \text{ Mbps}}{80 \text{ Kbps}} = 1250$$

Suponiendo que usamos toda la capacidad de nuestro equipo router y que nuestro ordenador, donde se encuentra el servidor ASTERISK tiene un alto rendimiento se podrán realizar 1250 llamadas simultáneas, pero, ¿El rendimiento de la Raspberry soportará dicha cantidad de llamadas, al ser un miniordenador?, lo más probable es que no, por lo que se tiene que llevar a cabo un último y muy importante análisis de rendimiento de nuestra Raspberry.

4.3.2. Llamadas simultáneas considerando el rendimiento de la Raspberry PI

El número de llamada simultaneas que el equipo soporte no determina el números de usuarios que la central tenga, por ejemplo la central podrá contar con 50 usuarios y soportar 10 llamadas simultaneas. Estos valores son los que se determinara con ayuda del software denominado SIPP que permite enviar tráfico SIP y permite llevar el equipo al estrés para encontrar estos valores.

La diferencia entre llamadas simultáneas y usuarios que adquiere la central nos permite determinar a qué tipo de soluciones se puede presentar el prototipo.

A continuación se presenta una tabla con los requerimientos mínimos recomendados y la cantidad de llamadas simultáneas soportadas, y se aclara que los valores son muy conservadores y que no son definitivos [26].

Tabla 7

Requerimientos mínimos recomendados – Asterisk[26]

Propósito	Número de canales	Mínimo recomendado
Sistema como hobby	No más de 5	400MHz x86, 256 MB RAM
SOHO(pequeña oficina/oficina en casa)	5 a 10	1GHz x86, 512 MB RAM
Sistema para una pequeña empresa	Hasta 25	3GHz x86, 1 GB RAM
Sistema mediano/grande	Más de 25	Dual CPUs o múltiples servidores

Por lo tanto realizando una comparación con el servidor que se montó sobre *Raspberry PI* podemos observar que todos los procesadores son x86 y la *raspberry* cuenta con un ARM, con estructura RISC que permite el paralelismo en la ejecución de instrucciones y tareas de menor tamaño, con procesos cortos y la disminución de la energía, por lo tanto teóricamente con

ayuda de la tabla 7 podemos decir que la *Raspberry Pi* es recomendada para escenarios SOHO.

Para determinar en la práctica el número de llamadas simultáneas se utilizara el software SIPP

4.3.2.1. SIPP

SIPP es un generador de tráfico SIP que trabaja con dos tipos de usuarios que son: UAC y UAS, hacen referencia a, desde donde se está generando el tráfico, en este proyecto se trabajó con UAC, es decir se genera el tráfico desde un dispositivo cliente, otra posibilidad de generar este tráfico en utilizando archivos .xml donde se podrá variar las especificaciones de cómo se desea que se presente el escenario de simulación. La instalación de este software se muestra en los anexos.

Para generar el tráfico es necesario crear una extensión en el sip.conf como se muestra a continuación:

```
[sipp]
type=friend
context=sipp
host=dynamic
user=sip
insecure = invite, port
canreinvite=no
disallow=all
allow=ulaw ; el codec puede ser variado entre los diferentes que existen
para realizar un análisis.
```

Mientras que en extensions.conf:

[sipp]

exten => 1234,1,Answer

exten => 1234,n,SetMusicOnHold(default)

exten => 1234,n,WaitMusicOnHold(30)

exten => 1234,n, Hangup

Como podemos observar se genera un patrón numérico al cual el software llamara, se reproducirá música con un tiempo de 30 segundos y se colgara la llamada.

Entonces el tráfico se generara de la siguiente manera, primero en el terminal se debe ingresar a la carpeta donde se encuentra los archivos de sipp, una vez ahí se ingresa los siguientes comandos:

```
$ sudo ./sipp -s 1234 -sn uac -m 10 -d 5000 -r10 -rp 1000 -i "ip cliente" "ip servidor"
```

Donde:

sn: es el número al cual se va a llamar y la extensión generada en el sip.conf

m: es el número de llamadas que se va a generar

d: es el espacio de tiempo de pause en milisegundos

r: es el número de intentos por llamada que se realizará en 1 segundo

i: por defecto es la dirección IP de nuestro cliente seguido de la IP de nuestro servidor ASTERISK.

Por lo tanto se obtuvieron los siguientes resultados:

Tabla 8

Llamadas simultáneas y Rendimiento de la Raspberry Pi

Número de llamadas simultáneas	% Rendimiento ASTERISK				% FINAL RENDIMIENTO ASTERISK
5	25	24	22	26	24.25
10	55	50	53	51	52.25
15	60	55	47	65	56.75
20	79	72	75	72	74.5
25	88	89	85	84	86.5
27	95	94	94	95	94.5
30	100	100	100	100	100
35	100	100	100	100	100

Observando la Tabla 8 se tiene que el rendimiento de la tarjeta llega al 100% con 35 llamadas simultáneas.

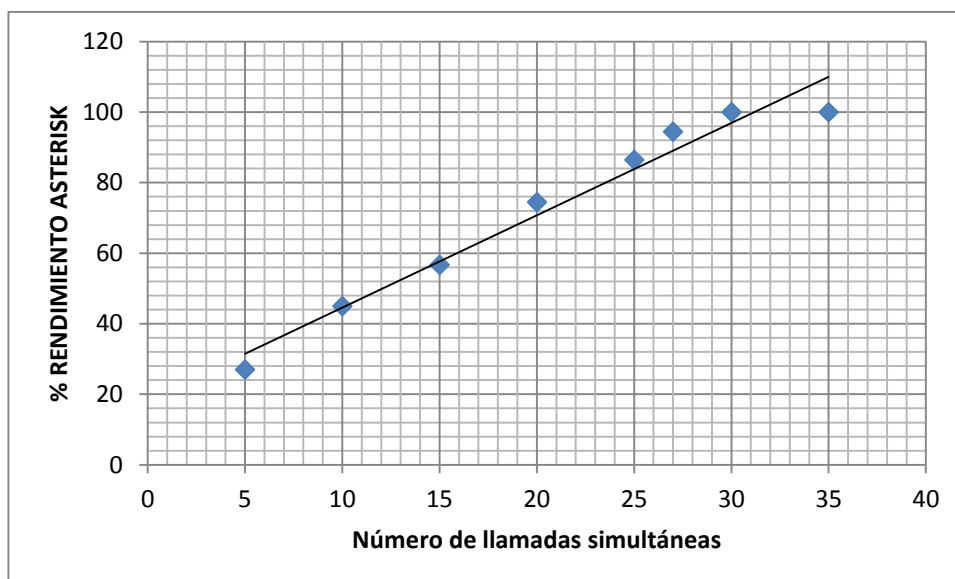


Figura 20. Llamadas simultáneas vs Rendimiento

Partiendo de que la tarjeta llega a su rendimiento máximo con 35 llamadas simultáneas, si se desea aumentar el número de llamadas, estas se empieza a perder como se observa en la tabla 9.

Tabla 9

Llamadas Pérdidas

Número de llamadas	Resultados
35	<pre> -----+-----+----- Incoming call created 0 0 OutGoing call created 0 35 Total Call created 0 35 Current Call 0 -----+-----+----- Successful call 0 35 Failed call 0 0 -----+-----+----- </pre>
36	<pre> Incoming call created 0 0 OutGoing call created 0 36 Total Call created 0 36 Current Call 0 -----+-----+----- Successful call 0 35 Failed call 0 1 -----+-----+----- </pre>
37	<pre> Incoming call created 0 0 OutGoing call created 0 37 Total Call created 0 37 Current Call 0 -----+-----+----- Successful call 0 36 Failed call 0 1 -----+-----+----- </pre>
40	<pre> -----+-----+----- Successful call 0 36 Failed call 0 4 -----+-----+----- </pre>

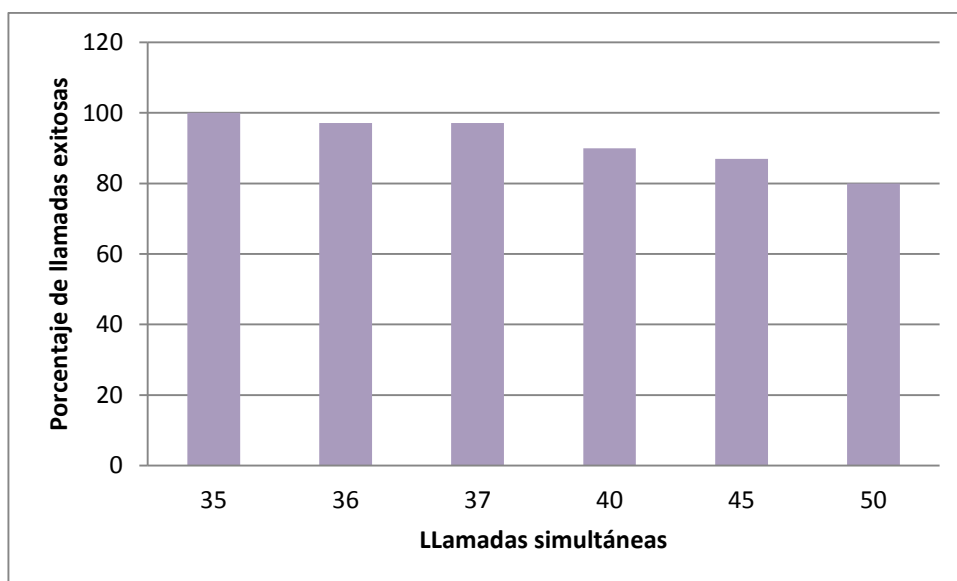


Figura 21. Porcentaje de Llamadas Pérdidas

Entonces:

Con las pruebas realizadas se puede decir que el servidor Asterisk implementado sobre la *Raspberry Pi* soportará alrededor de 35 llamadas simultáneas.

El número de usuarios con los que contara la central dependerá del servicio que se vaya a prestar en el lugar implementado, ya que si se requiere realizar más de 35 llamadas simultáneas y a la vez utilizando este mismo servidor para otras aplicaciones, se sobre saturará el rendimiento del minicomputador lo que provocará disminución en el número de llamadas simultáneas.

4.4. Análisis de QoS

Para analizar los parámetros antes desarrollados y partiendo de cómo se pueden presentar estos problemas y como solucionarlos. Se realizó tres tipos de pruebas, la primera se trata de realizar una llamada de sip-sip, la segunda se analizará con las 10 llamadas simultáneas y la tercera con 35 llamadas simultáneas que soporta la Raspberry PI. Para estas pruebas se utilizó el software wireshark

4.4.1. SIP-SIP

Para realizar esta pruebas se configuro dos extensiones una con ayuda del sofphone en el computador donde se va a correr wireshark y otra extensión en un diferente equipo.

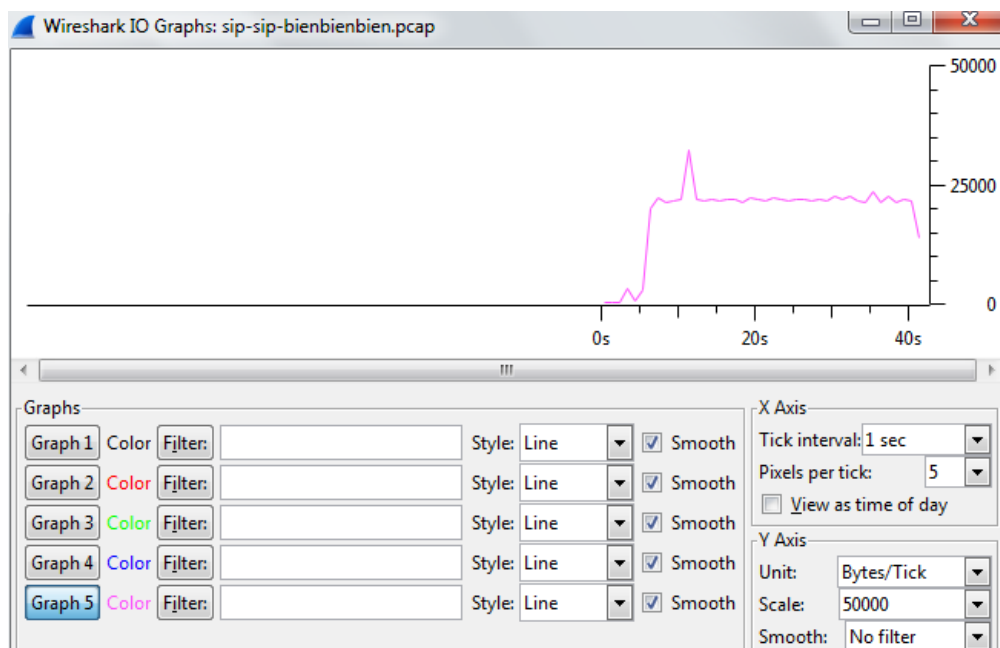


Figura 22. Tráfico TCP, RTP con una llamadas de SIP-SIP

En la figura 22 podemos observar todo el tráfico que cursa por la central tanto TCP como RTP, representada en bytes/segundo.

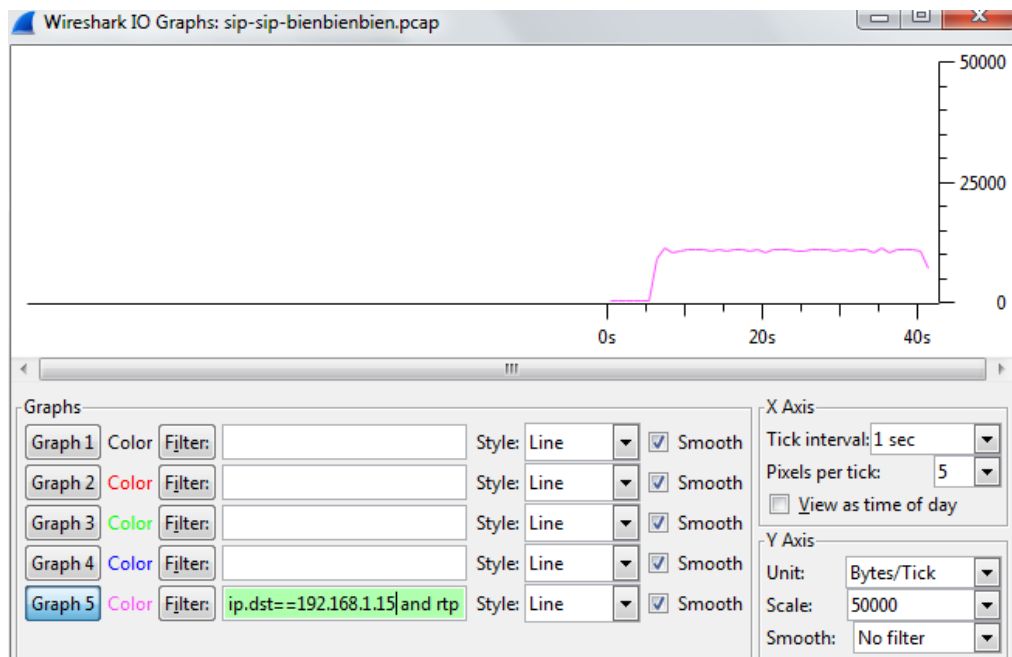


Figura 23. Tráfico RTP cursado por la Central con una llamadas de SIP-SIP

En la figura 23 podemos observar únicamente el tráfico de voz que cursa por la central, es decir los paquetes RTP que contiene el audio, representada en bytes/segundo.

Ahora vamos a observar el jitter y paquetes perdidos en la llamada realizada.

Detected 2 RTP streams. Choose one for forward and reverse direction for analysis									
Src addr	Src port	Dst addr	Dst port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)
192.168.1.15	8000	192.168.1.16	11596	0x7E3DB70	g711U	1786	0 (0.0%)	32.121000	5.450491
192.168.1.16	11596	192.168.1.15	8000	0x5E86660E	g711U	1774	0 (0.0%)	87.300000	25.804682

Figura 24. Llamada de SIP-SIP

En este caso se va analizar la trama hacia donde se realizó la llamada, por lo tanto podemos observar que no se tiene paquetes perdidos con jitter de 5,45 ms.

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
171	1693	19.99	2.07	0.24	81.60		[Ok]
173	1694	30.05	2.57	-9.81	81.60		[Ok]
174	1695	10.01	3.03	0.19	81.60		[Ok]
177	1696	21.11	2.91	-0.92	81.60		[Ok]
178	1697	20.00	2.73	-0.92	81.60		[Ok]
179	1698	20.02	2.56	-0.94	81.60		[Ok]
183	1699	30.02	3.02	-10.95	80.00		[Ok]
184	1700	9.97	3.46	-0.92	80.00		[Ok]

Analysing stream from 192.168.1.15 port 8000 to 192.168.1.16 port 11596 SSRC = 0x7E3DB70

Max delta = 32.12 ms at packet no. 2960
 Max jitter = 5.45 ms. Mean jitter = 2.98 ms.
 Max skew = -11.99 ms.
 Total RTP packets = 1786 (expected 1786) Lost RTP packets = 0 (0.00%) Sequence errors = 0
 Duration 35.70 s (-27 ms clock drift, corresponding to 7994 Hz (-0.08%))

Figura 25. Trama de la llamada de SIP-SIP

En la llamada analizada se tiene un jitter de 2,98 ms, 0% de paquetes perdidos y un ancho de banda de 80kbps. Por lo tanto el jitter es menor a 10 ms, y no hay paquetes perdidos, es decir en una llamada se tiene una calidad de VoIP excelente con un MOS excelente.

4.4.2. 10 Llamadas simultáneas

Las 10 llamadas simultáneas fueron realizadas con el software sipp para poder analizar los parámetros de QoS.

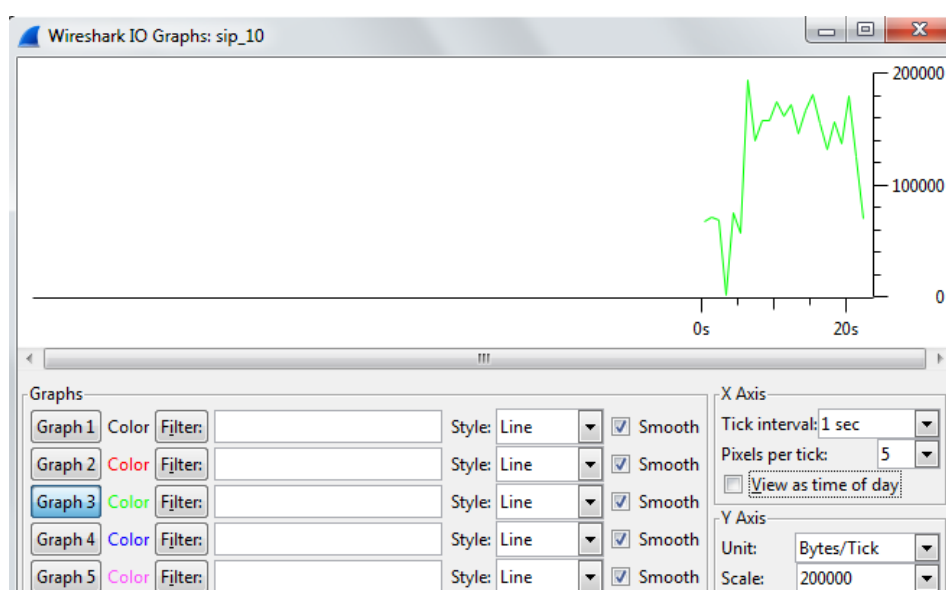


Figura 26. Tráfico TCP, RTP con 10 llamadas simultáneas

En la figura 26 podemos observar todo el tráfico que cursa por la central tanto TCP como RTP, representada en bytes/segundo.

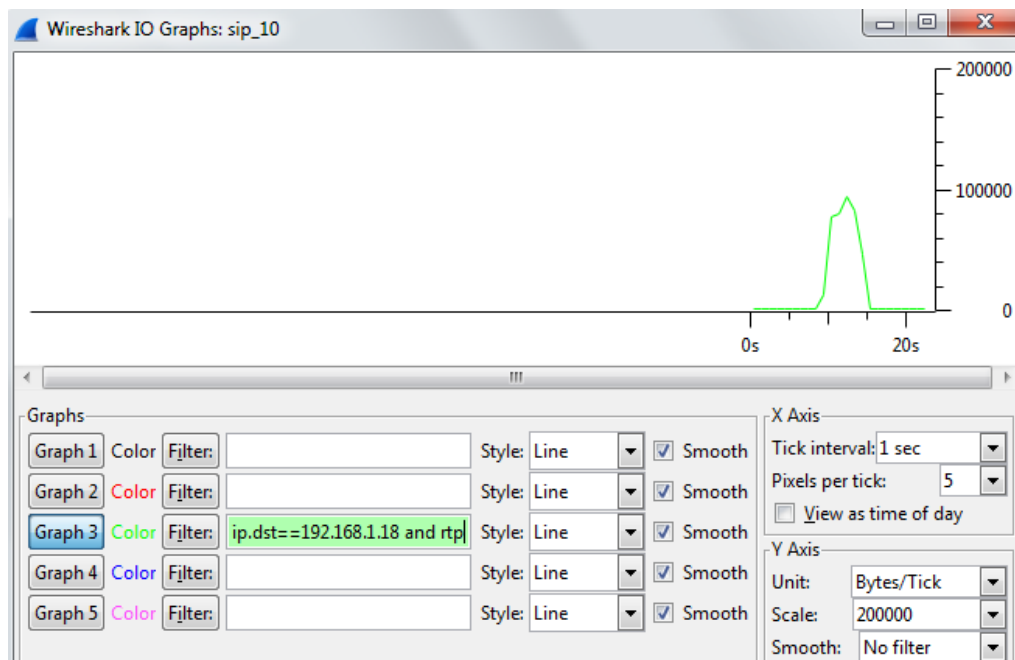


Figura 27. Tráfico RTP con 10 llamadas simultáneas

En la figura 27 podemos observar únicamente el tráfico de voz que cursa por la central, es decir los paquetes RTP que contiene el audio, representada en bytes/segundo.

Ahora vamos a observar el jitter y paquetes perdidos en cada una de las llamadas realizadas:

Wireshark: RTP Streams										
Detected 10 RTP streams. Choose one for forward and reverse direction for analysis										
Src addr	Src port	Dst addr	Dst port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)
192.168.1.16	30370	192.168.1.18	6000	0x38F39F9D	g711U	188	0 (0.0%)	64.499000	11.644385	7.132096
192.168.1.16	6850	192.168.1.18	6000	0x7D7588B4	g711U	187	3 (1.6%)	83.620000	12.634401	7.514717
192.168.1.16	18406	192.168.1.18	6000	0x5F0378DA	g711U	186	1 (0.5%)	62.700000	12.793879	7.294070
192.168.1.16	15416	192.168.1.18	6000	0x28F20E60	g711U	180	1 (0.6%)	69.530000	12.258082	7.938200
192.168.1.16	30108	192.168.1.18	6000	0x6E8942AE	g711U	181	0 (0.0%)	81.252000	14.483078	8.475182
192.168.1.16	11506	192.168.1.18	6000	0x491AD2FC	g711U	175	3 (1.7%)	73.796000	12.090462	8.745406
192.168.1.16	12692	192.168.1.18	6000	0x5042A33B	g711U	181	1 (0.5%)	62.604000	12.452550	8.013442
192.168.1.16	26970	192.168.1.18	6000	0x561740B0	g711U	179	0 (0.0%)	55.091000	11.171777	7.991028
192.168.1.16	13904	192.168.1.18	6000	0x3C68D668	g711U	180	0 (0.0%)	84.054000	13.510717	8.423911
192.168.1.16	19442	192.168.1.18	6000	0x52DE6F8E	g711U	185	0 (0.0%)	71.479000	11.706944	7.169805

Figura 28. 10 Llamadas realizadas

Como podemos observar el porcentaje de paquetes perdidos es menor al 1% a excepción de la segundo y sexta llamadas donde son mayor al 1,5%, en cuanto al jitter tenemos que se encuentra entre 7 y 9 ms.

Si analizamos una llamada con 0% de paquetes perdidos tenemos:

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
2217	2544	0.75	8.75	-477.36	62.40		[Ok]
2235	2545	34.94	9.14	-492.30	62.40		[Ok]
2245	2546	19.13	8.62	-491.43	62.40		[Ok]
2255	2547	20.08	8.09	-491.51	62.40		[Ok]
2265	2548	18.52	7.67	-490.04	64.00		[Ok]
2274	2549	22.33	7.34	-492.36	64.00		[Ok]
2285	2550	20.13	6.89	-492.49	65.60		[Ok]
2295	2551	18.35	6.56	-490.84	65.60		[Ok]

Analysing stream from 192.168.1.16 port 30370 to 192.168.1.18 port 6000 SSRC = 0x38F39F9D

Max delta = 64.50 ms at packet no. 2038
 Max jitter = 11.64 ms. Mean jitter = 7.13 ms.
 Max skew = -729.30 ms.
 Total RTP packets = 188 (expected 188) Lost RTP packets = 0 (0.00%) Sequence errors = 0
 Duration 4.47 s (-1674 ms clock drift, corresponding to 5004 Hz (-37.45%))

Figura 29. Trama de una llamada con 0% de paquetes perdidos

Si observamos la figura 29 podemos determinar que una de las llamadas tiene un jitter de 7,13ms, no tiene paquetes perdidos y utiliza un ancho de banda de 60 kbps.

Ahora observaremos una llamada que tiene más de 1% de paquetes perdidos.

Forward Direction		Reversed Direction					
Analysing stream from 192.168.1.16 port 11506 to 192.168.1.18 port 6000 SSRC = 0x491AD2FC							
Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
2618	54141	18.78	8.29	-621.24	72.00		[Ok]
2627	54142	17.58	7.93	-618.82	72.00		[Ok]
2659	54144	61.64	8.78	-640.46	70.40		Wrong sequence nr.
2667	54145	27.78	8.72	-648.24	70.40		[Ok]
2677	54146	11.35	8.72	-639.59	70.40		[Ok]
2687	54147	31.63	8.90	-651.22	70.40		[Ok]
2698	54148	30.55	9.00	-661.77	68.80		[Ok]
2713	54149	34.13	9.32	-675.90	68.80		[Ok]
----	-----	----	----	-----	----		----
Max delta = 73.80 ms at packet no. 1564 Max jitter = 12.09 ms. Mean jitter = 8.75 ms. Max skew = -895.00 ms. Total RTP packets = 178 (expected 178) Lost RTP packets = 3 (1.69%) Sequence errors = 3 Duration 4.43 s (-1911 ms clock drift, corresponding to 4553 Hz (-43.08%))							

Figura 30. Trama de una llamada con más del 1% de paquetes perdidos

Por lo tanto en la figura 30 se tiene una jitter de 8,75ms, con un ancho de banda de 70kbps y en color rosado uno de los paquetes perdidos.

En conclusión tenemos:

Tabla 10

Análisis de QoS con 10 llamadas simultáneas

	Paquetes Perdidos	Jitter	Ancho de Banda
1ra llamada	0%	7,13 ms	60 kbps
6ta llamada	1,7%	8,75 ms	70 kbps

Con estos datos y la ayuda de las tablas 4 y 6 podemos decir que en cuanto al jitter en los dos casos es menor a 10 ms por lo tanto con este parámetro podemos decir que la calidad de VoIP es excelente, en cuanto a los paquetes perdidos en el segundo caso podemos decir que tenemos una calidad de VoIP pobre ya que se tiene una pérdida de paquetes mayor al 1,5%. En general si analizamos los valores de la figura 28 podemos decir que la calidad de VoIP en este sistema con 10 llamadas simultáneas es excelente, por lo tanto se tendría una calificación de MOS excelente.

4.4.3. 35 Llamadas simultáneas

Las 35 llamadas simultáneas fueron realizadas con el software sipp para poder analizar los parámetros de QoS. En este caso a la vez de saturar el sistema también se estaba realizando la captura de paquetes con wireshark, por lo cual el rendimiento de la Raspberry PI llego al máximo y se perdieron 14 llamadas, es decir únicamente se realizaron 21 llamadas.

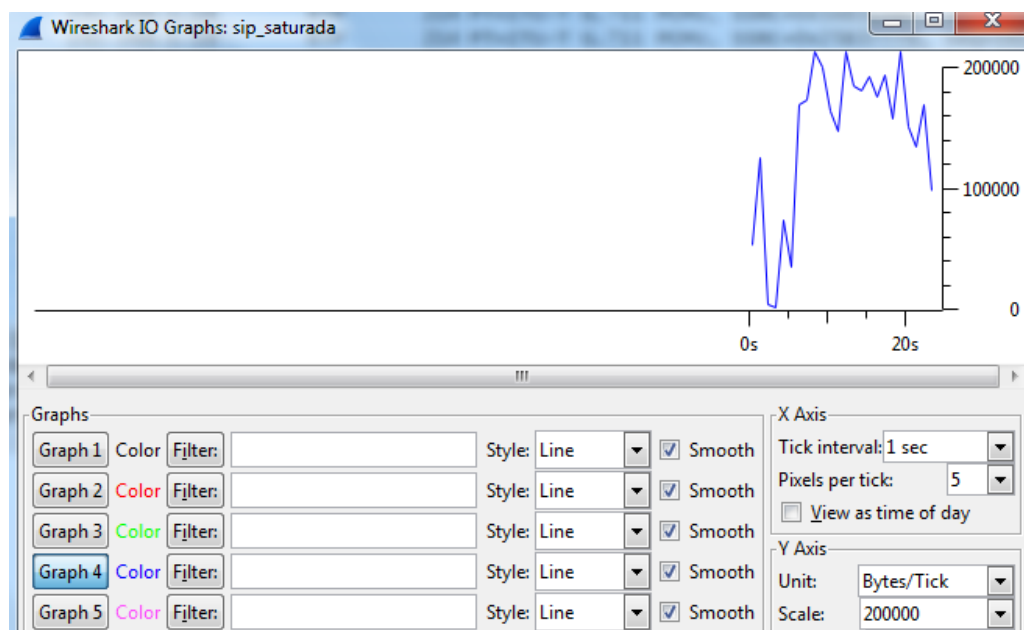


Figura 31. Tráfico TCP, RTP con 35 llamadas simultáneas

En la figura 31 podemos observar todo el tráfico que cursa por la central tanto TCP como RTP, representada en bytes/segundo.

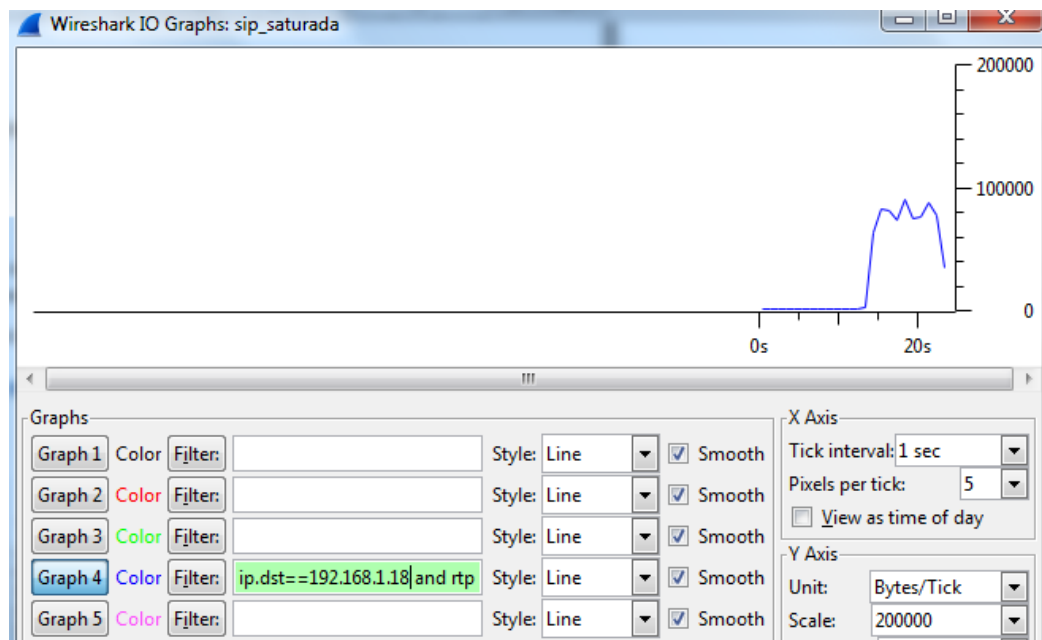
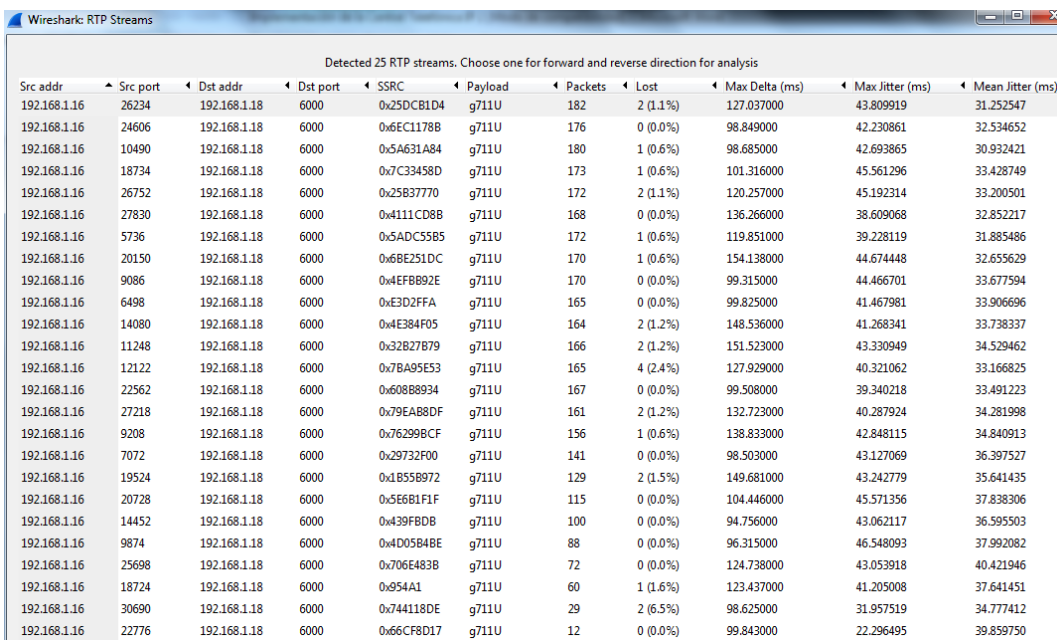


Figura 32. Tráfico RTP con 35 llamadas simultáneas

En la figura 32 podemos observar únicamente el tráfico de voz que cursa por la central, es decir los paquetes RTP que contiene el audio, representada en bytes/segundo.

Ahora vamos a observar el jitter y paquetes perdidos en cada una de las llamadas realizadas, en este caso las llamadas que se realizaron satisfactoriamente:

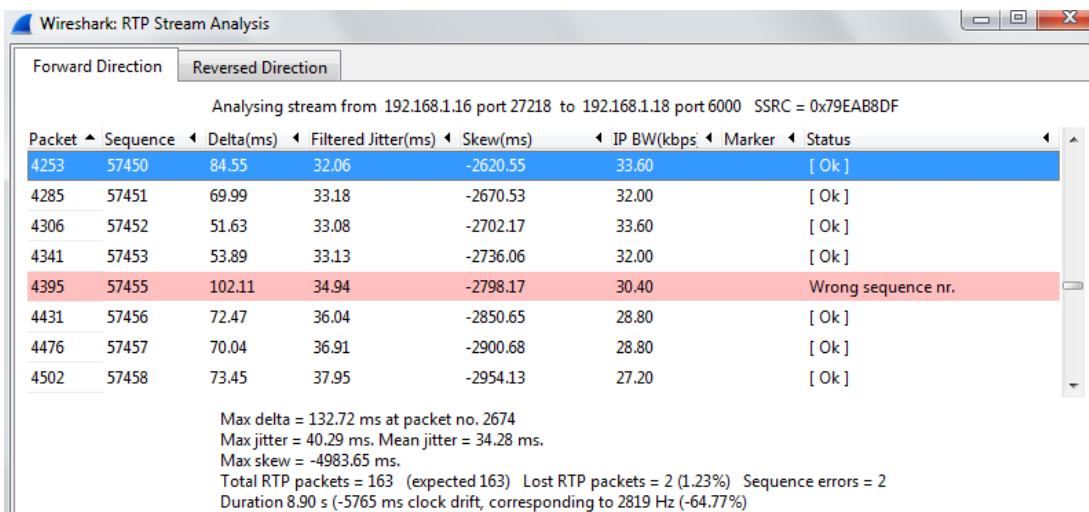


Src addr	Src port	Dst addr	Dst port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)
192.168.1.16	26234	192.168.1.18	6000	0x25DCB1D4	g711U	182	2 (1.1%)	127.037000	43.809919	31.252547
192.168.1.16	24606	192.168.1.18	6000	0x6EC11788	g711U	176	0 (0.0%)	98.849000	42.230861	32.534652
192.168.1.16	10490	192.168.1.18	6000	0x5A631A84	g711U	180	1 (0.6%)	98.685000	42.693865	30.932421
192.168.1.16	18734	192.168.1.18	6000	0x7C33458D	g711U	173	1 (0.6%)	101.316000	45.561296	33.428749
192.168.1.16	26752	192.168.1.18	6000	0x25B37770	g711U	172	2 (1.1%)	120.257000	45.192314	33.200501
192.168.1.16	27830	192.168.1.18	6000	0x4111CD8B	g711U	168	0 (0.0%)	136.266000	38.609068	32.852217
192.168.1.16	5736	192.168.1.18	6000	0x5ADC55B5	g711U	172	1 (0.6%)	119.851000	39.228119	31.885486
192.168.1.16	20150	192.168.1.18	6000	0x6BE251DC	g711U	170	1 (0.6%)	154.138000	44.674448	32.655629
192.168.1.16	9086	192.168.1.18	6000	0x4EFB892E	g711U	170	0 (0.0%)	99.315000	44.466701	33.677594
192.168.1.16	6498	192.168.1.18	6000	0xE3D2FFA	g711U	165	0 (0.0%)	99.825000	41.467981	33.906696
192.168.1.16	14080	192.168.1.18	6000	0x4E384F05	g711U	164	2 (1.2%)	148.536000	41.268341	33.738337
192.168.1.16	11248	192.168.1.18	6000	0x32B27B79	g711U	166	2 (1.2%)	151.523000	43.330949	34.529462
192.168.1.16	12122	192.168.1.18	6000	0x7BA95E53	g711U	165	4 (2.4%)	127.929000	40.321062	33.166825
192.168.1.16	22562	192.168.1.18	6000	0x608B8934	g711U	167	0 (0.0%)	99.508000	39.340218	33.491223
192.168.1.16	27218	192.168.1.18	6000	0x79EAB8DF	g711U	161	2 (1.2%)	132.723000	40.287924	34.281998
192.168.1.16	9208	192.168.1.18	6000	0x76299BCF	g711U	156	1 (0.6%)	138.833000	42.848115	34.840913
192.168.1.16	7072	192.168.1.18	6000	0x29732F00	g711U	141	0 (0.0%)	98.503000	43.127069	36.397527
192.168.1.16	19524	192.168.1.18	6000	0x1B55B972	g711U	129	2 (1.5%)	149.681000	43.242779	35.641435
192.168.1.16	20728	192.168.1.18	6000	0x5E6B1F1F	g711U	115	0 (0.0%)	104.446000	45.571356	37.838306
192.168.1.16	14452	192.168.1.18	6000	0x439FBDB	g711U	100	0 (0.0%)	94.756000	43.062117	36.595503
192.168.1.16	9874	192.168.1.18	6000	0x4D05B4BE	g711U	88	0 (0.0%)	96.315000	46.548093	37.992082
192.168.1.16	25698	192.168.1.18	6000	0x706E483B	g711U	72	0 (0.0%)	124.738000	43.053918	40.421946
192.168.1.16	18724	192.168.1.18	6000	0x954A1	g711U	60	1 (1.6%)	123.437000	41.205008	37.641451
192.168.1.16	30690	192.168.1.18	6000	0x744118DE	g711U	29	2 (6.5%)	98.625000	31.957519	34.777412
192.168.1.16	22776	192.168.1.18	6000	0x66CF8D17	g711U	12	0 (0.0%)	99.843000	22.296495	39.859750

Figura 33. 21 Llamadas realizadas

En la figura 33 podemos observar que el porcentaje de paquetes perdidos varía entre valores mayores y menores al 1,5% en cuanto al jitter tenemos que se encuentra entre 30 y 40 ms.

Si analizamos una llamada con paquetes perdidos menor al 1,5% tenemos:



Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
4253	57450	84.55	32.06	-2620.55	33.60	[Ok]	
4285	57451	69.99	33.18	-2670.53	32.00	[Ok]	
4306	57452	51.63	33.08	-2702.17	33.60	[Ok]	
4341	57453	53.89	33.13	-2736.06	32.00	[Ok]	
4395	57455	102.11	34.94	-2798.17	30.40	Wrong sequence nr.	
4431	57456	72.47	36.04	-2850.65	28.80	[Ok]	
4476	57457	70.04	36.91	-2900.68	28.80	[Ok]	
4502	57458	73.45	37.95	-2954.13	27.20	[Ok]	

Max delta = 132.72 ms at packet no. 2674
 Max jitter = 40.29 ms. Mean jitter = 34.28 ms.
 Max skew = -4983.65 ms.
 Total RTP packets = 163 (expected 163) Lost RTP packets = 2 (1.23%) Sequence errors = 2
 Duration 8.90 s (-5765 ms clock drift, corresponding to 2819 Hz (-64.77%))

Figura 34. Trama de una llamada con menos del 1,5% de paquetes perdidos

Si analizamos una de las llamadas, tenemos que tiene un jitter de 34,28 ms, con un porcentaje de 1,23% paquetes perdidos y utiliza un ancho de banda de 30 kbps.

Ahora se analizará una llamada que tiene más de 1,5% de paquetes perdidos.

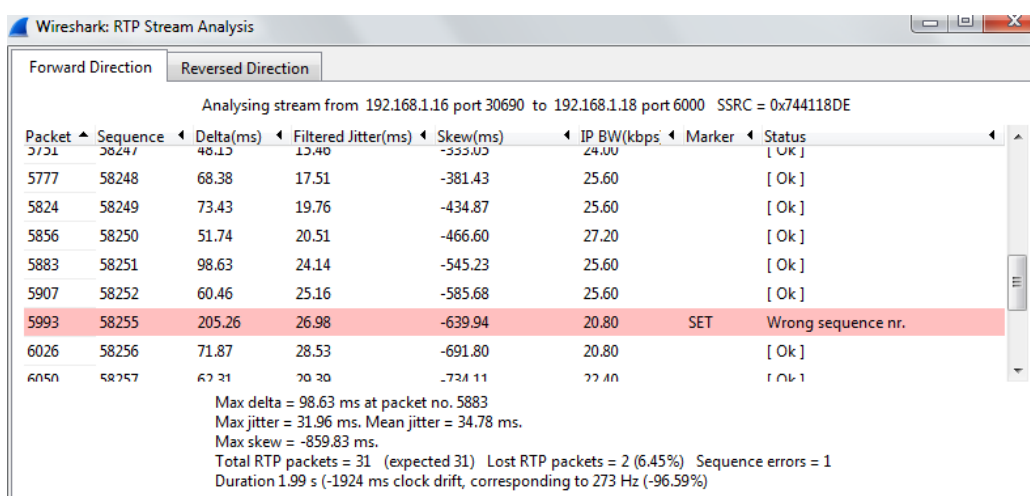


Figura 35. Trama de una llamada con más del 1,5% de paquetes perdidos

En la figura 35 se tiene una jitter de 34,78ms, con un ancho de banda de 25kbps, 6,45%de paquetes perdidos..

En conclusión tenemos:

Tabla 11

Análisis de QoS con 35 llamadas simultáneas

	Paquetes Perdidos	Jitter	Ancho de Banda
1ra llamada	1,2%	34,28 ms	30 kbps
20va llamada	6,45%	34,78 ms	25 kbps

Con estos datos y la ayuda de las tablas 4 y 6 podemos decir que en cuanto al jitter en los dos casos es mayor a 20 ms y menor a 50 ms por lo tanto con este parámetro podemos decir que la calidad de VoIP es aceptable, en cuanto a los paquetes perdidos en el segundo caso podemos decir que tenemos un calidad de VoIP pobre ya que se tiene una pérdida de paquetes mayor al 1,5%. En general si analizamos los valores de la figura 33 podemos decir que la calidad de VoIP en este sistema con 21 llamadas simultáneas ya que se perdieron 14 llamadas es aceptable, por lo tanto se tendría una calificación de MOS Bueno.

Para finalizar se va a realizar un resumen de los casos anteriormente analizados:

Tabla 12

Análisis de QoS con llamadas simultáneas

Llamadas simultáneas	Jitter	Paquetes Perdidos	Ancho de Banda	MOS
1	2,98 ms	0%	80kbps	excelente
10	8,75 ms	1,7%	70kbps	excelente
35	34,78 ms	6,45%	25kbps	bueno

4.5. Análisis de Costos

Una vez que se culminado con la implementación, evaluación del prototipo, podemos realizar un análisis comparativo de costos bajo dos aspectos:

- Implementación de la central en comparación con centrales IP y tradicionales que se encuentran en el mercado.
- Ahorro mensual del usuario de la central en llamadas realizadas.

4.5.1. Costos implementación central

Para realizar este análisis en cuanto a centrales IP se consultó el costo de varios tipos de centrales IP y se obtuvieron los siguientes costos:

Tabla 13

Análisis de Costos de Centrales IP

Tipo de Central	Número de extensiones	Llamadas simultáneas	Precio
Central Asterisk	-	-	\$ 300
Grandstream	500	30	\$ 458
Central Peq. IPo4	-	-	\$ 388
Grandstream	200	30	\$ 380
Panasonic	24	-	\$ 385
Panasonic	16	-	\$ 300
3com 3cr10551a	-	25	\$ 1395
RASPBERRY PI	-	35	\$ 160

Como podemos observar el costo de las centrales para un promedio de 100 extensiones esta alrededor de \$ 300 dependiendo de la marca del equipo. Ahora se analizará el costo de implementar una central con Raspberry Pi.

Tabla 14

Análisis de Costos de Raspberry como Central

		Precio
Raspberry Pi Model B	Alrededor de 100 líneas	\$ 80
Gateway Grandstream	1FXO, 1FXS	\$ 80

Si sumamos el costo de los dos equipos, el precio es de \$160 dólares, con una central de este tipo se podría utilizar para pequeñas oficinas o incluso en una casa. Además durante el desarrollo de este proyecto se pudo analizar que la central puede pasar perfectamente conectada todo un día sin que sufra recalentamientos o que el sistema tienda a ponerse lento, es decir es una central que funciona correctamente no desperdicia recursos y el precio es reducido prácticamente a la mitad de centrales de marcas conocidas.

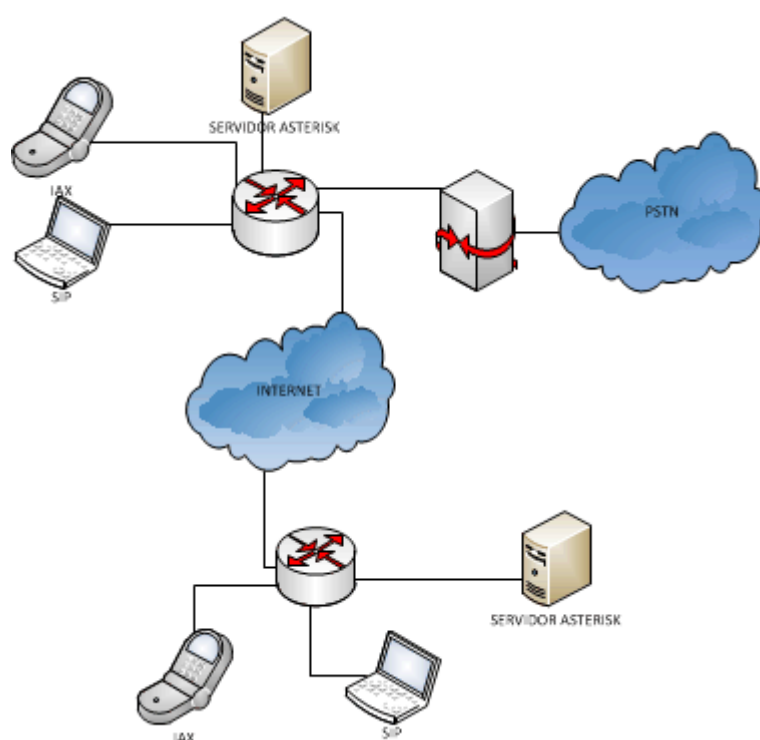


Figura 36. Escenario implementado

Por lo tanto si tenemos el siguiente escenario figura 36, donde se cuenta con dos sucursales remotamente ubicadas y se implementa un servicio de VoIP para comunicarlas, si es mediante un enlace dedicado de la propia empresa el costo se reduce significativamente ya que mientras se realice llamadas dentro de las dos sucursales no implica costos, en el caso de que se utilice la red de Internet el costo varía debido al valor del ancho de banda que se contrate de internet.

Si comparamos con la implementación con líneas tradicionales, debemos considerar que la instalación de cada línea tiene un costo de \$ 60 dólares, es decir si queremos tener 100 líneas, solo en implementación se estará pagando un valor de \$6000 dólares, un costo sumamente elevado al analizado anteriormente con Raspberry PI.

4.5.2. Ahorro mensual

Si consideramos el mismo escenario anteriormente descrito pero con la utilización de la red PSTN en donde el costo base mensual es:

Tabla 15

Costos de Instalación de un Línea tradicional

Tecnología	Precio
Cobre	\$ 6,20
GPON	\$ 20

Estos valores son los que debe pagar como base por cada línea incrementando su valor dependiendo del número de llamadas que se realice, con un costo de \$0,04 ctvos el minuto por llamar a la misma red, y \$0,10 a otras redes fijas.

Por lo tanto si consideramos tener 100 líneas con un consumo de 30min por línea al mes, mensualmente la empresa estaría pagando un valor de \$ 120 dólares, al contrario de una central IP en donde se pagará el costo del ancho de banda de internet que están alrededor de \$50 dólares dependiendo del ancho de banda, considerando que también se tiene datos.

En el caso de aumentar una nueva línea deberá pagar los valores antes mencionados, sin embargo en una central IP se puede incrementar extensiones sin ningún costo.

CAPÍTULO 5 : CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se implementó una central telefónica IP con software libre Asterisk, hardware de bajo costo Raspberry Pi, con una interfaz gráfica con Python amigable para que los estudiantes, donde podrán realizar configuraciones como IVR, ACD, CDR, extensiones, conexión a la PSTN, entre otras.
- La interfaz gráfica diseñada es sencilla de manejar, permite realizar las configuraciones que el estudiante requiera con pequeñas descripciones de ciertos parámetros como guía, además cuenta con la visualización de estas configuraciones en los archivos de Asterisk.
- La interfaz desarrollada es netamente para nivel académico para utilización en los laboratorios de telefonía de la Universidad de las Fuerzas Armadas, pero la central sobre Raspberry Pi puede soportar un mayor número de aplicaciones y configuraciones de las demostradas en este proyecto.
- El costo de un central sobre Raspberry Pi es menor en comparación con las existentes en el mercado, haciendo esta comparación con centrales de las mismas capacidades de las que proporciona la Raspberry Pi.
- Se determinó que el software para desarrollar la interfaz gráfica sea Python, una herramienta que permite realizar un diseño amigable, con mayores velocidades de ejecución, programación de alto nivel más sencilla comparada con java y PHP, además de que es soportado por Raspberry Pi, el hardware utilizado para la central telefónica.

- Se comprobó con la ayuda del software SIPP que la Raspberry Pi como central telefónica funciona correctamente, pudiéndose implementar en ambientes SOHO o pequeñas empresas ya que permite un aproximado de 35 llamadas simultáneas.
- Cuando se realizan entre 30 y 35 llamadas simultáneas la Raspberry Pi llega a utilizar el 100% de sus recursos, por lo que a partir de estos valores puede empezar a existir problemas con la comunicación como: pérdida de llamadas, jitter, pérdida de paquetes, etc.
- Se determinó que las 35 llamadas simultáneas se logran únicamente trabajando la Raspberry Pi como central, ya que si se corre otro tipo de aplicación sobre la tarjeta se sobre saturará, perdiéndose un mayor número de llamadas. Es decir las llamadas simultáneas disminuyen en un 40% dependiendo de la aplicación que se esté manejando a la par de la central.
- Para la comunicación entre centrales es necesario la utilización de un VPN para que la información viaje de forma segura a través de la red.
- Para la comunicación a la PSTN se debe contar con un Gateway de voz, actualmente el protocolo estandarizado y que se encuentra en la mayoría de los equipos es SIP, por lo tanto se trabajó con este protocolo para la comunicación a pesar de que utiliza mayor ancho de banda.

5.2. Recomendaciones

- Para trabajos futuros se puede realizar un análisis de la central para determinar los parámetros de QoS trabajando en la misma red con tráfico VoIP, HTTP, FTTP, etc.
- Realizar un estudio e implementación de la integración de una central VoIP con Asterisk a la Red GSM/ 3G con chan_dongle.
- Para la utilización de la Raspberry Pi es recomendable utilizar una memoria SD mínimo de 4GB de capacidad para evitar inconvenientes con aplicaciones que se van a realizar en las mismas.
- Para la selección de Asterisk se debe considerar para que tipo de aplicaciones se va a implementar la central, si es académica es recomendable utilizar una versión estándar ya que cuenta con actualizaciones cada cierto tiempo con nuevas aplicaciones, si es para nivel empresarial se utiliza un versión de Asterisk LTS que cuenta con actualizaciones en un periodo más largo que una versión estándar.
- La versión de Linux instalada en la Raspberry depende la capacidad de la persona para manejar este tipo de sistemas operativos, ya que existen diferentes versiones siendo Raspbian la más antigua y estable.
- Es necesario que se alimente correctamente a la Raspberry Pi ya que si no recibe la corriente necesaria se presenta problemas como: la tarjeta se pueda bloquear, no permita el arranque del sistema operativo, etc.

REFERENCIAS

- [1] F. Solís, X. Vaca, "Evaluación del sistema de telefonía IP Asterisk mediante la implementación de un prototipo de red en ambientes IPv4 e IPv6", Proyecto previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones, Escuela Politécnica Nacional.
- [2] Querea, Voz datos IP y telefonía IP. Disponible en: http://www.quarea.com/es/asterisk_funcionalidades_basicas_avanzadas.
- [3] G. Gorka, I. Baz, Voz sobre IP Y Asterisk, Irontec.
- [4] H.F. Santamaría, "Telefonía IP: Principios Básicos. Su inserción Actual en el Mercado Empresarial de Europa, Latinoamérica y la Argentina", Tesis doctoral, Universidad Tecnológica Nacional, Buenos Aires.
- [5] ITU World Telecommunication Policy Forum on IP Telephony, realizado el 06/03/01.
- [6] Microsoft, Conceptos de telefonía IP, Disponible en: [https://msdn.microsoft.com/es-es/library/cc739825\(v=ws.10\).aspx](https://msdn.microsoft.com/es-es/library/cc739825(v=ws.10).aspx).
- [7] D. Soto, J. Moreno, M. Diaz, "Artículo sobre telefonía VoIP ", Maestría en Telemática, Universidad DR. Rafael Bellosi Chacín, Maracaibo, 2009.
- [8] J.L. Vaca, "Diseño e implementación de un emulador de central telefónica IP utilizando el software de código abierto ASTERISK para la red de datos de la Facultad de Ingeniería Electrónica de la Escuela Politécnica del Ejército", Proyecto de grado para la obtención del título de Ingeniería, Departamento de Eléctrica y Electrónica, Escuela Politécnica del Ejército, Sangolquí, 2008.
- [9] Copaco, Descripción técnica-servicio de telefonía IP, Disponible en: https://www.copaco.com.py/portal/images/empresa/descripcion_tecnica.pdf.
- [10] A. Alarcon, "Estudio, implementación y análisis de tráfico de una red VoIP bajo el protocolo SIP", Trabajo de grado presentado como requisito parcial para optar por el título de Ingeniero Electrónico, Programa de Ingeniería Electrónica, Universidad Pontificia Bolivariana, Bucaramanga, 2008.
- [11] H.D. Suntaxi, "Diseño de un proyecto piloto de telefonía IP basado en Asterisk e integración al cisco Unified Communications Manager "CUCM"" de la Universidad Politécnica Salesiana sede Quito", Tesis previa a la obtención del

título de Ingeniero Electrónico, Carrera Ingeniería Electrónica, Universidad Politécnica Salesiana, Quito, 2013.

[12] A. Junge, Asterisk más que una central telefónica, Encuentro Linux 2008, Disponible en: http://2008.encuentrolinux.cl/charlas/Asterisk_mas_que_una_central_telefonica.pdf.

[13] D.N. Vargas, A.M. Loaiza, “Instalación y configuración de Software Open Source para monitorear el servicio y la carga de un sistema Asterisk”, Previo a la obtención del título de Ingeniero en Ciencias Computacionales, Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral, Guayaquil, 2009.

[14] Telecomunicaciones Para Gerentes, H.323 y SIP hacia la convergencia, 2014, Disponible en: <http://www.telecomunicacionesparagerentes.com/h-323-y-sip-hacia-la-convergencia/>

[15] J.Barberan, “Implementación de un sistema de VoIP basado en Asterisk”, Proyecto Final de Carrera de Ingeniería Técnica de Telecomunicaciones, Universidad Politecnica de Catalunya, Catalunya, 2009.

[16] A. Maritnez, “Migración de un sistema de telefonía en producción hacia VoIP con Asterisk”., Proyecto fin de carrera Universidad de Sevilla, España, 2011.

[17] CNTI, “Guía práctica de inicialización en el mundo de la telefonía IP. Caracas: Ministerio de poder popular para Ciencia, Tecnología e Industrias Intermedias Gobierno Bolivariano de Venezuela”, 2011.

[18] W, Andrade, “Instalación y configuración del Chan Mobile en Asterisk y su conexión a la red GSM”, Informe de materia de graduación, Escuela Superior Politécnica del Litoral, Guayaquil, 2009.

[19] D. Masip, “El lenguaje Python”, Universidad Oberta de Catalunya, Catalunya, 2011

[20] Rossum, Tutorial Python, 2009, Disponible en: <Http://ralsina.me/static/tutorial-8.pdf>.

[21] A. Reyes, A Byte of Python, 2004, Disponible en: Http://dev.laptop.org/~edsiper/byteofpython_spanish/ch01s02.html

- [22] V. Figueroa, Características de PHP, 2011, Disponible en: <https://es.scribd.com/doc/50288837/Caracteristicas-de-PHP>
- [23] C. Martínez, “Introducción a Raspberry Pi”, Campus Tecnológico URG, 2014, Disponible en: https://docs.google.com/presentation/d/1qUha8oKYnJWQU6TO8rxFt2uNOGxgg0HvlpJK3p_RVIM/edit?pli=1#slide=id.p
- [24] Ermez, “Calidad de Voz IP”, 2014, Disponible en: http://www.emez.com/soporte/documentacion/Todos/AQCT_32/Web/Servidor_Trafico/Calidad_Voz_Ip.htm.
- [25] Cisco, “Voz sobre IP - Ancho de banda por llamada Consumo”, 2006, Disponible en: <http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html>.
- [26] J. P Fernandez, “Diseño e implementación de una central telefónica IP basada en Raspberry Pi y Asterisk”, Facultad de Ingeniería Electrónica, Universidad de Costa Rica, Costa Rica, 2013.
- [27] A. A. Quigua, “Estudio, implementación y análisis de tráfico de una red VoIP basado en el protocolo SIP”, Programa de Ingeniería Electrónica, Escuela de Ingeniería y Administración, Universidad Pontificia Bolivariana, Bucaramanga, 2008
- [28] M. C. Llorente, “Configuración de una red telefónica empresarial mediante el uso del protocolo VoIP. Implementación, configuración y administración de centralita Asterisk y sus funcionalidades”, Proyecto Fin de Carrera, Escuela Técnica Superior de Ingeniería de Telecomunicaciones, Universidad Politécnica de Cartagena, 2013.